proceedings of the fourth
international conference on

computational creativity

ICCC 2013
sydney australia

editors
mary lou maher
tony veale
rob saunders
oliver bown

THE UNIVERSITY OF
SYDNEY

Proceedings of the Fourth International Conference on Computational Creativity

edited by

Mary Lou Maher, Tony Veale, Rob Saunders, Oliver Bown

ICCC 2013
sydney australia

Sydney, New South Wales, Australia

June 2013

Faculty of Architecture, Design and Planning
The University of Sydney
New South Wales
Australia

http://www.computationalcreativity.net/iccc2013/

First published 2013

TITLE: PROCEEDINGS OF THE FOURTH INTERNATIONAL CONFERENCE ON COMPUTATIONAL CREATIVITY

EDITORS: MARY LOU MAHER, TONY VEALE, ROB SAUNDERS, OLIVER BOWN

ISBN: 978-1-74210-317-4

About the cover: Designed by Rob Saunders. Made with Processing.

About the logo: Designed by Oliver Bown and Rob Saunders

Preface

The Fourth International Conference on Computational Creativity 2013 represents a growth and maturity of a conference series that builds on a series of workshops held over ten years and the first three international conferences: the first held in Portugal in 2010, the second held in Mexico in 2011, and the third held in Ireland in 2012. The purpose of this conference series is to make a scientific contribution to the field of computational creativity through discussion and publication on progress in fully autonomous creative systems, modeling human and computational creativity, computational support for human creativity, simulating creativity, and human/machine interaction in creative endeavors. Contributions come from many relevant disciplines, including computer science, artificial intelligence, engineering design, cognitive science, psychology, and art.

This year the conference received 65 paper submissions and 11 demonstration submissions. The peer review process for paper submissions has two stages: In the first stage, all paper submissions were reviewed by three members of the Program Committee. In the second stage, the anonymous reviews were available for comment by all members of the Program Committee and the authors. Decisions about paper acceptances were reviewed and approved by the Steering Committee and decisions about demonstration acceptances were approved by the Organizing Committee. The committees accepted 32 papers and included 8 Demonstrations from authors representing 13 countries: Australia, Canada, Finland, Germany, Ireland, Malta, Mexico, Portugal, Singapore, Spain, Sweden, UK, and USA.

In order to provide a snapshot of current progress in computational creativity and a glimpse of next steps, the conference invites and encourages two kinds of paper submissions: regular papers addressing foundational issues, describing original research on creative systems development and modeling, and position papers describing work-in-progress or research directions for computational creativity. The conference includes a balance of the two: 20 regular papers and 12 position papers. As in previous years, the conference also includes demonstrations in which conference attendees can play with specific implementations of computational creativity. The conference is organized into sessions that reflect the topics of interest this year: areas of creativity in which computation is playing a significant role: music, visual art, poetry, and narrative; and theoretical contributions to computational creativity: metaphor, computational evolution, creative processes, evaluating computational creativity, collective and social creativity, and embodied creativity.

The collection of papers in this conference proceedings shows a maturity in the field through new examples of computational creativity and theoretical advances in understanding generative systems and evaluation of computational creativity. The conference series demonstrates success as we see publications that build on the advances of previous years through references to papers published in this conference series. We look forward to this publication providing the foundation for future developments in computational creativity.

Mary Lou, Tony, Rob, and Ollie

June 2013

Conference Chairs
General Chair: Tony Veale, University College, Dublin, Ireland
Local Co-Chairs: Rob Saunders & Oliver Bown, University of Sydney, Australia
Program Chair: Mary Lou Maher, University of North Carolina, Charlotte, USA

Local Organizing Committee
Kazjon Grace, University of North Carolina, Charlotte, USA
Roger Dean, University of Western Sydney, Australia

Steering Committee
Amílcar Cardoso, University of Coimbra, Portugal
Simon Colton, Imperial College London, UK
Pablo Gervás, Universidad Complutense de Madrid, Spain
Nick Montfort, Massachusetts Institute of Technology, USA
Alison Pease, University of Edinburgh, UK
Rafael Pérez y Pérez, Autonomous Metropolitan University, México
Graeme Ritchie, University of Aberdeen, UK
Rob Saunders, University of Sydney, Australia
Dan Ventura, Brigham Young University, USA
Tony Veale, University College, Dublin, Ireland
Geraint A. Wiggins, Queen Mary, University of London, UK

Program Committee
Alison Pease, University of Edinburgh, UK
Amílcar Cardoso, University of Coimbra, Portugal
Andres Gomez De Silva, Instituto Tecnológico Autónomo de México
Anna Jordanous, King's College London, UK
Ashok Goel, Georgia Institute of Technology, USA
Dan Ventura, Brigham Young University, USA
David Brown, Worcester Polytechnic Institute, USA
David Moffat, Glasgow Caledonian University, UK
Diarmuid O'Donoghue, National University of Ireland, Maynooth, Ireland
Douglas Fisher, Vanderbilt University, USA
Geraint Wiggins, Goldsmiths College, University of London, UK
Graeme Ritchie, University of Aberdeen, UK
Hannu Toivonen, University of Helsinki, Finland
Henry Lieberman, Massachusetts Institute of Technology, USA
John Barnden, The University of Birmingham, UK
John Gero, Krasnow George Mason University, USA
Jon McCormack, Monash University, Australia
Kazjon Grace, University of Sydney, Australia
Kyle Jennings, University of California, Berkeley
Mark Riedl, Georgia Institute of Technology, USA
Nick Bryan-Kinns, Queen Mary University of London, UK
Nick Montfort, Massachusetts Institute of Technology, USA
Oliver Bown, University of Sydney, Australia
Oliviero Stock, European Alliance for Innovation, Italy
Pablo Gervas, Universidad Complutense de Madrid, Spain
Paulo Gomes, University Of Coimbra, Portugal
Paulo Urbano, University of Lisboa, Portugal

Contents

Session 5 Visual Art

Session 6 Computational Processes for Creativity

Session 7 Evaluating Computational Creativity

Session 9 Poetry

Session 10 Narrative

Session 11 Collective and Social Creativity

Session 12 Embodied Creativity

Demonstrations

Computationally Created Soundscapes with Audio Metaphor

Miles Thorogood and Philippe Pasquier
School of Interactive Art and Technology
Simon Fraser University
Surrey, BC V3T0A3 CANADA
mthorogo@sfu.ca

Abstract

Soundscape composition is the creative practice of processing and combining sound recordings to evoke auditory associations and memories within a listener. We present *Audio Metaphor*, a system for creating novel soundscape compositions. *Audio Metaphor* processes natural language queries derived from Twitter for retrieving semantically linked sound recordings from online user-contributed audio databases. We used a simple natural language processing to create audio file search queries, and we segmented and classified audio files based on general soundscape composition categories. We used our prototype implementation of Audio Metaphor in two performances, seeding the system with keywords of current relevance, and found that the system produced a soundscape that reflected Twitter activity and kept audiences engaged for more than an hour.

1 Introduction

Creativity is a preeminent attribute of the human condition that is being actively explored in artificial intelligence systems aiming at endowing machines with creative behaviours. Artificial creative systems have simulated or been inspired by human creative processes, including, painting, poetry, and music. The aim of these systems is to produce artifacts that humans would judge as creative. Much of the successful research in musical creative systems has focussed on symbolic representations of music, often with corpora of musical scores. Alternatively, non-symbolic forms of music have been little explored in as much detail.

Soundscape composition is a type of non-symbolic music aimed to rouse listeners memories and associations of soundscapes using sound recordings. A soundscape is the audio environment perceived by a person in a given locale at a given moment. A listener brings a soundscape to mind with higher cognitive functions like template matching of the perceived world with known sound environments and deriving meaning from the triggered associations (Botteldooren et al. 2011). People communicate their subjective appraisal of soundscapes using natural language descriptions, revealing the semiotic cues of soundscape experiences (Dubois and Guastavino 2006).

Soundscape composition is the creative practice of processing and combining sound recordings to evoke auditory associations and memories within a listener. It is positioned along a continuum with concrete music that uses found sound recordings, and electro-acoustic music that uses more abstracted types of sounds. Central to soundscape composition, is processing sound recordings. There are a range of approaches to using sound recordings. One approach is to portray a realistic place and time by using untreated audio recordings, or, recordings with only minor editing (such as cross-fades). Another is to evoke imaginary circumstances by applying more intensive processing. In some cases, these manufactured sound environments appear imaginary, by the combination of largely untreated with more highly processed sound recordings. For example, the soundscape composition *Island*, by Canadian composer Barry Truax (Truax 2009), adds a mysterious quality to a recognizable sound environment by contrasting clearly discernible wave sounds against less-recognizable background drone and texture sounds.

Soundscape composition requires many decisions about selecting and cutting audio recordings and their artistic combination. These processes become exceedingly time consuming for people when large amounts of audio data are available, as is now the case with online databases. As such, different generative soundscape composition systems have automated many sub-procedures of the composition process, but we have not found any systems in the literature to date that use natural language processing for generative soundscape composition. Likewise, automatic audio segmentation for soundscape composition specific categories is an area not yet explored.

The system described here searches online for the most recent Twitter posts about a small set of themes. Twitter provides an accessible platform for millions of discussions and shared experiences through short text-based posts (Becker, Naaman, and Gravano 2010). In our research, audio file search queries are generated from natural language queries derived from Twitter. However, these requests could be a memory described by a user, a phrase from a book, or a section of a research paper.

Audio Metaphor accepts a natural language query (NLQ), which is made into audio file search queries by our algorithm. The system searches online for audio files semantically related to word features in the NLQ. The resulting audio file recommendations are classified and segmented based

upon the soundscape categories *background*, *foreground*, and *background with foreground*. A composition engine autonomously processes and combines segmented audio files.

The title of *Audio Metaphor* refers to the idea that audio representations of NL queries that the system generates may not have literal associations. Although, in some cases, an object referenced in the NL query may have a direct referential sound such as with "raining outside" that results in a type of *audio analogy*. However, an example that is not as direct such as, "A brooding thought struck me down" has no such direct referent to an object in the world. In this latter case, *Audio Metaphor* would create a composition by processing sound recordings that have some semantic relationship with words in the NL query. For example, the sound of a storm and the percussive striking of an object are the types of sounds that would be processed in this case.

Margret A. Boden actively proposes types of creativity being synthesized by computational means (Boden 1998). She states, that *combinatorial* type creativity "involves novel (improbable) combinations of familiar ideas ... wherein newly associated ideas share some inherent conceptual structure." The artificial creative system here uses semantic inference driven by NLQs as a way to frame the soundscape composition and make use of semantic structures inherent in crowdsourced systems. Further to this, the system associates words with sound recordings for combining into novel representations of texts. For this reason, the system is considered to exhibit *combinatorial* creative behaviour.

Our contribution is a creative and autonomous soundscape composition system with a novel method of generating compositions from natural language input and crowd-sourced sound recordings. Furthermore, we present a method of audio file segmentation based on soundscape categories, and a soundscape composition engine that contrasts sound recording segments with different levels of processing.

We outline our research in the design of an autonomous soundscape composition system called *Audio Metaphor*. In the next section, we show the related works in the domains of soundscape studies and generative soundscape composition. We go on to describe the system architecture, including natural language processing, classification and segmentation, and the soundscape composition engine. The system is then disused in terms of a number of performances and presentations. We conclude with our ideas for future work.

2 Related Work

Birchfield, Mattar, and Sundaram (2005) describe a system that uses an adaptive user model for context-aware soundscape composition. In their work, the system has a small set of hand-selected and hand-labelled audio recordings that were autonomously mixed together with minimal processing. Similarly, Eigenfeldt and Pasquier (2011) employ a set of hand-selected and hand-labelled environmental sound recordings for the retrieval of sounds from a database by autonomous software agents. In their work, agents analyze audio when selecting sounds to mix based on low-level audio features. In both cases, listening and searching for selecting audio files is very time consuming.

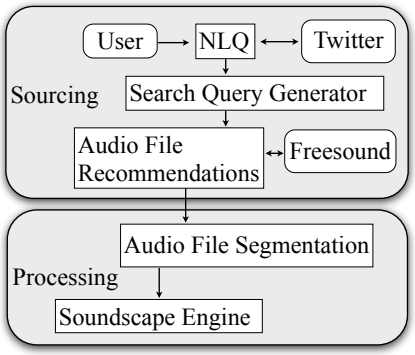


Figure 1: *Audio Metaphor* system architecture overview.

A different approach to selecting and labelling sound recordings is to take advantage of collaborative tagging of online user-contributed collections of sound recordings. This is a crowdsourcing process where a body of tags is produced collaboratively by human users connecting terms to documents (Halpin, Robu, and Shepherd 2007). In online environments, collaborative tags are part of a shared language made manifest by users (Marlow et al. 2006). Online audio repositories such as pdSounds (Mobius 2009) and Freesound (Akkermans et al. 2011) demonstrate collaborative tagging systems applied to sound recordings.

A system that uses collaborative tags to retrieve sound recordings is described by Janer, Roma, and Kersten (2011). In their work, a user defines a soundscape composition by entering locations on a map that has sounds tags associated with various locations. As the user navigates the map, a soundscape is produced. In related research, the locations on a map are used as a composition environment (Finney and Janer 2010). Their compositions use hand-selected sounds, which are placed in close and far proximity based upon semantic identifiers derived from tags.

3 System Architecture

Audio Metaphor creates unique soundscape compositions that represent the words in an NLQ using a series of processes as follows:

- Receive a NLQ from a user, or Twitter;
- Transforms a NLQ into audio file search queries;
- Search online for audio file recommendations;
- Segment audio files into soundscape regions;
- Process and combine audio segments for soundscape composition.

In the *Audio Metaphor* system, these processes are handled by sequentially as is shown in Figure 1. [1]

[1] A modular approach was taken for the system design. Accordingly, the system is flexible to be used for separate objectives, including, making audio file recommendations to a user from an NLQ, and deriving a corpus of audio segments.

rainy autumn day vancouver
rainy autumn day
autumn day vancouver
rainy autumn
autumn day
day vancouver
rainy
autumn
day
vancouver

Table 1: All sub-lists generated from a word-feature list from the query "On a rainy autumn day in Vancouver'.

3.1 Audio File Retrieval Using Natural Language Processing

The audio file recommendation module creates audio file search queries given a natural language request and a maximum number of audio file recommendations for each search.

The Twitter web API (Twitter API) is used to retrieve the 10 most recent posts related to a theme to find current associations. The longest of these posts is then used as a natural language query. To generate audio file search queries, a list of word features is extracted from the input text and generates a queue of all unique sublists. These sublists are used as search queries, starting with the longest first. The aim of the algorithm is to minimize the number of audio files returned and still represent all the word features in the list. When a search query returns a positive result, all remaining queries that contain any of the successful word features are removed from the queue.

To extract the word features from the natural language query, we use essentially the same method as that proposed by Thorogood, Pasquier, and Eigenfeldt (2012), but with some modifications. The algorithm first removes common words listed in the Oxford English Dictionary Corpus, leaving only nouns, verbs, and adjectives. Words are kept in order and treated as a list. For example, with the word feature list from the natural language query "The angry dog bit the crying man," "angry dog bit crying man," is more valid than "angry man bit crying dog."

The algorithm for generating audio file queries essentially extracts all the sublists from the NLQ that have a length greater than or equal to 1. For example, a simple request such as "On a rainy autumn day in Vancouver" is first processed to extract the word feature list: rainy, autumn, day, vancouver. After that, sub-lists are generated as shown in Table 1.

Audio Metaphor accesses the Freesound audio repository for audio files with the Freesound API. Freesound is an online collaborative database with over 120,000 audio clips. The indexed data includes user-entered descriptions and tags. The content of the audio file is inferred from user-contributed commentary and social tags. Although there is no explicit user rating of audio files, a download counter for each file provides a measure of its popularity, and search results are presented by descending popularity count.

The sublists are used to search online for semantically related audio files using an exclusive keyword search. Sublists are used in the order created, from largest to smallest. A search is considered successful when it returns one or more recommendations. Additionally, the algorithm optimizes audio file recommendations by ignoring future sublists that contain word features from a previously successful search. The most favourable result is a recommendation for the longest sub-list, with the worst case being no recommendations. In practice, the worst case is, typically, a recommendation for each singleton word feature.

For each query, the URLs of the recommendations are logged in a separate list. The list is constrained to a number specified at the system startup. Furthermore, if a list has less than the number of files requested it is considered sparsely populated and no further modification made to its items. For example, if the maximum number of recommendations specified for each query is five, and there are two queries where one returns nine recommendations and the other three, the longer list will be constrained to five, and the empty items of the second list are ignored.

The separate lists of audio file recommendations are then presented to the audio segmentation module.

3.2 Audio File Classification and Segmentation

Audio segmentation is an essential preprocessing step in many audio applications (Foote 2000). In soundscape composition, a composer will choose background and foreground sound regions to combine into new soundscapes.

Background and foreground sounds are general categories that refer to a signal's perceptual class. Background sounds seem to come from farther away than foreground sounds or occur often enough to belong to the aggregate of all sounds that make up the background texture of a soundscape. This is synonymous with a ubiquitous sound (Augoyard and Torgue 2006): a sound that is diffuse, omnidirectional, constant, and prone to sound absorption and reflection factors having an overall effect on the quality of the sound. Urban drones and the purring of machines are two examples of ubiquitous or background sound. Conversely, foreground sounds are typically heard standing out clearly against the background. At any moment in a sound recording, there may be either background sound, foreground sound, or a combination of both.

Segmenting an audio file is a process of listening to the recording for salient features and *cutting* regions for later use. To automate this process, we have designed an algorithm to classify segments of an audio file and concatenate neighbouring segments with the same label. An established technique for classification of an audio recording is to use a supervised machine learning algorithm trained with examples of classified recordings.

3.3 Audio Features Used for Segmentation

The classifier models the generic soundscape categories *background*, *foreground*, and *background with foreground*. We use a vector of the low-level audio features total-loudness, and the first three mel-frequency cepstral coefficients (MFCC). These features reflect the behaviour of the human auditory system, which is an important aspect of

soundscape studies. They are extracted at a frame-level from an audio signal with a window of 23 ms and a step size of 11.5 ms using the Yaafe audio feature extraction software package (Mathieu et al. 2010).

MFCC audio features represent the spectral characteristics of a sound by a small number of coefficients calculated by the logarithm of the magnitude of a triangular filter bank. We use an implementation of MFCC that builds a logarithmically spaced filter bank according to 40 coefficients mapped along the perceptual Mel-scale by:

$$M(f) = 1127 \log \left(1 + \frac{f}{700}\right) \quad (1)$$

where f is the frequency in Hz.

Total loudness is the characteristic of a sound associated with the sensation of intensity. The human auditory system affects the perception of intensity of different frequencies. One model of loudness (Zwicker 1961) takes into account the disparity of loudness at different frequencies along the Bark scale, which corresponds to the first 24 critical bands of hearing. Bands near human speech frequencies have a lower threshold than those of low and high frequencies. The conversion from a frequency in Hz f to the equivalent frequency in the Bark scale B is calculated with the following formula (Traunmuller 1990).

$$B(f) = 13 \arctan(0.00076f) + 3.5 \arctan\left(\frac{f}{7500}\right)^2 \quad (2)$$

Where f is the frequency in Hz. A specific loudness is the loudness calculated at each Bark band; the total loudness is the sum of individual specific loudnesses over all bands. Because a soundscape is perceived by a human not at the sample level, but over longer time periods, we use a so called *bag of frames approach* (Aucouturier and Defreville 2007) to account for longer signal durations. Essentially, this kind of approach considers frames that represent a signal have possibly different values, and the density distribution of frames provides a more effective representation than a singular frame. Statistical methods, such as the mean and standard deviation of features, recapitulate the texture of an audio signal, and provides a more effective representation than a single frame.

In our research, audio segments are represented with an eight-dimensional feature vector of the means and standard deviations from the total loudness and the first 3 MFCC. The mean and standard deviation of the feature vector models the *background*, *foreground*, and *background with foreground* soundscape categories well. For example, sounds distant from the listener and considered background sound will typically have a smaller mean total loudness. Sounds that occur often enough will have a smaller standard deviation of those in foreground listening. MFCC takes into account the spectrum of the sound affected by its source placement in the environment.

3.4 Supervised Classifier Used for Segmentation

We used a Support Vector Machine classifier (SVM) to classify audio segments. SVMs have been used in environmental sound classification problems, and consistently demonstrated good classification accuracy. A SVM is a non-probabilistic classifier that learns optimal separating hyperplanes in a higher dimensional space from the input. Typically, classification problems present non-linearly separable data that can be mapped to a higher-dimensional space with a kernel function. We use the C-support vector classification (C-SVC) algorithm shown by Chang and Lin (2011). This algorithm uses a radial basis function as a kernel, which is suited to a vector with a small number of features and takes into account the relation between class labels and attributes being non-linear.

Training Corpus The classifier was trained using feature vectors from a pre-labelled corpus of audio segments. The training corpus consists of 30 segments between 2 and 7 seconds long. Audio segments were labelled from a consensus vote by human subjects in an audio segment classification study. The study was conducted online through a web browser. Audio was played to participants using an HTML5 audio player object. This player allowed participants to repeatedly listen to a segment. Depending on the browser software, the audio format of segments was either MP3 at 196 kps, or Vorbis at an equivalent bit rate. Participants selected a category from a set of radio buttons and each selection was confirmed when the participant pressed a button to listen to the next segment.

There were 15 unique participants in the study group from Canada and the United States. Before the study started, an example for each of the categories, background, foreground, and background with foreground, was played, and a short description of the categories was displayed. Participants were asked to use headphones or audio monitors to listen to segments. Each participant was asked to listen to the randomly ordered soundscape corpus. On completing the study, the participant's classification results were uploaded into a database for analysis.

The results of the study were used to label the recordings by a majority vote. Figure 2 shows the results of the vote. Results of the vote gave the labelling to the recordings. There are a total of 10 recordings for each of the categories.

A quantitative analysis of the voter results shows the average agreement of recordings for each category as follows: background 84.6% (SD=18.6%); foreground 77.0% (SD=10.4%), and; background with foreground 76.2% (SD=13.4%). The overall agreement was shown to be 79.3% (SD=4.6%).

Classifier Evaluation We evaluated the classifier, using the training corpus, with a 10-fold cross validation. The results summary is shown in Table 2. The classifier achieved an overall sample accuracy of 80%, which shows that the classifier was human competitive against the overall human agreement statistic of 79.3%.

The kappa statistic is a chance-corrected measure showing the accuracy of prediction among each k-fold model. A kappa score of 0 means the classifier is performing only as well as chance; 1 implies a perfect agreement; and a kappa score of .7 is generally considered satisfactory. The kappa score of .7 in the results shows a good classification accuracy was achieved using the described method.

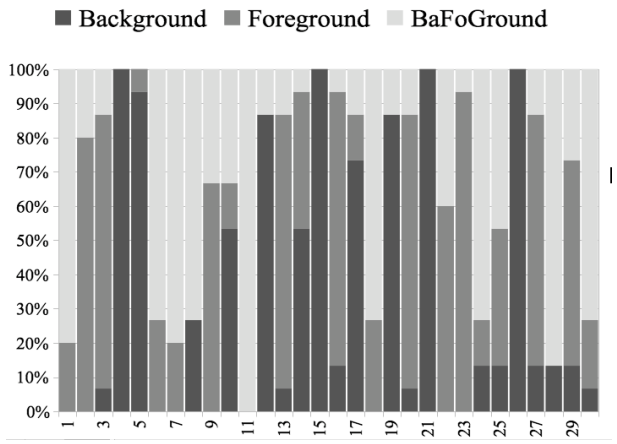


Figure 2: Audio classification vote results from human participants for 30 sound recordings with three categories: Background, Foreground, and Background with Foreground (BaForound) sound.

Table 2: Summary of SVM classifier with the mean and standard deviation for features total loudness and 3 MFCC.

Correctly classified instances	24	80%
Incorrectly classified instances	6	20%
Kappa statistic	0.7	

These performance measures are reflected by the confusion matrix in Table 3. All 10 of the audio segments labelled "background" from the study were classified correctly. The remaining audio segments, labelled "foreground" and "background with foreground," were correctly classified 7 out of 10 times, with the highest level of confusion between these latter categories.

3.5 Background-Foreground Segmentation

In our segmentation method, we use a 500 ms sliding analysis window with a hop size of 250 ms. We found that for our application an analysis window of this length provided reasonable information for the bag of frames approach and ran with satisfactory computation time. The resulting feature vector is classified and labelled as belonging to one of the three categories. In order to create labelled regions of more than one window, neighbouring windows with the same label are concatenated and the start and end time of the new window are logged.

To demonstrate the segmentation algorithm, we used a 9 second audio file containing a linear combination of background, foreground, and background with foreground regions. Figure 3. shows the ground truth with the solid black line, and algorithm segmentation of the audio file with background, foreground, and background with foreground labelled regions applied. We use the SuperCollider3 software package for visualizing the segmented waveform sc3. This example shows concatenated segments labelled as regions. One of the background with foreground segments was misclassified resulting in a slightly longer foreground region than the ground truth classification.

Table 3: Confusion matrix of SVM classifier for the categories background (BG), foreground (FG), and background with foreground (BgFg).

Bg	Fg	BgFg	
10	0	0	Bg
0	7	3	Fg
1	2	7	BgFg

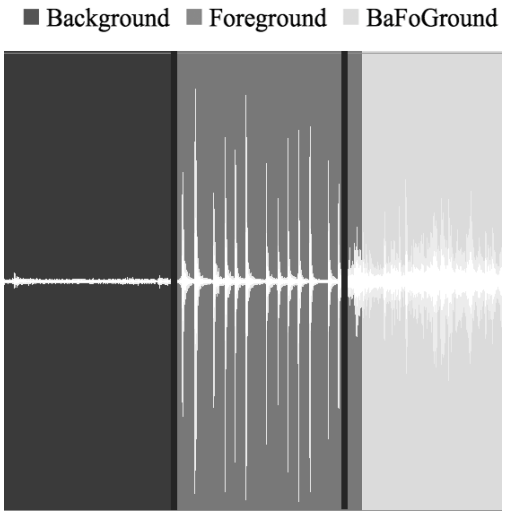


Figure 3: Segmentation of the audio file with ground-truth regions (black line) and segmented regions Background (dark-grey), Foreground (mid-grey), and Background with Foreground (light-grey).

The audio files and the accompanying segmentation data are then presented to the composition module.

3.6 Composition

The composition module creates a layered two-channel soundscape composition by processing and combining classified audio segments. Each layer in the composition consists of processed background, foreground, and background with foreground sound recordings. Moreover, an agent-based model is used in conjunction with a heuristic in order to handle different sound recordings and mimic the decisions of a human composer. Specifically, we based this heuristic from production notes for the soundscape composition *Island*, by Canadian composer Barry Truax. In these production notes, Truax gives detailed information on how sound recordings are effected, and the temporal arrangement of sounds.

In our modelling of these processes, we chose to use the first page of the production notes, which corresponds to around 2 minutes of the composition. Furthermore, we framed the model to comply with the protocol of the seg-

mentation labels and aesthetic evaluations by the authors. A summary of the model is as follows:

- Regions labelled *background* are played sequentially in the order presented by the segmentation. They are processed to form a dramatic textured background. This processing is carried out by first playing the region at 10% of its original speed and applying a stereo time domain granular pitch shifter with ratios 1:0.5 (down an octave) and 1:0.667 (down a 5th). We added a Freeverb reverb (Smith 2010) with a room size of 0.25 to give the texture a more spacious quality. A low pass filter with a cutoff frequency at 800 Hz is used to obscure any persistent high end detail. Finally, a slow spatialization is applied in the stereo field at a rate of 0.1 Hz.
- Regions labelled *foreground* are chosen from the foreground pool by a roll of the dice. They are played individually, separated by a period proportional to the duration of the current region played $t = d^{.75} + d + C$, where t is the time between playing the next region, d is the duration of the current region, and C is a constant controlling the minimum duration between regions. In order to separate them from the background texture, foreground regions are processed by applying a band pass filter with a resonant frequency 2,000 Hz and high Q value of 0.5. Finally, a moderate spatialization is applied in the stereo field at a rate of .125 Hz.
- Regions labelled *background with foreground* are slowly faded in and out to evoke a mysterious quality to the soundscape. They are chosen from the pool of regions by a roll of the dice and are played for an arbitrarily chosen duration of between 10 and 20 seconds. Regions with a length less than the chosen duration are looped. In order to achieve a separation from the background texture and foreground sounds, regions are processed by applying a band pass filter with a resonant frequency 8,000 Hz and high Q value of 0.1. The addition of a Freeverb reverb with a room size of 0.125 and a relatively fast spatialization at a rate of 1 Hz was used to further add to the mysterious quality of the sound.

This composition model is deployed individually by each of agents of the system, who are responsible for processing a different audio file. An agents decisions are, choosing labelled regions of an audio recording, processing and combining them in a layered soundscape composition according to the composition model.

Because of the potentially large number of audio files available to the system, and in order to limit the acoustic density of a composition, a maximum number of agents are specified on system start-up. If there are more audio file results than there are agents to handle them, the extra results are ignored. Equally, if the number of results is smaller then the number of agents, agents without tasks are temporarily ignored.

An agent uses the region labels of the audio file to decide which region to process. An audio file may have a number of labelled regions. If there is no region of a type then that type is ignored. The agent can play one of each types of region simultaneously.

4 Qualitative Results

Audio Metaphor has been used in performance environments. In one case, the system was seeded with the words "nature," "landscape," and "environment." There were roughly 150 people in the audience. They were told that the system was responding to live Twitter posts and shown the console output of the search results. During the performance, there was an earthquake off the coast of British Columbia, Canada, and the current Twitter posts focused on news of the earthquake. *Audio Metaphor* used these as natural language requests, searched online for sound recordings related to earthquakes, and created a soundscape composition. The sound recordings processed by the system included an earthquake warning announcement, the sound of alarms, and a background texture of heavy destruction. The audience reacted by checking to see if this event was indeed real. This illustrated how the semantic space of the soundscape composition effectively maps to the concepts of a natural language request.

In a separate performance, *Audio Metaphor* was presented to a small group of artists and academics. This took place during the height of the 2012 conflict in Syria, and the system was seeded with the words "Syria," "Egypt," and "conflict." The soundscape composition presented segments of spoken word, traditional instruments, and other sounds. The audience listened to the composition for over an hour without losing its engagement with the listening experience. One comment was, "It was really good, and we didn't get bored." The sounds held peoples' attention because they were linked to current events, and the processing of sound recordings added to the interest of the composition.

Because the composition model deployed in *Audio Metaphor* is based of a relatively short section of a composition, there was not a great deal of variation in the processing of sound recordings. The fact that people were engaged for such long periods of time suggests that other factors contributed to the novel stimulus. Our nascent hypothesis is that the dynamic audio signal of recordings, in addition to the processing of audio files contributed to listeners ongoing engagement. [2]

5 Conclusions and Future Work

We describe a soundscape composition engine that chooses audio segments using natural language queries, segments and classifies the resulting files, processes them, and combines them into a soundscape composition at interactive speeds. This implementation uses current Twitter posts as natural language queries to generate search queries and retrieves audio files that are semantically linked to queries from the Freesound audio repository.

The ability of *Audio Metaphor* to respond to current events was shown to be a strong point in audience engagement. The presence of signifier sounds evoked listeners' associations of concepts. Listener engagement was further reinforced through the artistic processing and combination of sound recordings.

[2] Sound examples of *Audio Metaphor* using the composition engine can be found at http://www.audiometaphor.ca/aume

Audio Metaphor can be used to help sound artists and autonomous systems retrieve and cut sound field recordings from online audio repositories. Although, its primary function, as we have demonstrated, is autonomous machine generated soundscapes for performance environments and installations. In the future, we will evaluate people's response to these compositions by distributing them to user-contributed music repositories and analyzing user comments. These comments can then be used to inform the *Audio Metaphor* soundscape composition engine.

Although the system generates engaging and novel soundscape compositions, the composition structure is tightly regulated by the handling of background and foreground segments. In future work, we aim toward equipping our system with the ability to evaluate its audio output, in order to make more in-depth composition decisions. By developing these methods, *Audio Metaphor* will be not only be capable of processing audio files to create novel compositions, but, additionally, be able to respond to the compositions it has made.

6 Acknowledgments

This research was funded by a grant from the Natural Sciences and Engineering Research Council of Canada. The authors would also like to thank Barry Truax for his composition and production documentation.

References

Akkermans, V.; Font, F.; Funollet, J.; de Jong, B.; Roma, G.; Togias, S.; and Serra, X. 2011. Freesound 2: An Improved Platform for Sharing Audio Clips. In *International Society for Music Information Retrieval Conference*.

Aucouturier, J.-J., and Defreville, B. 2007. Sounds like a park: A computational technique to recognize soundscapes holistically, without source identification. *19th International Congress on Acoustics*.

Augoyard, J., and Torgue, H. 2006. *Sonic Experience: A Guide to Everyday Sounds*. McGill-Queen's University Press.

Becker, H.; Naaman, M.; and Gravano, L. 2010. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, 291–300. New York, NY, USA: ACM.

Birchfield, D.; Mattar, N.; and Sundaram, H. 2005. Design of a generative model for soundscape creation. In *International Computer Music Conference*.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103(1–2):347 – 356.

Botteldooren, D.; Lavandier, C.; Preis, A.; Dubois, D.; Aspuru, I.; Guastavino, C.; Brown, L.; Nilsson, M.; and Andringa, T. C. 2011. Understanding urban and natural soundscapes. In *Forum Acusticum*, 2047–2052. European Accoustics Association (EAA).

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Dubois, D., and Guastavino, C. 2006. *In search for soundscape indicators : Physical descriptions of semantic categories*.

Eigenfeldt, A., and Pasquier, P. 2011. Negotiated content: Generative soundscape composition by autonomous musical agents in coming together: Freesound. In *Proceedings of the Second International Conference on Computational Creativity*, 27–32. México City, México: ICCC.

Finney, N., and Janer, J. 2010. Soundscape Generation for Virtual Environments using Community-Provided Audio Databases. In *W3C Workshop: Augmented Reality on the Web*.

Foote, J. 2000. Automatic audio segmentation using a measure of audio novelty. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, 452 –455 vol.1.

Halpin, H.; Robu, V.; and Shepherd, H. 2007. The complex dynamics of collaborative tagging. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, 211–220. New York, NY, USA: ACM.

Janer, J.; Roma, G.; and Kersten, S. 2011. Authoring augmented soundscapes with user-contributed content. In *ISMAR Workshop on Authoring Solutions for Augmented Reality*.

Marlow, C.; Naaman, M.; Boyd, D.; and Davis, M. 2006. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, HYPERTEXT '06, 31–40. New York, NY, USA: ACM.

Mathieu, B.; Essid, S.; Fillon, T.; J.Prado; and G.Richard. 2010. YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software. In *Proceedings of the 2010 International Society for Music Information Retrieval Conference (ISMIR)*. Utrecht, Netherlands: ISMIR.

Mobius, S. 2009. pdSounds. Available online at http://www.pdsounds.org/; visited on April 12th 2012.

Smith, J. O. 2010. *Physical Audio Signal Processing*. W3K Publishing. online book.

Thorogood, M.; Pasquier, P.; and Eigenfeldt, A. 2012. Audio metaphor: Audio information retrieval for soundscape composition. In *Proceedings of the 6th Sound and Music Computing Conference*.

Traunmuller, H. 1990. Analytical expressions for the tonotopic sensory scale. *The Journal of the Acoustical Society of America* 88(1):97–100.

Truax, B. 2009. Island. In Soundscape Composition DVD. DVD-ROM (CSR-DVD 0901). Cambridge Street Publishing.

Twitter API. Available online at https://dev.twitter.com/docs/; visited on April 12th 2012.

Zwicker, E. 1961. Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen). *The Journal of the Acoustical Society of America* 33(2):248.

Generating Apt Metaphor Ideas for Pictorial Advertisements

Ping Xiao, Josep Blat
Department of Information and Communication Technologies
Pompeu Fabra University
C./ Tànger, 122-140 Barcelona 08018 Spain
{ping.xiao, josep.blat}@upf.edu

Abstract

Pictorial metaphor is a popular way of expression in creative advertising. It attributes certain desirable quality to the advertised product. We adopt a general two-stage computational approach in order to generate apt metaphor ideas for pictorial advertisements. The first stage looks for concepts which have high imageability and the selling premise as one of their prototypical properties. The second stage evaluates the aptness of the candidate vehicles (found in the first stage) in regard to four metrics, including affect polarity, salience, secondary attributes and similarity with tenor. These four metrics are conceived based on the general characteristics of metaphor and its specialty in advertisements. We developed a knowledge extraction method for the first stage and utilized an affect lexicon and two semantic relatedness measures to implement the aptness metrics of the second stage. The capacity of our computer program is demonstrated in a task of reproducing the pictorial metaphor ideas used in three real advertisements. All the three original metaphors were replicated, as well as a few other vehicles recommended, which, we consider, would make effective advertisements as well.

Introduction

A pictorial advertisement is a short discourse about the advertised product, service or idea (all referred to as 'product' afterwards). Its core message, namely the selling premise, is a proposition that attributes certain desirable quality to the product (Maes and Schilperoord 2008). A single proposition can be expressed virtually in an unlimited number of ways, among which some are more effective than the others. The 'how to say' of an ad is conventionally called the 'idea'. 'Pictorial metaphor' is the most popular way of expression in creative advertising (Goldenberg, Mazursky and Solomon 1999). A pictorial metaphor involves two dimensions, 'structural' and 'conceptual' (Forceville 1996; Phillips and McQuarrie 2004; Maes and Schilperoord 2008). The structural dimension concerns how visual elements are arranged in a 2D space. The conceptual dimension deals with the semantics of the visual elements and how they together construct a coherent message. We see that the operations in the structural and conceptual dimensions are quite different issues. In any of these two dimensions, computational creativity is not a trivial issue. In this paper, we are focusing on only one dimension, the conceptual one.

The conceptual dimension of pictorial metaphors is not very different from verbal metaphors (Foss 2005). A metaphor involves two concepts, namely 'tenor' and 'vehicle'. The best acknowledged effect of metaphor is highlighting certain aspect of the tenor or introducing some new information about the tenor. Numerous theories have been proposed to account for how metaphors work. The interaction view is the dominant view of metaphor, which we also follow. It was heralded by Richards (1936) and further developed by Black (1962). According to Black, the principal and subsidiary subjects of metaphor are regarded as two systems of "associated commonplaces" (commonsense knowledge about the tenor and vehicle). Metaphor works by applying the system of associated commonplaces of the subsidiary subject to the principal subject, "to construct a corresponding system of implications about the principal subject". Any associated commonplaces of the principal subject that conform the system of associated commonplaces of the subsidiary subject will be emphasized, and any that does not will be suppressed. In addition, our view of the subsidiary subject is also altered.

Besides theories, more concrete models have been proposed, mainly the salience imbalance model (Ortony 1979), the domain interaction model (Tourangeau and Sternberg 1982), the structure mapping model (Gentner 1983; Gentner and Clement 1988), the class inclusion model (Gluckberg and Keysar 1990, 1993) and the conceptual scaffolding and sapper model (Veale and Keane 1992; Veale, O'Donoghue and Keane 1995). Furthermore, these models suggest what make good metaphors, i.e. metaphor aptness, which is defined as "the extent to which a comparison captures important features of the topic" (Chiappe and Kennedy 1999).

In the rest of this paper, we first specify the problem of generating apt metaphor ideas for pictorial advertisements. Then, the relevant computational approaches in the literature are reviewed. Next, we introduce our approach to the stated problem and the details of our implementation. Subsequently, an experiment with the aim of reproducing three pictorial metaphors used in real advertisements and the

results generated by our computer program are demonstrated. In the end, we conclude the work presented in this paper and give suggestion about future work.

Problem Statement

The whole range of non-literal comparison, from mere-appearance to analogy (in the terms of Gentner and Markman (1997)), is featured in pictorial advertisements. But, analogies are rare. What appear most frequently are metaphors with the mapping of a few attributes or relations. This type of pictorial metaphors is the target of this paper. To generate pictorial metaphors for advertisements, our specific problem is searching for concepts (vehicles), given the product (tenor), its selling premise (the property concept) and some other constraints specified in an advertising brief. The metaphor vehicles generated have to be easy to visualize and able to establish or strengthen the connection between the product and the selling premise.

There are two notes specific to advertisements that we would like to mention. One is about the tenor of metaphor. In pictorial ads, not only the product, but also "the internal components of the product and the objects that interact with it" are often used as tenors (Goldenberg, Mazursky and Solomon 1999). The other note is about the selling premise. Metaphors in advertisements are more relevant to communicating intangible, abstract qualities than talking about concrete product facts (Phillips and McQuarrie 2009). Therefore, we are primarily considering abstract selling premises in this paper. In the next section, we review the computational approaches to metaphor generation that are related to the problem just stated.

Computational Approaches to Metaphor Generation

Abe, Sakamoto and Nakagawa (2006) employed a three-layer feedforward neural network to transform adjective-modified nouns, e.g. 'young, innocent, and fine character' into 'A like B' style metaphors, e.g. 'the character is like a child'. The nodes of the input layer correspond to a noun and three adjectives. The nodes of the hidden layer correspond to the latent semantic classes obtained by a probabilistic latent semantic indexing method (Kameya and Sato 2005). A semantic class refers to a set of semantically related words. Activation of the input layer is transferred to the semantic classes (and the words in each class) of the hidden layer. In the output layer, the words that receive most activation (from different semantic classes) become metaphor vehicles. In effect, this method outputs concepts that are the intermediates between the semantic classes to which the input noun and three adjectives are strongly associated. If they are associated to different semantic classes, this method produces irrelevant and hard to visualize vehicles.

A variation of the above model was created by Terai and Nakagawa (2009), who made use of a recurrent neural network to explicitly implement feature interaction. It differs with the previous model at the input layer, where each feature node has bidirectional edge with every other feature node. The performance of these two models was compared in an experiment of generating metaphors for two tenors. The model with feature interaction produced better results.

Besides, Terai and Nakagawa (2010) proposed a method of evaluating the aptness of metaphor vehicles generated by the aforementioned two computational models. A candidate vehicle is judged based on the semantic similarity between the corresponding generated metaphor and the input expression. A candidate vehicle is more apt when the meaning of the corresponding metaphor is closer to the input expression. The semantic similarity is calculated based on the same language model used in the metaphor generation process. The proposed aptness measure was tested in an experiment of generating metaphors for one input expression, which demonstrated that it improved the aptness of generated metaphors.

Veale and Hao (2007) created a system called Sardonicus which can both understand and generate property-attribution metaphors. Sardonicus takes advantage of a knowledge base of entities (nouns) and their most salient properties (adjectives). This knowledge base is acquired from the web using linguistic patterns like 'as ADJ as *' and 'as * as a/an NOUN'. To generate metaphors, Sardonicus searches the knowledge base for nouns that are associated with the intended property. The aptness of the found nouns is assessed according to the category inclusion theory, i.e. "only those noun categories that can meaningfully include the tenor as a member should be considered as potential vehicles". For each found noun, a query in the format 'vehicle-like tenor' is sent through a search engine. If there are more than zero results returned, the noun is considered an apt vehicle. Otherwise, it is considered not apt or extremely novel.

The above reviewed effort of generating metaphor converges at a two-stage approach. These two stages are:

- Stage 1: Search for concepts that are salient in the property to be highlighted
- Stage 2: Evaluate the aptness of the found concepts as metaphor vehicles

This two-stage approach of metaphor generation is adopted by us. We provide methods of searching and evaluating metaphor vehicles, which are different from the literature. In addition, special consideration is given to the aptness of metaphor in the advertising context.

An Approach of Generating Apt Metaphor Ideas for Pictorial Advertisements

We adopt a general two-stage computational approach of metaphor generation (as introduced in the last section) to generate apt metaphor ideas for pictorial advertisements. At the first stage, we look for concepts which have high Imageability (Paivio, Yuille and Madigan 1968; Toglia, and Battig 1978) and the selling premise as one of their prototypical properties. At the second stage, we evaluate the aptness of the candidate vehicles using four metrics, including affect polarity, salience, secondary attributes and

similarity with tenor. Vehicles that are validated by all the four metrics are considered apt for a specific advertising task. In the following sections, we explain the rationale of our approach and its computational details.

Stage 1: Search Candidate Metaphor Vehicles

To find entities which have the selling premise as one of their prototypical properties, our strategy is searching for concepts that are strong semantic associations of the selling premise. One note to mention is that the concepts sought-after do not need to be the 'absolute' associations, because the meaning of a metaphor, i.e. which aspect of the tenor and vehicle becomes prominent, does not only depend on the vehicle, but on the interaction between the tenor and vehicle. In the past, we developed an automatic knowledge extraction system, namely VRAC (Visual Representations for Abstract Concepts), for providing concepts of physical entities to represent abstract concepts (Xiao and Blat 2011). Here we give a brief introduction of this work.

We look for semantic associations in three knowledge bases, includingword association databases (Kiss, Armstrong, Milroy and Piper 1973; Nelson, McEvoy and Schreiber 1998), a commonsense knowledge base called ConceptNet (Liu and Singh 2004) and Roget's Thesaurus (Roget 1852). The reason for using three of them is that we want to take use of the sum of their capacity, in terms of both the vocabulary and the type of content. The nature of these three knowledge bases ensures that the retrieved concepts have close association with the selling premise.

Vehicles of pictorial metaphors should have high imageability, in order to be easily visualized in advertisements. Imageability refers to how easy a piece of text elicits mental image of its referent. It is usually measured in psychological experiments. The available data about word imageability, at the scale of thousands, does not satisfy our need of handling arbitrary words and phrases. As imageability is highly correlated with word concreteness, we developed a method of estimating concreteness using the ontological relations in WordNet (Fellbaum 1998), as an approximation of imageability.

To evaluate the capacity of VRAC, we collected thirty-eight distinct visual representations of six abstract concepts used in past successful advertisements. These abstract concepts have varied parts of speech and word usage frequency. We checked if these visual representations were included in the concepts output by VRAC, with the corresponding abstract concept as input. On average, VRAC achieved a hit rate of 57.8%. The concepts suggested by VRAC are mostly single objects. It lacks the concepts of scenes or emergent cultural symbols, which also play a role in mass visual communication.

Stage 2: Evaluate the Aptness of Candidate Vehicles

The aptness of the candidate vehicles generated in Stage 1 is evaluated based on four metrics, including affect polarity, salience, secondary attributes and similarity with tenor.

Affect Polarity Most of the time, concepts with negative emotions are avoided in advertising (Kohli and Labahn, 1997; Amos, Holmes and Strutton 2008). Even in provocative advertisements, negative concepts are deployed with extreme caution (De Pelsmacker and Van Den Bergh 1996; Vézina and Paul 1997; Andersson, Hedelin, Nilsson and Welander 2004). In fact, negative concepts are often discarded at the first place (Kohli and Labahn 1997). Therefore, we separate candidate vehicles having negative implication from the ones having positive or neutral implication. For this purpose, affective lexicons, which provide affect polarity values of concepts, come in handy. We decided to use SentiWordNet 3.0 (Baccianella, Esuli and Sebastiani 2010), due to its big coverage (56,200 entries) and fine grained values. It provides both the positive and negative valences, which are real values ranging from 0.0 to 1.0. If a candidate vehicle is found in SemtiWordNet 3.0, its affect polarity is calculated by subtracting the negative valence from the positive valence. The candidate vehicles which are not included in SemtiWordNet 3.0 are considered being emotionally neutral.

Salience Salience refers to how strongly a symbol evokes certain meaning in humans' mind. The candidate vehicles found by VRAC have varying association strength with the selling premise, from very strong to less. The vehicle of a metaphor has to be more salient in the intended property than the tenor (Ortony 1979; Glucksberg and Keysar 1990). We interpret salience as a kind of semantic relatedness (Budanitsky and Hirst 2006), which reflects how far two concepts are in the conceptual space of a society. We calculate the semantic relatedness between each candidate vehicle and the selling premise, and between the product and the selling premise. Candidate vehicles that are more remote from the selling premise than the product are discarded. We will talk more about semantic relatedness and the specific measures we used in a later section.

Secondary Attributes Metaphors that capture the appropriate number of relevant features are considered especially apt (Glucksberg and Keysar 1990, 1993; Chiappe and Kennedy 1999). Phillips (1997) found that strong implicatures as well as weak implicatures were drawn from pictorial advertisements. Strong implicatures correspond to the selling premise of an ad, while we use 'secondary attributes' for referring to the weak implicatures. We have not seen literature on the salience of the secondary attributes in metaphor vehicles. We think the candidate vehicles should, at least, not contradict the secondary attributes prescribed to a product. For this end, we use a semantic relatedness measure to filter candidate vehicles that are very distant from the secondary attributes. This is 'soft' filtering, in contrast to the 'hard' filtering used in the previous two metrics, i.e. affect polarity and salience, in the sense that the current criterion might need be tighten in order to ensure the aptness of generated metaphors.

We compare the above approach with an alternative, which is using both the selling premise and the secondary attributes to search for candidate vehicles. This alternative

method indeed looks for concepts that are salient in all these properties. This is possible, but rare. Most of the time, no result will be returned. On the other hand, there is a natural distinction of priority in the attributes (for a product) desired by advertisers (recall the strong and weak implicatures just mentioned). To represent this distinction, weighting of attributes is necessary.

The computational model proposed by Terai and Nakagawa (2009) also uses multiple features to generate metaphors. The weights of the edges connecting the feature nodes in the input layer vary with the tenor. Specifically, the weight of an edge equals to the correlation coefficient between the two features respecting the tenor. The calculation is based on a statistic language model built on a Japanese corpus (Kameya and Sato 2005), which means the weighting of features (of a tenor) is intended to be near reality. However, this idea does not suit advertising, because the features attributed to a product are much more arbitrary. Very often, a product is not thought possessing those features before the appearance of an advertisement.

Similarity with Tenor Good metaphors are those whose tenor and vehicle are not too different yet not too similar to each other (Aristotle 1924; Tourangeau and Sternberg 1981; Marschark, Kats and Paivio 1983). For this reason, we calculate the semantic relatedness between the product and each candidate vehicle. Firstly, candidate vehicles which have zero or negative semantic relatedness values are discarded, because they are considered too dissimilar to the product. Then, the candidate vehicles with positive relatedness values are sorted in the descending order of relatedness. Among this series of values, we look for values that are noticeably different from the next value, i.e. turning points. Turning points divide relatedness values into groups. We use the discrete gradient to measure the change of value, and take the value with the biggest change as the turning point. Candidate vehicles with their relatedness value bigger than or equal to the turning point are abandoned, for being too similar to the tenor. Figure 1 shows the sorted relatedness values between the candidate vehicles and the tenor 'child' in the ad of the National Museum of Science and Technology. The turning point in this graph corresponds to the concept 'head'.

Semantic Relatedness Measures In general, semantic relatedness is measured through distance metrics in certain materialized conceptual space, such as knowledge bases and raw text. A number of semantic relatedness measures have been proposed. Each measure has its own merits and weakness. We employed two different measures in the current work, including PMI-IR (Pointwise Mutual Information and Information Retrieval) (Turney 2001) and LSA through Random Indexing (Kanerva, Kristofersson and Holst 2000). PMI-IR is used to compute salience, because we found it gives more accurate results than other available measures when dealing with concept pairs of high semantic relatedness. The relatedness between the selling premise and candidate vehicles is deemed high. Therefore, we use

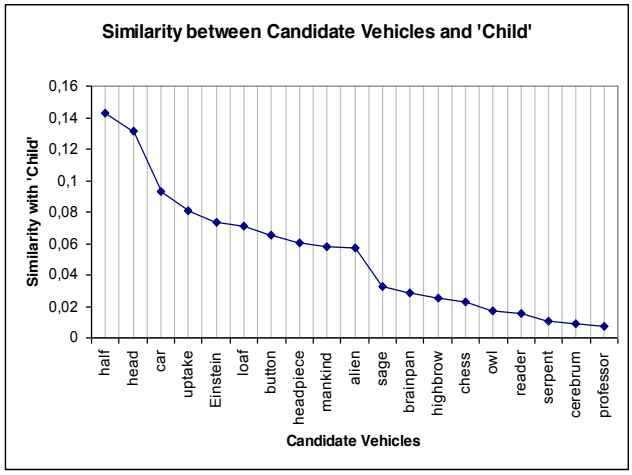


Figure 1: Similarity between candidate vehicles and 'Child'

PMI-IR to give a delicate ordering of their association strength. LSA is employed for the metrics of secondary attributes and similarity with tenor. The motivation behind this choice is to capitalize on LSA's ability of 'indirect inference' (Landauer and Dumais 1997), i.e. discovering connection between terms which do not co-occur. Recall that candidate vehicles are assumed to have strong association with the selling premise, but not necessarily the secondary attributes. In most cases, the association between a candidate vehicle and a secondary attribute is not high. Thus, we need a measure which is sensitive to the low-range semantic relatedness. LSA has demonstrated capacity in this respect (Waltinger, Cramer and Wandmacher 2009). For LSA, values close to 1.0 indicate very similar concepts, while values close to 0.0 and under 0.0 indicate very dissimilar concepts. In our computer program, we utilize the implementation of Random Indexing provided by the Semantic Vectors package[1]. Two-hundred term vectors are acquired from the LSA process for computing semantic relatedness. In the present work, both PMI-IR and LSA are based on the Wikipedia corpus, an online encyclopedia of millions of articles. We obtained the English Wikipedia dumps, offered by the Wikimedia Foundation[2] on October 10th, 2011. The compressed version of this resource is about seven gigabytes.

An Example

We intend to evaluate our approach of generating apt metaphor ideas for pictorial advertisements based on checking whether this approach can reproduce the pictorial metaphors used in past successful advertisements. We have been collecting a number of real ads and the information about the product, selling premise, secondary attributes, and the tenor and vehicle of metaphor in these ads. Nonetheless, it is a tedious process.

[1] http://code.google.com/p/semanticvectors/
[2] http://download.wikipedia.org/

In this paper, we use the information of three real ads to show what our computer program generates. These three ads are for the Volvo S80 car, The Economist newspaper and the National Museum of Science and Technology in Stockholm respectively. Each of them has a pictorial metaphor as its center of expression. All the three ads have the same selling premise: 'intelligence'. However, three different vehicles are used, including 'chess', 'brain' and 'Einstein' respectively. The selection of these particular ads aims at testing whether our aptness metrics are able to differentiate different tenors.

Table 1 summarizes the three aspects of the three ads, including product, secondary attributes and the tenor of metaphor. For both of the car and newspaper ads, the tenors of metaphor are the products. For the museum ad, the tenor is the target consumer, children.

We found the secondary attributes of the Volvo S80 car in its product introduction[3]. For the other two ads, the Economist newspaper and the National Museum of Science and Technology, we have not found any secondary attributes specified. Instead, their subject matter is used to distinguish them from the products of the same categories

Furthermore, we think it is more accurate to use the Boolean operations 'AND' and 'OR' in describing the relation between multiple secondary attributes. As consequence, candidate vehicles have to be reasonably related to both attributes at the two sides of AND; at least one of the two attributes connected by OR.

Product	Secondary Attributes	Tenor
car[4]	elegance AND luxury AND sophisticated	car
newspaper[5]	international politics OR business news	newspaper
museum[6]	science OR technology	child

Table 1: Information about the three real ads

For the concept 'intelligence', VRAC provides eighty-seven candidate vehicles, including single words and phrases. We keep the single-word concepts and extract the core concept of a phrase, in order to reduce the complexity of calculating the aptness metrics at the later stage. An example of the core concept of a phrase is the word 'owl' in the phrase 'wise as an owl'. The core concepts are extracted automatically based on syntactic rules. This process introduces noise, i.e. concepts not related to 'intelligence', such as 'needle' of the phrase 'sharp as a needle' and 'button' of the phrase 'bright as a button'. In total, there are thirty-four candidate vehicles of single words. All the three metaphor vehicles used in the three real ads are included.

[3] http://www.volvocars.com/us/all-cars/volvo-s80/pages/5-things.aspx, retrieved on April 1st, 2012.
[4] http://adsoftheworld.com/media/print/volvo_s80_iq
[5] http://adsoftheworld.com/media/print/the_economist_brain
[6] http://adsoftheworld.com/media/print/the_national_museum_of_science_and_technology_little_einstein

As to affect polarity, the majority of the candidate vehicles, thirty out of thirty-four, are emotionally neutral. Besides, 'highbrow' is marked as positive, while 'geek' and 'serpent' as negative.

The ranking of candidate vehicles by its salience in the selling premise is shown in Table 2. The semantic relatedness calculated by PMI-IR correctly captured the main trend of salience. 'IQ', 'Mensa' and 'brain' are ranked top, while 'needle', 'button' and 'table', which are the noise introduced by the core concept extraction method, are ranked very low. The positions of the products are marked in italic. Only candidate vehicles having higher salience than a product are seen as valid. For instance, 'horse', ranked the twenty-sixth, is not selected for the Volvo S80 car ad, since car is judged as more intelligent than horse by PMI-IR. On the other hand, all the metaphor vehicles used in the original ads, i.e. chess, brain and Einstein, have higher rankings than the corresponding tenors, which supports Ortony's salience imbalance theory.

Rank	Vehicle	Rank	Vehicle
1	IQ	19	reader
2	Mensa	*20*	*child*
3	brain	21	sage
4	computer	22	serpent
5	cerebrum	23	owl
6	alien	*24*	*car*
7	mankind	25	whale
8	highbrow	26	horse
9	Einstein	27	pig
10	head	38	half
11	professor	29	needle
12	dolphin	30	button
13	chess	31	table
14	lecturer	32	uptake
15	geek	33	storey
16	headpiece	34	loaf
17	*newspaper*	35	brainpan
18	atheist	36	latitudinarian

Table 2: Candidate vehicles sorted in the descending order of salience

Table 3 shows how candidate vehicles are filtered by the secondary attributes of products, where candidate vehicles that are not contradictory to the secondary attributes are presented. Table 4 shows the candidate vehicles that are not too different yet not too similar with the tenors of the three ads respectively. For both results, the metaphor vehicles used in the original ads survived the filtering, which gives support to the domain interaction theory proposed by Tourangeau and Sternberg. Nevertheless, there is also flaw in the results produced by the LSA-IR measure. For instance, regarding the fourth column of Table 3, we suspect

'brain' should not have nothing to do with 'science' and consulted several other semantic relatedness measures, which confirmed our skepticism.

Product	car	newspaper	museum
Secondary Attributes	elegance AND luxury AND sophisticated	international politics OR business news	science OR technology
Candidate Vehicle	chess half geek	IQ brain computer cerebrum mankind highbrow head professor dolphin chess lecturer geek headpiece atheist reader sage owl car whale horse half needle button table uptake storey brainpan	IQ Mensa computer cerebrum alien mankind highbrow Einstein head professor chess lecturer headpiece atheist reader sage owl whale half needle button table storey loaf brainpan

Table 3: Candidate vehicles NOT contradictory to the secondary attributes of the three products respectively

Tenor	car	newspaper	child (museum)
Candidate Vehicle	pig storey mankind uptake button half serpent whale lecturer chess latitudinarian sage professor alien horse IQ	professor loaf whale table atheist geek mankind brainpan head Mensa button dolphin brain sage pig headpiece uptake storey	car uptake Einstein loaf button headpiece mankind alien sage brainpan highbrow chess owl reader serpent cerebrum professor

Table 4: Candidate vehicles that are not too different yet not too similar with the tenors of the three ads respectively

We show in Table 5 the metaphor vehicles suggested by our computer program for each of the three ads after applying all the four aptness metrics. For all the three ads, the vehicles used in the original ads are included in the vehicles suggested by our computer program, as marked in italic. For the Volvo S80 car ad, the original metaphor vehicle is the only one recommended by our program. For the other two ads, our program also proposed other five and eight vehicles respectively. Considering that there are thirty four candidate vehicles input to the second stage, we think the four aptness metrics together did an acceptable job.

Regarding the generated vehicles other than the one used in the original ad: are they equally effective? We will have a closer look at the metaphor vehicles generated for the ad of the National Museum of Science and Technology, since it has the most suggested vehicles. It is easy to spot a semantic cluster among these eight vehicles. Five out of eight are humans or human-like entities bearing high intellect, including 'Einstein', 'mankind', 'alien', 'highbrow' and 'professor'. 'Einstein', as the most prototypical within this cluster, fits best this specific advertising task. Besides, other vehicles in this cluster are also highly relevant to a setting like museum for people, especially children, to increase knowledge and encounter inspiration. They may be optimal for other advertising tasks with slightly different focus. The only exception is 'mankind', which is a very general concept. As to the rest of the suggested metaphor vehicles, certain 'headpiece' is possibly kind of symbol of intelligence; playing 'chess' shows someone is intelligent, and 'cerebrum' is strongly associated with intelligence. It is not difficult to imagine a picture of juxtaposing a headpiece and a child, a child playing chess or a child whose cerebrum is emphasized, all of which would be effective to associate a child with intelligence. However, strictly speaking, they are not metaphors.

On the other hand, the existence of candidate vehicles other than the ones used in the original ads may suggest, firstly, our implementation of the four aptness metrics may not sufficiently reduce inapt vehicles. Secondly, more metrics, representing other factors that affect metaphor aptness, may be necessary.

Ad	**Tenor**	**Vehicle**
Volvo S80 car	car	*chess*
The Economist newspaper	newspaper	professor
		mankind
		head
		dolphin
		brain
		headpiece
National Museum of Science and Technology	child	*Einstein*
		headpiece
		mankind
		alien
		highbrow

		chess
		cerebrum
		professor

Table 5: Metaphor vehicles considered apt for the three ads respectively

Conclusions

In the work presented in this paper, we adopted a general two-stage computational approach to generate apt metaphor ideas for pictorial advertisements. The first stage looks for concepts which have high imageability and the selling premise as one of their prototypical properties. The second stage evaluates the aptness of the candidate vehicles (found in the first stage) with regard to four aspects, including affect polarity, salience, secondary attributes and similarity with tenor. These four metric are conceived based on the general characteristics of metaphor and its specialty in advertising. For the first stage, we developed an automatic knowledge extraction method to find concepts of physical entities which are strongly associated with the selling premise. For the second stage, we utilized an affect lexicon and two semantic relatedness measures to implement the four aptness metrics. The capacity of our computer program is demonstrated in a task of reproducing the pictorial metaphors used in three real advertisements. All the three original metaphors were replicated, as well as a few other vehicles recommended, which, we consider, would make effective advertisements, though less optimal. In short, our approach and implementation are promising in generating diverse and apt pictorial metaphors for advertisements.

On the other hand, to have a more critical view of our approach and implementation, larger scale evaluation is in need. Continuing the evaluation design introduced in this paper, more examples of pictorial metaphors used in real advertisements have to be collected and annotated. This corpus would not only contribute to building our metaphor generator, but also be an asset for the research on metaphor and creativity in general.

Moreover, the results provided by our aptness metrics support both the salience imbalance theory and the domain interaction theory.

Future Work

We intend to compute more ways of expression appeared in pictorial advertisements. Firstly, our current implementation can be readily adapted to generate visual puns. In a pun, the product (or something associated to it) also has the meaning of the selling premise. An example is an existing ad which uses the picture of an owl to convey the message 'zoo is a place to learn and gain wisdom'. As we all know, owl is both a member of the zoo and a symbol of wisdom. Secondly, we found some other fields of study are very relevant to computing advertising expression, such as the research and computational modeling of humor (Raskin 1985; Attardo and Raskin 1991; Ritchie 2001; Binsted, Bergen, Coulson, Nijholt, Stock, Strapparava, Ritchie, Manurung, Pain, Waller and O'Mara, 2006). Finally, we are especially interested in investigating hyperbole. Hyperbole has nearly universal presence in advertisements, but its theoretic construction and computational modeling are minimal. There exist some ad-hoc approaches: for instance, we can find the exaggeration of the selling proposition by the AlsoSee relation in WordNet; or, we should first think about a cognitive or linguistic model of hyperbole instead.

References

Abe, K., Sakamoto, K., and Nakagawa, M. 2006. A computational model of metaphor generation process. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, 937–942.

Amos, C., Holmes, G., and Strutton, D. 2008. Exploring the relationship between celebrity endorser effects and advertising effectiveness: A quantitative synthesis of effect size. *International Journal of Advertising* 27(2):209–234.

Andersson, S., Hedelin, A., Nilsson, A., and Welander, C. 2004. Violent advertising in fashion marketing. *Journal of Fashion Marketing and Management* 8:96-112.

Aristotle. 1924. *The Art of Rhetoric*. Translated by W. Rhys Roberts. The Internet Classics Archive, accessed October 1, 2012. http://classics.mit.edu//Aristotle/rhetoric.html.

Attardo, S., and Raskin, V. 1991. Script theory revis(it)ed: Joke similarity and joke representation model. *Humor: International Journal of Humor Research* 4(3-4):293-347.

Baccianella, S., Esuli, A., and Sebastiani, F. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC'10)*, Valletta, MT.

Binsted, K., Bergen, B., Coulson, S., Nijholt, A., Stock, O., Strapparava, C., Ritchie, G., Manurung, R., Pain, H., Waller, A., and O'Mara, D. 2006. Computational humor. *IEEE Intelligent Systems* March-April.

Black, M. 1962. *Models and metaphors*. NY: Cornell University Press.

Budanitsky, A., and Hirst, G. 2006. Evaluating WordNet-based measures of semantic distance. *Journal of Computational Linguistics* 32(1):13–47.

Chiappe, D., and Kennedy, J. 1999. Aptness predicts preference for metaphors or similes, as well as recall bias. *Psychonomic Bulletin and Review* 6:668–676.

De Pelsmacker, P., and Van Den Bergh, J. 1996. The communication effects of provocation in print advertising. *International Journal of Advertising* 15(3):203–22.

Fellbaum, C. D., ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

Forceville, C. 1996. *Pictorial Metaphor in Advertising*. London: Routledge.

Foss, S. K. 2005. Theory of Visual Rhetoric. In Smith, K., Moriarty, S., Barbatsis, G., and Kenney, K., eds., *Hand- book of Visual Communication: Theory, Methods, and Media*. Mahwah, New Jersey: Lawrence Erlbaum. 141-152.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7:155-170.

Gentner, D., and Clements, C. 1988. Evidence for relational selectivity in the interpretation of analogy and metaphor. In Bower, G., ed., *The Psychology of Learning and Motivation*, Vol. 22. Orlando, FL: Academic Press.

Gentner, D., and Markman, A. B. 1997. Structure mapping in analogy and similarity. *American Psychologist* 52:45-56.

Glucksberg, S., and Keysar, B. 1990. Understanding Metaphorical comparisons: beyond similarity. *Psychological Review* 97:3-18.

Glucksberg, S., and Keysar, B. 1993. How metaphors work. In Ortony, A. ed., *Metaphor and Thought*, 2nd ed. Cambridge: Cambridge University Press. 401- 424.

Goldenberg, J., Mazursky, D., and Solomon, S. 1999. Creativity templates: towards identifying the fundamental schemes of quality advertisements. *Marketing Science* 18(3):333-351.

Kameya, Y., and Sato, T. 2005. Computation of probabilistic relationship between concepts and their attributes using a statistical analysis of Japanese corpora. In *Proceedings of Symposium on Large-Scale Knowledge Resources*, 65-68.

Kanerva, P., Kristofersson, J., and Holst, A. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society (CogSci'00)*, Erlbaum.

Kiss, G. R., Armstrong, C., Milroy, R., and Piper, J. 1973. An associative thesaurus of English and its computer analysis. In Aitken, A. J., Bailey, R. W., and Hamilton-Smith, N. eds., *The Computer and Literary Studies*, Edinburgh: University Press. 153-165.

Kohli, C., and Labahn, D. W. 1997. Creating effective brand names: A study of the naming process. *Journal of Advertising Research* 37(1):67–75.

Landauer, T. K., and Dumais, S. T. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* 104:211-240.

Liu, H., and Singh, P. 2004. ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal* 22(4):211-26.

Maes, A., and Schilperoord, J. 2008. Classifying visual rhetoric: Conceptual and structural heuristics. In McQuarrie, E. F., and Phillips, B. J., eds., *Go Figure: New Directions in Advertising Rhetoric*. Armonk, New York: Sharpe. 227-257.

Marschark, M., Katz, A. N., and Paivio, A. Dimensions of metaphor, *Journal of Psycholinguistic Research* 12(1):17.

Nelson, D. L., McEvoy, C. L., and Schreiber, T. A. 1998. The university of south Florida word association, rhyme, and word fragment norms. *University of South Florida Website*, accessed January 13, 2012. http://www.usf.edu/FreeAssociation/.

Ortony, A. 1979. Beyond literal similarity. *Psychological Review* 86:161-180.

Paivio, A., Yuille, J. C., and Madigan, S. 1968. Concreteness, imagery, and meaningfulness values of 925 nouns. *Journal of Experimental Psychology* 76:1-25.

Phillips, B. J. 1997. Thinking into it: Consumer interpretation of complex advertising images. *Journal of Advertising* 26(2):77–87.

Phillips, B. J., and McQuarrie, E. F. 2004. Beyond visual metaphor: A new typology of visual rhetoric in advertising. *Marketing Theory* 4(1/2):111-134.

Phillips, B. J., and McQuarrie, E. F. 2009. Impact of advertising metaphor on consumer beliefs: delineating the contribution of comparison versus deviation factors. *Journal of Advertising* 38(1):49-61.

Raskin, V. 1985. *Semantic Mechanisms of Humor*. Dordrecht, Boston, Lancaster: D. Reidel Publishing Company.

Richards, I. A. 1936. *The Philosophy of Rhetoric*. London: Oxford University Press.

Ritchie, G. D. 2001. Current directions in computational humour. *Artificial Intelligence Review* 16(2):119-135.

Roget, P. M. 1852. *Roget's Thesaurus of English Words and Phrases*. Harlow, Essex, England: Longman Group Ltd.

Terai, A., and Nakagawa, M. 2009. A neural network model of metaphor generation with dynamic interaction. In Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G., eds., *ICANN 2009*, Part I. LNCS, 5768, Springer, Heidelberg . 779–788.

Terai, A., and Nakagawa, M. 2010. A computational system of metaphor generation with evaluation mechanism. In Diamantaras, K., Duch, W., Iliadis, L. S., eds., *ICANN 2010*, Part II. LNCS, 6353, Springer, Heidelberg. 142–147.

Toglia, M. P., and Battig, W. F. 1978. *Handbook of Semantic Word Norms*. Hillsdale, NJ: Erlbaum.

Tourangeau, R., and Sternberg, R. J. 1981. Aptness in metaphor. *Cognitive Psychology* 13: 27-55.

Tourangeau, R., and Sternberg, R. J. 1982. Understanding and appreciating metaphors. *Cognition* 11: 203-244.

Turney, P. D. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning*, 491-502. Berlin: Springer-Verlag.

Veale, T., and Keane, M. T. 1992. Conceptual Scaffolding: A spatially-founded meaning representation for metaphor comprehension. *Computational Intelligence* 8(3):494-519.

Veale, T., O Donoghue, D., and Keane, M. T. 1995. Epistemological issues in metaphor comprehension: A comparative analysis of three models of metaphor interpretation. In *Proceedings of ICLC'95, the Fourth Conference of The International Cognitive Linguistics Association*, Albuquerque NM.

Veale, T., and Hao, Y. 2007. Comprehending and generating apt metaphors: A Web-driven, case-based approach to figurative language. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, Vancouver, Canada.

Vézina, R., and Paul, O. 1997. Provocation in advertising: A conceptualization and an empirical assessment. *International Journal of Research in Marketing* 14(2):177-192.

Waltinger, U., Cramer, I., and Wandmacher, T. 2009. From social networks to distributional properties: A comparative study on computing semantic relatedness. In *Proceedings of the Thirty-First Annual Meeting of the Cognitive Science Society, CogSci 2009*, 3016- 3021. Cognitive Science Society, Amsterdam, Netherlands.

Xiao, P., and Blat, J. 2012. Image the imageless: Search for pictorial representations of abstract concepts. In *Proceedings of the Seventh International Conference on Design Principles and Practices*, Chiba, Japan.

Once More, With *Feeling!*
Using Creative Affective Metaphors to Express Information Needs

Tony Veale
Web Science & Technology Division, KAIST / School of Computer Science and Informatics, UCD
Korean Advanced Institute of Science & Technology, South Korea / University College Dublin, Ireland.
Tony.Veale@gmail.com

Abstract

Creative metaphors abound in language because they facilitate communication that is memorable, effective and elastic. Such metaphors allow a speaker to be maximally suggestive while being minimally committed to any single interpretation, so they can both supply and elicit information in a conversation. Yet, though metaphors are often used to articulate affective viewpoints and information needs in everyday language, they are rarely used in information retrieval (IR) queries. IR fails to distinguish between creative and uncreative uses of words, since it typically treats words as *literal mentions* rather than *suggestive allusions*. We show here how a computational model of affective comprehension and generation allows IR users to express their information needs with creative metaphors that concisely allude to a dense body of assertions. The key to this approach is a lexicon of stereotypical concepts and their affective properties. We show how such a lexicon is harvested from the open web and from local web n-grams.

Creative Truths

Picasso famously claimed that "art is a lie that tells the truth." Fittingly, this artful contradiction suggests a compelling reason for why speakers are so wont to use artfully suggestive forms of creative language – such as metaphor and irony – when less ambiguous and more direct forms are available. While literal language commits a speaker to a tightly fixed meaning, and offers little scope to the listener to contribute to the joint construction of meaning, creative language suggests a looser but potentially richer meaning that is amenable to collaborative elaboration by each participant in a conversation.

A metaphor *X is Y* establishes a *conceptual pact* between speaker and listener (Brennan & Clark, 1996), one that says 'let us agree to speak of X using the language and norms of Y' (Hanks, 2006). Suppose a speaker asserts that "X is a snake". Here, the stereotype "snake" conveys the speaker's negative stance toward X, and suggests a range of talking points for X, such as that X is *charming* and *clever* but also *dangerous*, and is not to be *trusted* (Veale & Hao, 2008). A listener may now respond by elaborating the metaphor, even when disagreeing with the basic conceit, as in "I agree that X can be charming, but I see no reason to distrust him". Successive elaboration thus allows a speaker and listener to arrive at a mutually acceptable construal of a metaphorical "snake" in the context of X.

Metaphors achieve a balance of suggestiveness and concision through the use of *dense descriptors*, familiar terms like "snake" that evoke a rich variety of stereotypical properties and behaviors (Fishelov, 1992). Though every concept has the potential to be used creatively, casual metaphors tend to draw their dense descriptors from a large pool of familiar stereotypes shared by all speakers of a language (Taylor, 1954). A richer, more conceptual model of the lexicon is needed to allow any creative uses of stereotypes to be inferred as needed in context. We will show here how a large lexicon of stereotypes is mined from the web, and how stereotypical representations can be used selectively and creatively, to highlight relevant aspects of a given target concept in a specific metaphor.

Because so many familiar stereotypes have polarizing qualities – think of the endearing and not-so-endearing qualities of babies, for instance – metaphors are ideal vehicles for conveying an affective stance toward a topic. Even stereotypes that are not used figuratively, as in the claim "Steve Jobs was a great leader", are likely to elicit metaphors in response, such as "yes, a true pioneer" or "what an artist!", or even "but he could be such a tyrant!". Proper-names can also be used as evocative stereotypes, as when Steve Jobs is compared to the fictional inventor *Tony Stark*, or Apple is compared to *Scientology*, or Google to *Microsoft*. We use stereotypes effortlessly, and their exploitations are common currency in everyday language.

Information retrieval, however, is a language-driven application where the currency of metaphor has little or no exchange value, not least because IR fails to discriminate literal from non-literal language (Veale 2004, 2011, 2012). Speakers use metaphor to provide and elicit information in

casual conversation, but IR reduces any metaphoric query to literal keywords and key-phrases, which are matched near-identically in texts (Salton, 1968; Van Rijsbergen 1979). Yet everyday language shows that metaphor is an ideal form for expressing our information needs. A query like "Steve Jobs as a good leader" can be viewed by an IR system as a request to consider all the ways in which leaders are stereotypically good, and to then consider all the metaphors that are typically used to convey these viewpoints. The IR staple of query expansion (Vernimb, 1977; Vorhees, 1994,1998; Navigli & Velardi, 2003; Xu & Croft, 1996) can be made both *affect-driven* and *metaphor-aware*. In this paper we show how an affective stereotype-based lexicon can both comprehend and generate affective metaphors that capture or shape a user's feelings, and show how this capability can lead to more creative forms of IR.

Related Work and Ideas

Metaphor has been studied within computer science for four decades, yet it remains at the periphery of NLP research. The reasons for this marginalization are, for the most part, pragmatic ones, since metaphors can be as varied and challenging as human creativity will allow. The greatest success has been achieved by focusing on conventional metaphors (e.g., Martin, 1990; Mason, 2004), or on very specific domains of usage, such as figurative descriptions of mental states (e.g., Barden, 2006).

From the earliest computational forays, it has been recognized that metaphor is essentially a problem of knowledge representation. Semantic representations are typically designed for well-behaved mappings of words to meanings – what Hanks (2006) calls *norms* – but metaphor requires a system of soft preferences rather than hard (and brittle) constraints. Wilks (1978) thus proposed his *preference semantics* model, which Fass (1991,1997) extended into a *collative semantics*. In contrast, Way (1990) argues that metaphor requires a dynamic concept hierarchy that can stretch to meet the norm-bending demands of figurative ideation, though her approach lacks computational substance.

More recently, some success has been obtained with statistical approaches that side-step the problems of knowledge representation, by working instead with implied or latent representations that are derived from word distributions. Turney and Littman (2005) show how a statistical model of relational similarity can be constructed from web texts for retrieving the correct answer to proportional analogies, of the kind used in SAT tests. No hand-coded knowledge is employed, yet Turney and Littman's system achieves an average human grade on a set of 376 real SAT analogies.

Shutova (2010) annotates verbal metaphors in corpora (such as "to *stir* excitement", where "stir" is used metaphorically) with the corresponding conceptual metaphors identified by Lakoff and Johnson (1980). Statistical clustering techniques are then used to generalize from the annotated exemplars, allowing the system to recognize and retrieve other metaphors in the same vein (e.g. "he *swallowed* his anger"). These clusters can also be analyzed to identify literal paraphrases for a metaphor (such as "to *provoke* excitement" or "*suppress* anger"). Shutova's approach is noteworthy for operating with Lakoff & Johnson's inventory of conceptual metaphors without using an explicit knowledge representation.

Hanks (2006) argues that metaphors exploit distributional norms: to understand a metaphor, one must first recognize the norm that is exploited. Common norms in language are the preferred semantic arguments of verbs, as well as idioms, clichés and other multi-word expressions. Veale and Hao (2007a) suggest that stereotypes are conceptual norms that are found in many figurative expressions, and note that stereotypes and similes enjoy a symbiotic relationship that has some obvious computational advantages. Similes use stereotypes to illustrate the qualities ascribed to a topic, while stereotypes are often promulgated via proverbial similes (Taylor, 1954). Veale and Hao (2007a) show how stereotypical knowledge can be acquired by harvesting "Hearst" patterns of the form "as P as C" (e.g. "*as smooth as silk*") from the web (Hearst, 1992). They show in (2007b) how this body of stereotypes can be used in a web-based model of metaphor generation and comprehension.

Veale (2011) employs stereotypes as the basis of a new *creative information retrieval* paradigm, by introducing a variety of non-literal wildcards in the vein of Mihalcea (2002). In this system, **@Noun** matches any adjective that denotes a stereotypical property of *Noun* (so e.g. *@knife* matches *sharp*, *cold*, etc.) while **@Adj** matches any noun for which *Adj* is stereotypical (e.g. *@sharp* matches *sword, laser, razor*, etc.). In addition, **?Adj** matches any property or behavior that co-occurs with, and reinforces, the property denoted by *Adj*; thus, *?hot* matches *humid, sultry* and *spicy*. Likewise, *?Noun* matches any noun that denotes a pragmatic neighbor of *Noun*, where two words are neighbors if they are seen to be clustered in the same ad-hoc set (Hanks, 2005), such as "lawyers and doctors" or "pirates and thieves". The knowledge needed for **@** is obtained by mining text from the open web, while that for **?** is obtained by mining ad-hoc sets from Google n-grams.

There are a number of shortcomings to this approach. For one, Veale (2011) does not adequately model the affective profile of either stereotypes or their properties. For another, the stereotype lexicon is static, and focuses primarily on adjectival properties (like *sharp* and *hot*). It thus lacks knowledge of everyday verbal behaviors like *cutting*, *crying*, *swaggering*, etc. So we build here on the work of Veale (2011) in several important ways.

First, we enrich and enlarge the stereotype lexicon, to include more stereotypes and behaviors. We determine an affective polarity for each property or behavior and for

each stereotype, and show how polarized +/- viewpoints on a topic can be calculated on the fly. We show how proxy representations for ad-hoc proper-named stereotypes (like *Microsoft*) can be constructed on demand. Finally, we show how metaphors are mined from the Google n-grams, to allow the system to understand novel metaphors (like *Google is another Microsoft* or *Apple is a cult*) as well as to generate plausible metaphors for users' affective information needs (e.g., Steve Jobs was a *great leader*, Google is *too powerful*, etc.).

Once more, with *feeling!*

If a property or behavior P is stereotypical of a concept C, we should expect to frequently observe P in instances of C. In linguistic terms, we can expect to see collocations of "P" and "C" in a resource like the Google n-grams (Brants and Franz, 2006). Consider these 3-grams for "cowboy" (numbers in parentheses are Google database frequencies).

a	lonesome	cowboy	432
a	mounted	cowboy	122
a	grizzled	cowboy	74
a	swaggering	cowboy	68

N-gram patterns of the above form allow us to find frequent ascriptions of a quality to a noun-concept, but frequently observed qualities are not always noteworthy qualities (e.g., see Almuhareb and Poesio, 2004,2005). However, if we also observe these qualities in similes – such as "*swaggering like a cowboy*" or "*as grizzled as a cowboy*" – this suggests that speakers see these as typical enough to anchor a figurative comparison. So for each hypothesis *P is stereotypical of C* that we derive from the Google n-grams, we generate the corresponding simile form: we use the "like" form for verbal behaviors such as "swaggering", and the "as-as" form for adjectival properties such as "lonesome". We then dispatch each simile as a phrasal query to Google: a hypothesis is validated if the corresponding simile is found on the web.

This mining process gives us over 200,000 validated hypotheses for our stereotype lexicon. We now filter these hypotheses manually, to ensure that the contents of the lexicon are of the highest quality (investing just weeks of labor produces a very reliable resource; see Veale 2012 for more detail). We obtain rich descriptions for commonplace ideas, such as the dense descriptor *Baby*, whose 163 highly salient qualities – a set denoted *typical*(Baby) – includes *crying*, *drooling* and *guileless*. After this manual phase, the stereotype lexicon maps 9,479 stereotypes to a set of 7,898 properties / behaviors, to yield more than 75,000 pairings.

Determining Nuanced Affect

To understand the affective uses of a property or behavior, we employ the intuition that those which reinforce each other in a single description (e.g. "as *lush and green* as a jungle" or "as *hot and humid* as a sauna") are more likely to have the same affect than those which do not. To construct a support graph of mutually reinforcing properties, we gather all Google 3-grams in which a pair of stereotypical properties or behaviors X and Y are linked via coordination, as in "*hot and spicy*" or "*kicking and screaming*". A bidirectional link between X and Y is added to the graph if one or more stereotypes in the lexicon contain both X and Y. If this is not so, we consider whether both descriptors ever reinforce each other in web similes, by posing the web query "*as X and Y as*". If this query has a non-zero hit set, we still add a link between X and Y.

Next, we build a reference set **-R** of typically negative words, and a disjoint set **+R** of typically positive words. Given a few seed members for **-R** (such as *sad*, *evil*, *monster*, etc.) and a few seed members for **+R** (such as *happy*, *wonderful*, *hero*, etc.), we use the **?** operator of Veale (2011) to successively expand this set by suggesting neighboring words of the same affect (e.g., "sad and *pathetic*", "happy and *healthy*"). After three iterations in this fashion, we populate **+R** and **-R** with approx. 2000 words each. If we can anchor enough nodes in the graph with + or – labels, we can interpolate a nuanced positive / negative score for all nodes in the graph. Let N(p) denote the set of neighboring terms to a property or behavior p in the support graph. Now, we define:

$$(1)\quad N^{+}(p) = N(p) \cap \mathbf{+R}$$

$$(2)\quad N^{-}(p) = N(p) \cap \mathbf{-R}$$

We assign positive / negative affect scores to p as follows:

$$(3)\quad pos(p) = \frac{|N^{+}(p)|}{|N^{+}(p) \cup N^{-}(p)|}$$

$$(4)\quad neg(p) = 1 - pos(p)$$

Thus, *pos(p)* estimates the probability that p is used in a positive context, while *neg(p)* estimates the probability that p is used in a negative context. The *X and Y* 3-grams approximate these contexts for us.

Now, if a term S denotes a stereotypical idea that is described in the lexicon with the set of typical properties and behaviors denoted *typical(S)*, then:

$$(5)\quad pos(S) = \frac{\sum_{p \in typical(S)} pos(p)}{|typical(S)|}$$

$$(6)\quad neg(S) = 1 - pos(S)$$

So we simply calculate the mean affect of the properties and behaviors of s, as represented in the lexicon via *typical(s)*. Note that (5) and (6) are simply gross defaults.

One can always use (3) and (4) to separate the elements of *typical(s)* into those which are more negative than positive (a negative spin on *s*) and those which are more positive than negative (a positive spin on *s*). Thus, we define:

(7) posTypical(S) = $\{p \in \text{typical}(S) \mid \text{pos}(p) > \text{neg}(p)\}$

(8) negTypical(S) = $\{p \in \text{typical}(S) \mid \text{neg}(p) > \text{pos}(p)\}$

For instance, the positive stereotype of *Baby* contains the qualities such as *smiling*, *adorable* and *cute*, while the negative stereotype contains qualities such as *crying*, *wailing* and *sniveling*. As we'll see next, this ability to affectively "spin" a stereotype is key to automatically generating affective metaphors on demand.

Generating Affective Metaphors, N-gram style

The Google n-grams is also a rich source of copula metaphors of the form *Target is Source*, such as "politicians are crooks", "Apple is a cult", "racism is a disease" and "Steve Jobs is a god". Let *src(T)* denote the set of stereotypes that are commonly used to describe T, where commonality is defined as the presence of the corresponding copula metaphor in the Google n-grams. To also find metaphors for proper-named entities like "Bill Gates", we analyse n-grams of the form *stereotype First [Middle] Last*, such as "*tyrant* Adolf Hitler". For example:

src(racism) = {*problem, disease, joke, sin, poison, crime, ideology, weapon*}

src(Hitler) = {*monster, criminal, tyrant, idiot, madman, vegetarian, racist, ...*}

We do not try to discriminate literal from non-literal assertions, nor indeed do we try to define literality at all. Rather, we assume each putative metaphor offers a potentially useful perspective on a topic T.

Let *srcTypical*(T) denote the aggregation of all properties ascribable to T via metaphors in *src*(T):

(9) $srcTypical(T) = \bigcup_{M \in src(T)} typical(M)$

We can also use the *posTypical* and *negTypical* variants of (7) and (8) to focus only on metaphors that place a positive or negatve spin on a topic T. In effect, (9) provides a feature representation for topic T as viewed through the creative lens of metaphor. This is useful when the source *S* in the metaphor *T is S* is not a stereotype in the lexicon, as happens when one describes *Rasputin as Karl Rove*, or *Apple as Scientology*. When the set *typical*(S) is empty, *srcTypical*(*S*) may not be, so *srcTypical*(*S*) can act as a proxy representation for S in these cases.

The properties and behaviors that are salient to the interpretation of *T is S* are given by:

(10) $salient(T,S) = [srcTypical(T) \cup typical(T)] \cap [srcTypical(S) \cup typical(S)]$

In the context of *T is S*, the metaphorical stereotype $M \in src(S) \cup src(T) \cup \{S\}$ is an apt vehicle for T if:

(11) $apt(M, T,S) = |salient(T,S) \cap typical(M)| > 0$

and the degree to which M is apt for T is given by:

(12) $aptness(M,T,S) = \dfrac{|salient(T, S) \cap typical(M)|}{|typical(M)|}$

We can now construct an interpretation for *T is S* by considering the stereotypes in *src*(T) that are apt for T in the context of *T is S*, and by also considering the stereotypes that are commonly used to describe S that are also potentially apt for T:

(13) $interpretation(T, S) = \{M \in src(S) \cup src(T) \cup \{S\} \mid apt(M, T, S)\}$

In effect, the interpretation of the creative metaphor *T is S* is itself a set of more conventional metaphors that are apt for T and which expand upon S. The elements $\{M_i\}$ of *interpretation*(T, S) can be sorted by $aptness(M_i\ T,S)$ to produce a ranked list of interpretations ($M_1 \ldots M_n$). For a given interpretation M, the salient features of M are thus:

(14) $salient(M, T,S) = typical(M) \cap salient(T,S)$

So if *T is S* is a creative IR query – to find documents in which T is viewed as S – then *interpretation*(T, S) is an expansion of *T is S* that includes the common metaphors that are consistent with T viewed as S. In turn, for any viewpoint M_i in *interpretation*(T, S), then $salient(M_i, T, S)$ is an expansion of M_i that includes all of the qualities that T is likely to exhibit when it behaves like M_i.

A Worked Example: Metaphor Generation for IR

Consider the creative query "*Google is Microsoft*", which expresses a user's need to find documents in which Google exhibits qualities typically associated with Microsoft. Now, both *Google* and *Microsoft* are complex concepts, so there are many ways in which they can be considered similar or dissimilar, whether in a good or a bad light. However, we can expect the most salient aspects of Microsoft to be those that underpin our common metaphors for Microsoft, i.e., the stereotypes in *src*(Microsoft). These metaphors will provide the talking points for an interpretation.

The Google n-grams yield up the following metaphors, 57 for Microsoft and 50 for Google:

src(Microsoft) = {*king, master, threat, bully, giant, leader, monopoly, dinosaur …*}

src(Google) = {*king, engine, threat, brand, giant, leader, celebrity, religion …*}

So the following qualities are aggregrated for each:

srcTypical(Microsoft) = {*trusted, menacing, ruling, threatening, overbearing, admired, commanding, …*}

srcTypical(Google) = {*trusted, admired, reigning, lurking, crowned, shining, ruling, determined, …*}

Now, the salient qualities highlighted by the metaphor, namely *salient*(Google, Microsoft), are:

{*celebrated, menacing, trusted, challenging, established, threatening, admired, respected, …*}

Finally, *interpretation*(Google,Microsoft) contains:

{*king, criminal, master, leader, bully, threatening, giant, threat, monopoly, pioneer, dinosaur, …*}

Let's focus on the expansion "Google is king", since according to (12), *aptness*(*king*, Google, Microsoft) = 0.48 and this is the highest ranked element of the interpretation. Now, *salient*(*king*, Google, Microsoft) contains:

{*celebrated, revered, admired, respected, ruling, arrogant, commanding, overbearing, reigning, …*}

Note that these properties / behaviours are already implicit in our consensus perception of Google, insofar as they are highly salient aspects of the stereotypical concepts to which Google is frequently compared on the web. These properties / behaviours can now be used to perform query expansion for the query term "Google", to find documents where the system believes Google is acting like Microsoft.

The metaphor "Google is Microsoft" is diffuse and lacks an affective stance. So let's consider instead the metaphor "Google is **-Microsoft**", where **-** is used to impart a negative spin (and where + can likewise impart a positive spin). In this case, *negTypical* is used in place of *typical* in (9) and (10), so that:

srcTypical(**-Microsoft**) =

{*menacing, threatening, twisted, raging, feared, sinister, lurking, domineering, overbearing, …*}

and

salient(Google, **-Microsoft**) =

{*menacing, bullying, roaring, dreaded…*}

Now, *interpretation*(Google, **-Microsoft**) becomes:

{*criminal, giant, threat, bully, evil, victim, devil, …}*

In contrast, *interpretation*(Google, +**Microsoft**) is:

{*king, master, leader, pioneer, classic, partner, …}*

More focus is achieved with this query in the form of a simile: "Google is as **-powerful** as Microsoft". For explicit similes, we need to focus on just a sub-set of salient properties, as in this varient of (10):

$\{p \in salient(\text{Google, Microsoft}) \mid p \in \text{N}^{-}(\text{powerful})\}$

In this case, the final interpretation becomes:

{*bully, threat, giant, devil, monopoly, dinosaur, …}*

A few simple concepts can thus yield a wide range of options for the creative IR user who is willing to build queries around affective metaphors and similes.

Empirical Evaluation

The affective stereotype lexicon is the cornerstone of the current approach, and must reliably assign meaningful polarity scores both to properties and to the stereotypes that exemplify them. Our affect model is simple in that it relies principally on +/- affect, but as demonstrated above, users can articulate their own expressive moods to suit their needs: for Stereotypical example, one can express disdain for too much power with the term **-powerful**, or express admiration for guile with **+cunning** and **+devious**.

The Effect of Affect: Stereotypes and Properties

Note that the polarity scores assigned to a property *p* in (3) and (4) do not rely on any prior classification of *p*, such as whether *p* is in **+R** or **-R**. That is, **+R** and **-R** are not used as training data, and (3) and (4) receive no error feedback. Of course, we expect that for $p \in$ **+R** that *pos*(p) > *neg*(p), and for $p \in$ **-R** that *neg*(p) > *pos*(p), but (3) and (4) do not iterate until this is so. Measuring the extent to which these simple intuitions are validated thus offers a good evaluation of our graph-based affect mechanism.

Just five properties in **+R** (approx. **0.4**% of the 1,314 properties in **+R**) are given a positivity of less than 0.5 using (3), leading those words to be misclassified as more negative than positive. The misclassified property words are: *evanescent, giggling, licking, devotional* and *fraternal.*

Just twenty-six properties in **-R** (approx. **1.9**% of the 1,385 properties in **-R**) are assigned a negativity of less than 0.5 via (4), leading these to be misclassified as more positive than negative. The misclassified words are: *cocky, dense, demanding, urgent, acute, unavoidable, critical, startling, gaudy, decadent, biting, controversial, peculiar, disinterested, strict, visceral, feared, opinionated, humbling, subdued, impetuous, shooting, acerbic, heartrending, ineluctable* and *groveling*.

Because **+R** and **-R** have been populated with words that have been chosen for their perceived +/- slants, this result is hardly surprising. Nonetheless, it does validate the key intuitions that underpin (3) and (4) – that the affective polarity of a property *p* can be reliably estimated as a simple function of the affect of the co-descriptors with which it is most commonly used in descriptive contexts.

The sets **+R** and **-R** are populated with adjectives, verbal behaviors and nouns. **+R** contains 478 nouns denoting positive stereotypes (such as *saint* and *hero*) while **-R** contains 677 nouns denoting negative stereotypes (such as *tyrant* and *monster*). When these reference stereotypes are used to test the effectiveness of (5) and (6) – and thus, indirectly, of (3) and (4) and of the stereotype lexicon itself – **96.7%** of the positive stereotype exemplars are correctly assigned a mean positivity of more than 0.5 (so, *pos(S) > neg(S)*) and **96.2%** of the negative exemplars are correctly assigned a mean negativity of more than 0.5 (so, *neg(S) > pos(S)*). Though it may seem crude to assess the affect of a stereotype as the mean of the affect of its properties, this does appear to be a reliable measure of polar affect.

The Representational Adequacy of Metaphors

We have argued that metaphors can provide a collective representation of a concept that has no other representation in a system. But how good a proxy is *src*(S) or *srcTypical*(S) for an *S* like *Karl Rove* or *Microsoft*? Can we reliably estimate the +/- polarity of *S* as a function of *src*(S)? We can estimate these from metaphors as follows:

$$(15)\quad \text{pos}(S) = \frac{\sum_{M \in src(S)} pos(M)}{|src(S)|}$$

$$(16)\quad \text{neg}(S) = \frac{\sum_{M \in src(S)} neg(M)}{|src(S)|}$$

Testing this estimator on the exemplar stereotypes in **+R** and **-R**, the correct polarity (+ or -) is estimated **87.2**% of the time. Metaphors in the Google n-grams are thus broadly consistent with our perceptions of whether a topic is positively or negatively slanted.

When we consider all stereotypes *S* for which $|src(S)| > 0$ (there are 6,904 in the lexicon), *srcTypical*(S) covers, on average, just **65.7**% of the typical properties of S (that is, of *typical*(S)). Nonetheless, this shortfall is precisely why we use novel metaphors. Consider this variant of (9) which captures the longer reach of these novel metaphors:

$$(17)\quad srcTypical^2(T) = \bigcup_{S \in src(T)} srcTypical(S)$$

Thus, $srcTypical^2(T)$ denotes the set of qualities that are ascribable to T via the expansive interpretation of all metaphors *T is S* in the Google n-grams, since *S* can now project onto T any element of *srcTypical*(*S*). Using macro-averaging over all 6,904 cases where $|src(S)| > 0$, we find that $srcTypical^2(S)$ covers **99.2**% of *typical*(S) on average. A well-chosen metaphor enables us to emphasize almost any quality of a topic T we might wish to highlight.

Affective Text Retrieval with Creative Metaphors

Suppose we have a database of texts $\{D_1 \ldots D_n\}$ in which each document D_i offers a creative perspective on a topic T. We might have texts that view politicians as crooks, popes as kings, or hackers as heroes. So given a query **+T**, can we retrieve only those texts that view T positively, and given **-T** can we retrieve only the negative texts about T?

We first construct a database of artificial figurative texts. For each stereotype *S* in the lexicon, and for each $M \in src(S) \cap (+R \cup -R)$, we construct a text D_{SM} in which *S* is viewed as *M*. The title of document D_{SM} is "S is M", while the body of D_{SM} contains all the words in *src*(M). D_{SM} uses the typical language of *M* to talk about S. For each D_{SM}, we know whether D_{SM} conveys a positive or negative viewpoint on *S*, since *M* sits in either in **+R** or **-R**.

The affect lexicon contains 5,704 stereotypes *S* for which $src(S) \cap (+R \cup -R)$ is non-empty. On average, each of these stereotypes is described in terms of 14 other stereotypes (5.8 are negative and 8.2 are positive, according to **+R** and **-R**) and we construct a representative document for each of these viewpoints. We construct a set of 79,856 artificial documents in total, to convey figurative perspectives on 5,704 different stereotypical topics:

Table 1. Macro-Average P/R/F1 scores for affective retrieval of + and - viewpoints for 5,704 topics.

Macro Average **(5704 topics)**	***Positive*** **viewpoints**	***Negative*** **viewpoints**
Precision	.86	.93
Recall	.95	.78
F-Score	.90	.85

For each document retrieved for T, we estimate its polarity as the mean of the polarity of the words it contains. Table 1 presents the results of this experiment, in which we attempt to retrieve only the positive viewpoints for T with a query **+T**, and only the negative viewpoints for T using **-T**. The results are sufficiently encouraging to support the further development of a creative text retrieval engine that is capable of ranking documents by the affective figurative perspective that they offer on a topic.

Concluding Thoughts: The Creative Web

Metaphor is a creative *knowledge multiplier* that allows us to expand our knowledge of a topic T by using knowledge of other ideas as a magnifying lens. We have presented here a robust, stereotype-driven approach that embodies this practical philosophy. Knowledge multiplication is achieved using an expansionary approach, in which an affective query is expanded to include all of the metaphors that are commonly used to convey this affective viewpoint. These viewpoints are expanded in turn to include all the qualities that are typically implied by each. Such an approach is ideally suited to a creative re-imagining of IR.

An implementation of these ideas is available for use on the web. Named *Metaphor Magnet*, the system allows users to enter queries of the form shown here (such as *Google is –Microsoft, Steve Jobs as Tony Stark, Rasputin as Karl Rove*, etc.). Each query is expanded into a set of apt metaphors mined from the Google n-grams, and each metaphor is expanded into a set of contextually apt qualities. In turn, each quality is expanded into an IR query that is used to retrieve relevant hits from Google. In effect, the system – still an early prototype – allows users to interface with a search engine like Google using metaphor and other affective language forms. The system can currently be accessed at this URL:

http://boundinanutshell.com/metaphor-magnet

Metaphor Magnet is just one possible application of the ideas presented here, which constitute not so much a philosophical or linguistic theory of metaphor, but an engineering-oriented toolkit of reusable concepts for imbuing a wide range of text applications with a robust competence in linguistic creativity. Human speakers do not view metaphor as a problem but as a solution. It is time our computational systems took a similarly constructive view of this remarkably creative cognitive tool.

In this vein, *Metaphor Magnet* continues to evolve as a creative web service. In addition to providing metaphors on demand, the service now also provides a poetic framing facility, whereby the space of possible interpretations for a given metaphor is crystallized into a single poetic form. More generally, poetry can be viewed as a means of reducing information overload, by summarizing a complex metaphor – or the set of texts retrieved using that metaphor via creative IR – whose interpretation entails a rich space of affective possibilities. A poem can thus be seen in functional terms as both an information summarization tool and as a visualization device. *Metaphor Magnet* adopts a simple, meaning-driven approach to poetry generation: given a topic T, a set of candidate metaphors with the desired affective slant is generated. One metaphor is chosen at random, and the elements of its interpretation are sampled to produce different lines of the resulting poem. Each element, and the sentiment it best evokes, is rendered in natural language using one of a variety of poetic tropes.

For example, *Metaphor Magnet* produces the following as a distillation of the space of feelings and associations that arise from the interpretation of *Marriage is a Prison*:

The legalized regime of this marriage

My marriage is a tight prison
The most unitary federation scarcely organizes so much
Intimidate me with the official regulation of your prison
Let your close confines excite me
O Marriage, you disgust me with your undesirable security

Each time we dip into the space of possible interpretations, a new poem is produced. One can use *Metaphor Magnet* to sample the space at will, hopping from one interpretation to the next, or from one poem to another. Here is an alternate rendition of the same metaphor in poetic form:

The official slavery of this marriage

My marriage is a legitimate prison
No collective is more unitary, or organizes so much
Intimidate me with the official regulation of your prison
Let your sexual degradation charm me
O Marriage, you depress me with your dreary consecration

In the context of our earlier worked example, which generated a space of metaphors to negatively describe Microsoft's perceived misuse of power, consider the following, which distills the assertion *Microsoft is a Monopoly* into an aggressive ode:

No Monopoly Is More Ruthless

Intimidate me with your imposing hegemony
No crime family is more badly organized,
or controls more ruthlessly
Haunt me with your centralized organization
Let your privileged security support me
O Microsoft, you oppress me with your corrupt reign

Poetry generation in *Metaphor Magnet* is a recent addition to the service, and its workings are beyond the scope of the current paper (though they may be observed in practice by visiting the aforementioned URL). For details of a related approach to poetry generation – one that also uses the stereotype-bearing similes described in Veale (2012) – the reader is invited to read Colton, Goodwin & Veale (2012).

Metaphor Magnet forms a key element in our vision of a *Creative Web*, in which web services conveniently provide creativity on tap to any third-party software application that requests it. These services include *ideation* (e.g. via metaphor generation & knowledge discovery), *composition* (e.g. via analogy, bisocation & conceptual blending) and *framing* (via poetry generation, joke & story generation, etc.). Since CC does not distinguish itself through distinct algorithms or representations, but through its unique goals

and philosophy, such a pooling of services will not only help the field achieve a much-needed critical mass, it will facilitate a greater penetration of CC ideas and approaches into the commercial software industry.

Acknowledgements

This research was supported by the WCU (World Class University) program under the National Research Foundation of Korea (Ministry of Education, Science and Technology of Korea, Project no. R31-30007).

References

Almuhareb, A. and Poesio, M. 2004. Attribute-Based and Value-Based Clustering: An Evaluation. In *Proc. of EMNLP 2004*. Barcelona.

Almuhareb, A. and Poesio, M. 2005. Concept Learning and Categorization from the Web. In *Proc. of the 27th Annual meeting of the Cognitive Science Society*.

Barnden, J. A. 2006. Artificial Intelligence, figurative language and cognitive linguistics. In: G. Kristiansen, M. Achard, R. Dirven, and F. J. Ruiz de Mendoza Ibanez (Eds.), *Cognitive Linguistics: Current Application and Future Perspectives*, 431-459. Berlin: Mouton de Gruyter.

Brants, T. and Franz, A. 2006. *Web 1T 5-gram Ver. 1*. Linguistic Data Consortium.

Brennan, S. E. and Clark, H. H. 1996. Conceptual Pacts and Lexical Choice in Conversation. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 22(6):1482-93.

Colton, S., Goodwin, J. and Veale, T. 2012. Full-FACE Poetry Generation.In *Proc. of ICCC 2012, the 3rd International Conference on Computational Creativity.* Dublin, Ireland.

Fass, D. 1991. Met*: a method for discriminating metonymy and metaphor by computer. *Computational Linguistics* 17(1):49-90.

Fass, D. 1997. Processing Metonymy and Metaphor. *Contemporary Studies in Cognitive Science & Technology*. New York: Ablex.

Fishelov, D. 1992. Poetic and Non-Poetic Simile: Structure, Semantics, Rhetoric. *Poetics Today*, 14(1), 1-23.

Hanks, P. 2005. Similes and Sets: The English Preposition 'like'. In: Blatná, R. and Petkevic, V. (Eds.), *Languages and Linguistics: Festschrift for Fr. Cermak*. Charles Univ., Prague.

Hanks, P. 2006. Metaphoricity is gradable. In: Anatol Stefanowitsch and Stefan Th. Gries (Eds.), *Corpus-Based Approaches to Metaphor and Metonymy*,. 17-35. Berlin: Mouton de Gruyter.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. *In Proc. of the 14th Int. Conf. on Computational Linguistics*, pp 539–545.

Martin, J. H. 1990. A Computational Model of Metaphor Interpretation. New York: Academic Press.

Mason, Z. J. 2004. CorMet: A Computational, Corpus-Based Conventional Metaphor Extraction System. *Computational Linguistics*, 30(1):23-44.

Mihalcea, R. 2002. The Semantic Wildcard. In *Proc. of the LREC Workshop on Creating and Using Semantics for Information Retrieval and Filtering*. Spain, May 2002.

Navigli, R. and Velardi, P. 2003. An Analysis of Ontology-based Query Expansion Strategies. Proc. of the workshop on Adaptive Text Extraction and Mining (ATEM 2003), at ECML, the 14th European Conf. on Machine Learning, 42–49

Salton, G. 1968. *Automatic Information Organization and Retrieval*. New York: McGraw-Hill.

Shutova, E. 2010. Metaphor Identification Using Verb and Noun Clustering. In *the Proc. of the 23rd International Conference on Computational Linguistics*, 1001-1010.

Taylor, A. 1954. Proverbial Comparisons and Similes from California. *Folklore Studies* 3. Berkeley: University of California Press.

Turney, P.D. and Littman, M.L. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning* 60(1-3):251-278.

Van Rijsbergen, C. J. 1979. *Information Retrieval*. Oxford: Butterworth-Heinemann.

Veale, T. 2004. The Challenge of Creative Information Retrieval. *Computational Linguistics and Intelligent Text Processing: Lecture Notes in Computer Science*, Vol. 2945/2004, 457-467.

Veale, T. and Hao, Y. 2007a. Making Lexical Ontologies Functional and Context-Sensitive. *In Proc. of the 46th Annual Meeting of the Assoc. of Computational Linguistics.*

Veale, T. and Hao, Y. 2007b. Comprehending and Generating Apt Metaphors: A Web-driven, Case-based Approach to Figurative Language. *In Proc. of AAAI 2007, the 22nd AAAI Conference on Artificial Intelligence.* Vancouver, Canada.

Veale, T. and Hao, Y. 2008. Talking Points in Metaphor: A concise, usage-based representation for figurative processing. *In Proceedings of ECAI'2008, the 18th European Conference on Artificial Intelligence.* Patras, Greece, July 2008.

Veale. T. 2011. Creative Language Retrieval: A Robust Hybrid of Information Retrieval and Linguistic Creativity. In Proc. of ACL'2011, the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.

Veale, T. 2012. Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity. London: Bloomsbury Academic.

Vernimb, C. 1977. Automatic Query Adjustment in Document Retrieval. *Information Processing & Mgmt,* 13(6):339-53.

Voorhees, E. M. 1994. Query Expansion Using Lexical-Semantic Relations. In *the proc. of SIGIR 94, the 17th International Conference on Research and Development in Information Retrieval*. Berlin: Springer-Verlag, 61-69.

Voorhees, E. M. 1998. Using WordNet for text retrieval. *WordNet, An electronic lexical database,* 285–303. MIT Press.

Way, E. C. 1991. Knowledge Representation and Metaphor. *Studies in Cognitive systems*. Holland: Kluwer.

Wilks, Y. 1978. Making Preferences More Active, *Artificial Intelligence* 11.

Xu, J. and Croft, B. W. 1996. Query expansion using local and global document analysis. In *Proc. of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*.

Evolving Figurative Images Using Expression-Based Evolutionary Art

João Correia and **Penousal Machado**
CISUC, Department of Informatics Engineering
University of Coimbra
3030 Coimbra, Portugal
jncor@dei.uc.pt, machado@dei.uc.pt

Juan Romero and **Adrian Carballal**
Faculty of Computer Science
University of A Coruña
Coruña, Spain
jj@udc.es, adrian.carballal@udc.es

Abstract

The combination of a classifier system with an evolutionary image generation engine is explored. The framework is composed of an object detector and a general purpose, expression-based, genetic programming engine. Several object detectors are instantiated to detect faces, lips, breasts and leaves. The experimental results show the ability of the system to evolve images that are classified as the corresponding objects. A subjective analysis also reveals the unexpected nature and artistic potential of the evolved images.

Introduction

Expression based Evolutionary Art (EA) systems have, in theory, the potential to generate any image (Machado and Cardoso 2002; McCormack 2007). In practice, the evolved images depend on the representation scheme used. As a consequence, the results of expression-based EA systems tend to be abstract images. Although this does not represent a problem, there is a desire to evolve figurative images by evolutionary means since the start of EA. An early example of such an attempt can be found in the work of Steven Rooke (World 1996).

McCormack (2005; 2007) identified the problem of finding a symbolic-expression that corresponds to a known "target" image as one of the open problems of EA. More exactly, the issue is not finding a symbolic-expression, since this can be done trivially as demonstrated by Machado and Cardoso (2002), the issue is finding a compact expression that provides a good approximation of the "target" image and that takes advantage of its structure. We address this open problem by generalizing the problem – i.e., instead of trying to match a target image we evolve individuals that match a given class of images (e.g. lips).

The issue of evolving figurative images has been tackled by two main types of approach: (i) Developing tailored EA systems which resort to representations that promote the discovery of figurative images, usually of a certain kind; (ii) Using general purpose EA systems and developing fitness assignment schemes that guide the system towards figurative images. In the scope of this paper we are interested in the second approach.

Romero et al. (2003) suggest combining a general purpose evolutionary art system with an image classifier trained to recognize faces, or other types of objects, to evolve images of human faces. Machado, Correia, and Romero (2012a) presented a system that allowed the evolution of images resembling human faces by combining a general-purpose, expression-based, EA system with an off-the-shelf face detector. The results showed that it was possible to guide evolution and evolve images evocative of human faces.

Here, we demonstrate that other classes of object can be evolved, generalizing previous results. The autonomous evolution of figurative images using a general purpose EC system has rarely been accomplished. As far as we know, evolving different types of figurative images using the same expression-based EC system and the same approach has never been accomplished so far (with the exception of user-guided systems).

We show that this can be attained with off-the-shelf classifiers classifiers, which indicates that the approach is generalizable, and also with purpose-built ones, which indicates that it is relatively straightforward to customize it to specific needs. We chose a rather ad-hoc set of classifiers in an attempt to demonstrate the generality of the approach.

The remainder of this paper is structured as follows: A brief overview of the related work is made in the next section; Afterwards we describe the approach for the evolution of objects describing the framework, the Genetic Programming (GP) engine, the object detection system, and fitness assignment; Next we explain the experimental setup, the results attained and their analysis and; Finally we draw overall conclusions and indicate future research.

Related Work

The use of Evolutionary Computation (EC) for the evolution of figurative images is not new. Baker (1993) focuses on the evolution of line drawings, using a GP approach. Johnston and Caldwell (1997) use a Genetic Algorithm (GA) to recombine portions of existing face images, in an attempt to build a criminal sketch artist. With similar goals, Frowd, Hancock, and Carson (2004) use a GA, Principal Components Analysis and eigenfaces to evolve human faces. The evolution of cartoon faces (Nishio et al. 1997) and cartoon face animations (Lewis 2007) through GAs has also been explored. Additionally, Lewis (2007) evolved human figures.

The previously mentioned approaches share two common aspects: the systems have been specifically designed for the

evolution a specific type of image; the user guides evolution by assigning fitness. The work of Baker (1993) is an exception, the system can evolve other types of line drawings, however it is initialized with hand-built line drawings of human faces.

These approaches contrast with the ones where general purpose evolutionary art tools, which have not been designed for a particular type of imagery, are used to evolve figurative images. Although the images created by their systems are predominantly abstract, Steven Rooke (World 1996) and Machado and Romero (see, e.g., 2011), among others, have successfully evolved figurative images using expression-based GP systems and user guided evolution. More recently, Secretan et al. (2011) created picbreeder, a user-guided collaborative evolutionary engine. Some of the images evolved by the users are figurative, resembling objects such as cars, butterflies and flowers.

The evolution of figurative images using hardwired fitness functions has also been attempted. The works of by Ventrella (2010) and DiPaola and Gabora (2009) are akin to a classical symbolic regression problem in the sense that a target image exists and the similarity between the evolved images and the target image is used to assign fitness. In addition to similarity, DiPaola and Gabora (2009) also consider expressiveness when assigning fitness. This approach results in images with artistic potential, which was the primary goal of these approaches, but that would hardly be classified as human faces. As far as we know, the difficulty to evolve a specific target image, using symbolic regression inspired approaches, is common to all "classical" expression-based GP systems.

The concept of using a classifier system to assign fitness is also a researched topic: in the seminal work of Baluja, Pomerlau, and Todd (1994) an Artificial Neural Network trained to replicate the aesthetic assessments is used; Saunders and Gero (2001) employ a Kohonen Self-Organizing network to determine novelty; Machado, Romero, and Manaris (2007) use a bootstrapping approach, relying on a neural network, to promote style changes among evolutionary runs; Norton, Darrell, and Ventura (2010) train Artificial Neural Networks to learn to associate low-level image features to synsets that function as image descriptors and use the networks to assign fitness.

Overview of the Approach

Figure 1 depicts an overview of the framework, which is composed of two main modules, an evolutionary engine and a classifier.

The approach can be summarized as follows:

1. Random initialization of the population;
2. Rendering of the individuals, i.e., genotype-phenotype mapping;
3. Apply the classifier to each phenotype;
4. Use the results of the classification to assign fitness; This may require assessing internal values and intermediate results of the classification;

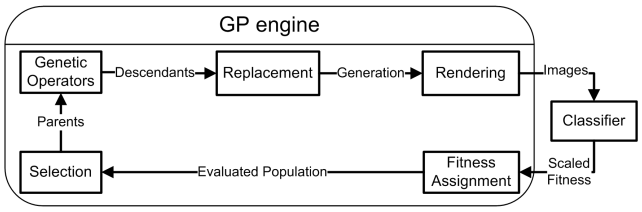


Figure 1: Overview of the system.

5. Select progenitors; Apply genetic operators, create descendants; Use the replacement operator to update the current population;
6. Repeat from 2 until some stopping criterion is met.

The framework was instantiated with a general-purpose GP-based image generation engine and with a Haar Cascade Classifier. To create a fitness function able to guide evolution it is necessary to convert the binary output of the detector to one that can provide suitable fitness landscape. This is attained by accessing internal results of the classification task that give an indication of the degree of certainty in the classification. In the following sections we explain the components of the framework, namely, the evolutionary engine, the classifier and the fitness function.

Genetic Programming Engine

The EC engine used in these experiments is inspired by the works of Sims (1991). It is a general purpose, expression-based, GP image generation engine that allows the evolution of populations of images. The genotypes are trees composed of a lexicon of functions and terminals. The function set is composed of simple functions such as arithmetic, trigonometric and logic operations. The terminal set is composed of two variables, x and y, and randomly initialized constants. The phenotypes are images that are rendered by evaluating the expression-trees for different values of x and y, which serve both as terminal values and image coordinates. In other words, to determine the value of the pixel in the (0,0) coordinates one assigns zero to x and y and evaluates the expression-tree (see figure 2). A thorough description of the GP engine can be found in (Machado and Cardoso 2002).

Figure 3 displays typical imagery produced via interactive evolution using this EC system.

Object Detection

For classification purposes we use Haar Cascade classifiers (Viola and Jones 2001). The classifier assumes the form of a cascade of small and simple classifiers that use a set of Haar features (Papageorgiou, Oren, and Poggio 1998) in combination with a variant of the Adaboost (Freund and Schapire 1995), and is able to attain efficient classifiers. This classification approach was chosen due to its state of the art relevance and for its fast classification. Both code and executables are integrated in the OpenCV API[1].

The face detection process can be summarized as follows:

[1]OpenCV — `http://opencv.org/`

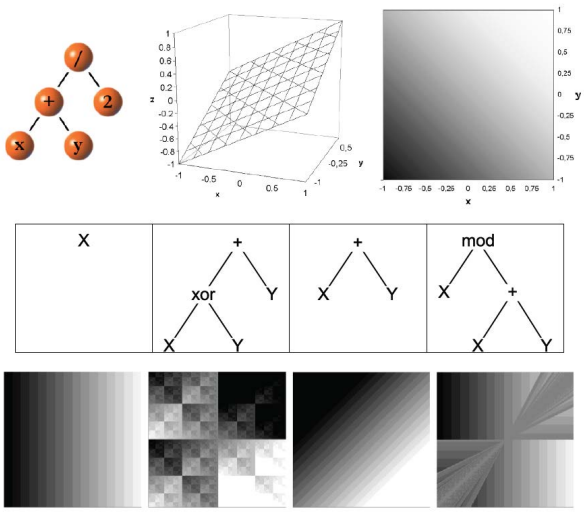


Figure 2: Representation scheme with examples of functions and the corresponding images.

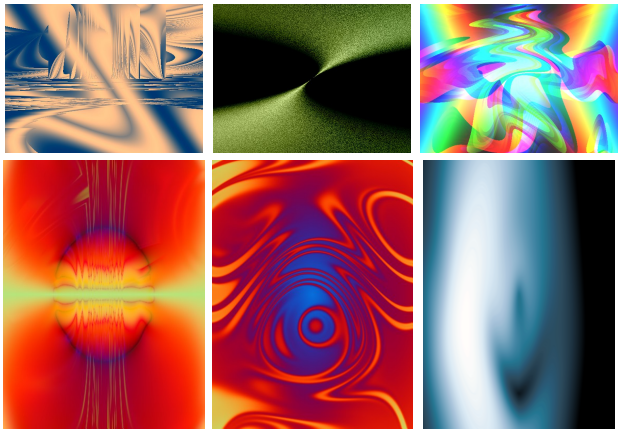


Figure 3: Examples of images generated by the evolutionary engine using interactive evolution.

1. Define a window of size w (e.g. 20×20).
2. Define a scale factor s greater than 1. For instance 1.2 means that the window will be enlarged by 20%.
3. Define W and H has the size of the input image.
4. From $(0, 0)$ to (W, H) define a sub-window with a starting size of w for calculation.
5. For each sub-window apply the cascade classifier. The cascade has a group of stage classifiers, as represented in figure 4. Each stage is composed, at its lower level, of a group of Haar features 5. Apply each feature of each corresponding stage to the sub-window. If the resulting value is lower than the stage threshold the sub-window is classified as a non-object and the search terminates for the sub-window. If it is higher continue to next stage. If all cascade stages are passed, the sub-window is classified as containing an object.
6. Apply the scale factor s to the window size w and repeat 5 until window size exceeds the image in at least one dimension.

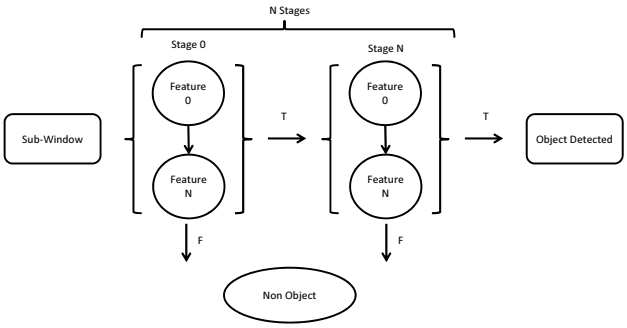


Figure 4: Cascade of classifiers with N stages, adapted from (Viola and Jones 2001).

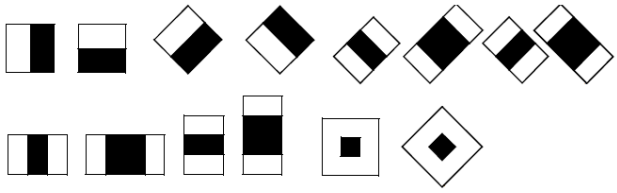


Figure 5: The set of possible features, adapted from (Lienhart and Maydt 2002).

Fitness Assignment

The process of fitness assignment is crucial from an evolutionary point of view, and therefore it holds a large importance for the success of the described system. The goal is to evolve images that the object detector classifies as an object of the positive class. However, the binary output of the detector is inappropriate to guide evolution. A binary function gives no information of how close an individual is to being a valid solution to the problem and, as such, the EA would be performing, essentially, a random search. It is necessary to extract additional information from the classification detection process in order to build a suitable fitness function.

This is attained by accessing internal results of the classification task that give an indication of the degree of certainty in the classification. Based on results of past experiments (Machado, Correia, and Romero 2012a; 2012b) we employ the following fitness function:

$$fitness(x) = \sum_{i}^{nstages_x} stagedif_x(i) * i + nstages_x * 10 \quad (1)$$

The underlying rational is the following: images that go through several classification stages, and closer to be classified as an object, have higher fitness than those rejected in early stages. Variables $nstages_x$ and $stagedif_x(i)$

Table 1: Haar Training parameters.

Parameter	Setting
Number of stages	30
Min True Positive rate per stage	99.9%
Max False Positive rate per stage	50%
Object Width	20 or 40(breasts,leaf)
Object Height	20 or 40(leaf)
Haar Features	ALL
Number of splits	1
Adaboost Algorithm	GentleAdaboost

are extracted from the object detection algorithm. Variable $nstages_x$, holds the number of stages that image, x, has successfully passed. That is, an image that passes several stages is likely to be closer of being recognized as having a object than one that passes fewer stages. In other words, passing several stages is a pre-condition to be classified as having the object. Variable $stagedif_x(i)$ holds the maximum difference between the threshold necessary to overcome stage i and the value attained by the image at the i^{th} stage. Images that are clearly above the thresholds are preferred over ones that are only slightly above them. Obviously, this fitness function is only one of the several possible ones.

Experimentation

Within the scope of this paper we intend to evolve the following objects: faces, lips, breasts and leaves. For the first two we use off-the-shelf classifiers that were already trained and used by other researchers in different lines of investigation (Lienhart and Maydt 2002; Lienhart, Kuranov, and Pisarevsky 2003; Santana et al. 2008). For the last two we created our own classifiers, by choosing suitable datasets and training the respective object classifier.

In order to construct an object classifier we need to construct two datasets: (i) positive – examples of images that contain the object we want to detect; (ii) negative – images that do not contain the object. Furthermore, for the positive examples, we must identify the location of the object in the images (see figure 6) in order to build the ground truth file that will be used for training.

For these experiments, the negative dataset was attained by picking images from a random search using image search engines, and from the Caltech-256 Object Category dataset (Griffin, Holub, and Perona 2007). Figure 7 depicts some of the images used as negative instances. In what concerns the positive datasets: the breast object detector was built by searching images on the web; the leaf dataset was obtained from the Caltech-256 Object Category dataset and from web searches. As previously mentioned, the face and lip detector are off-the-shelf classifiers. Besides choosing datasets we must also define the training parameters. Table 1 presents the parameters used for training of the cascade classifier.

The success of the approach is related to the performance of the classifier itself. By defining a high *number of stages* we are creating several stages that the images must overcome to be considered a positive example. The high *true positive rate* ensures that almost every positive example is

Figure 6: Examples of images used to train a cascade classifier for leaf detection. On the top row the original image, on the bottom row the croped example used for training.

learned per stage. The max *false positive rate* creates some margin for error, allowing the training to achieve the *minimum true positive rate* per stage and a low positive rate at the end of the cascade. Similar parameters were used and discussed in (Lienhart, Kuranov, and Pisarevsky 2003).

Once the classifiers are obtained, they are used to assign fitness in the course of the evolutionary runs in an attempt to find images that are recognized as faces, lips, breasts and leaves. We performed 30 independent evolutionary runs for each of these classes. In summary we have 4 classifiers, with 30 independent EC runs, totaling 120 EC runs.

The settings of the GP engine, presented in table 2, are similar to those used in previous experimentation in different problem domains. Since the classifiers used only deal with greyscale information, the GP engine was also limited to the generation of greyscale images. The population size used in this experiments 100 while in previous experiments we used a population size of 50 (Machado, Correia, and Romero 2012a). This allows us to sample a larger portion of the search space, contributing to the discovery of images that fit the positive class.

In all evolutionary runs the GP engine was able to evolve images classified as the respective objects. Similarly to the behavior reported by Machado, Correia, and Romero (2012a; 2012a), the GP engine was able to exploit weaknesses of the classifier, that is, the evolved images are classified as the object but, from a human perspective, they often fail to resemble the object. In figure 8 we present examples of such failures. As it can be observed, it is hard to recognize breasts, faces, leafs or lips in the presented images. It is important to notice that these weaknesses are not a byproduct of the fitness assignment scheme, as such they cannot be solved by using a different fitness function, nor particular to the classifiers used. Although different classi-

Figure 7: Examples of images belonging to the negative dataset used for training the cascade classifiers.

Table 2: Parameters of the GP engine. See (Machado and Cardoso 2002) for a detailed description.

Parameter	Setting
Population size	100
Number of generations	100
Crossover probability	0.8 (per individual)
Mutation probability	0.05 (per node)
Mutation operators	sub-tree swap, sub-tree replacement, node insertion, node deletion, node mutation
Initialization method	ramped half-and-half
Initial maximum depth	5
Mutation max tree depth	3
Function set	$+$, $-$, $\times$, $/$, min, max, abs, neg, warp, sign, sqrt, pow, mdist, sin, cos, if
Terminal set	x, y, random constants

fiers have different weaknesses, we confirmed that several of the evolved images that do not resemble faces are also recognized as faces by commercially available and widely used classifiers.

These results have opened a series of possibilities, including the use of this approach to assess the robustness of object detection systems, and also the use of evolved images as part of the training set of these classifiers in order to overcome some of their shortcomings. Although we already are pursuing that line of research and promising results have been obtained (Machado, Correia, and Romero 2012b), it is beyond the scope of the current paper.

When one builds a face detector, for instance, one is typically interested in building one that recognizes faces of all types, sizes, colors, sexes, in different lighting conditions, against clear and cluttered backgrounds, etc. Although the inclusion of all these examples may lead to a robust classifier that is able to detect all faces present in an image, it will also means that this classifier will be prone to recognize faces even when only relatively few features are present. In contrast, when building classifiers for the purpose described in this paper, one may select for positive examples clear and iconic images. Such classifiers would probably fail to identify a large portion of real-world images containing the object. However, they are would be extremely selective and, as such, the evolutionary runs would tend to converge to images that clearly match the desired object. Thus, although this was not explored, building a selective classifier can significantly reduce the number of runs that converge to atypical images such as the ones depicted in figure 8.

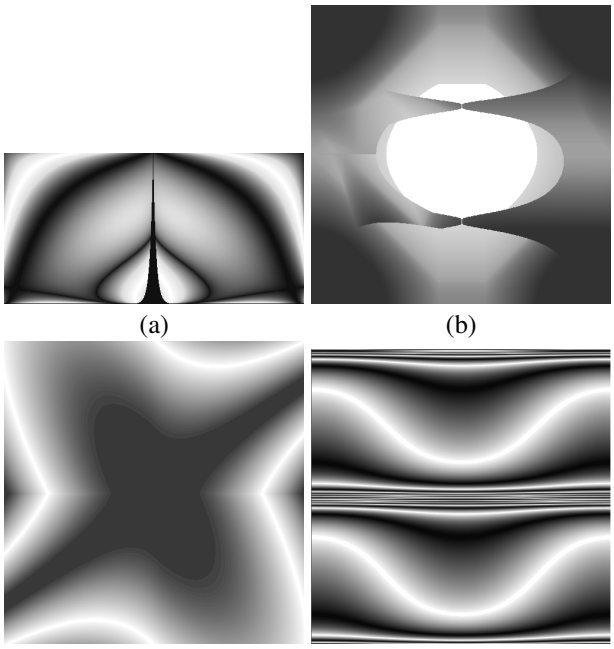


Figure 8: Examples of evolved images identified as objects by the classifiers that do not resemble the corresponding objects from a human perspective. This images were recognized as breasts (a), faces (b), leafs (c) and lips (d).

According to our subjective assessment, some runs were able to find images that actually resemble the object that we are trying to evolve. These add up to 6 runs from the face detector, 5 for the lip detector, 4 for the breast detector and 4 for the leaf detector.

In figures 9,10, 11 and 12 we show, according to our subjective assessment, some of the most interesting images evolved. These results allow us to state that, at least in some instances, the GP engine was able to create figurative images evocative of the objects that the object detector was design to recognize as belonging to the positive class.

By looking at the faces, figure 9, we can observe the presence of at least 3 facial features per image (such as eyes, lips, nose and head contour). The images from the first row have been identified by users as resembling wolverine. The

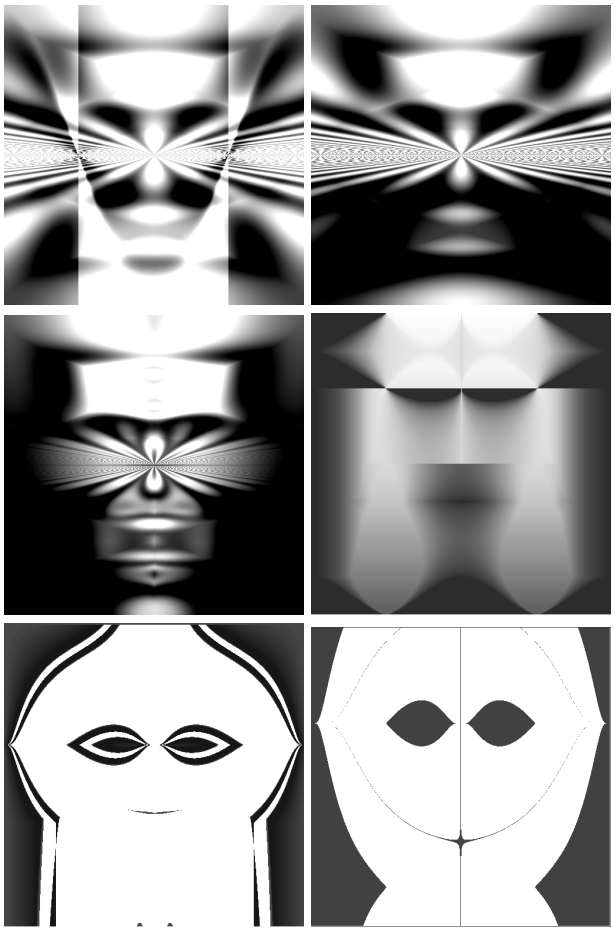


Figure 9: Examples of some of the most interesting images that have been evolved using face detection to assign fitness.

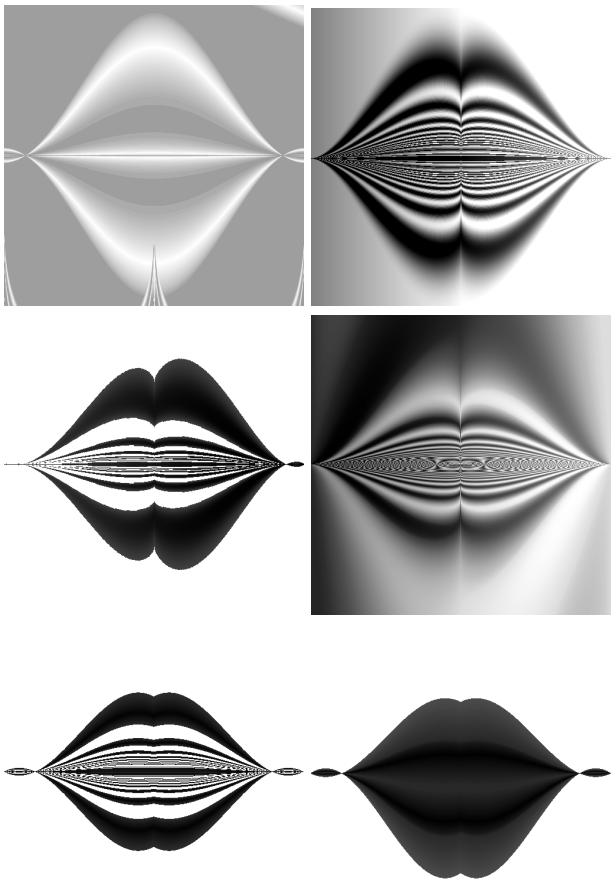


Figure 10: Examples of some of the most interesting images that have been evolved using a detector of lips to assign fitness.

ones of the second row, particularly the one on the left, have been identified as masks (more specifically african masks). In what concerns the images from the last row, we believe that their resemblance "ghost-like" cartoons is striking.

In what concerns the images resulting from the runs where a lip detector was used to assign fitness, we consider that their resemblance with lips, caricatures of lips, or lip logos, is self evident. The iconic nature of the images from the last row is particularly appealing to us.

The results obtained with the breast detector reveal images with well-defined or exaggerated features. We found little variety in these runs, with changes occurring mostly at the pixel intensity and contrast level. As previously mentioned, most of these runs resulted in unrecognizable images (see figure 8), which is surprising since the nature of the function set would lead us to believe that it should be relatively easy to evolve such images. Nevertheless, the successful runs present images that are clearly evocative of breasts.

Finally the images from the leaf detector, vary in type and shape. They share however a common feature they tend to be minimalist, resembling logos. In each of the images of the first row the detector identified two leaf shapes. On the

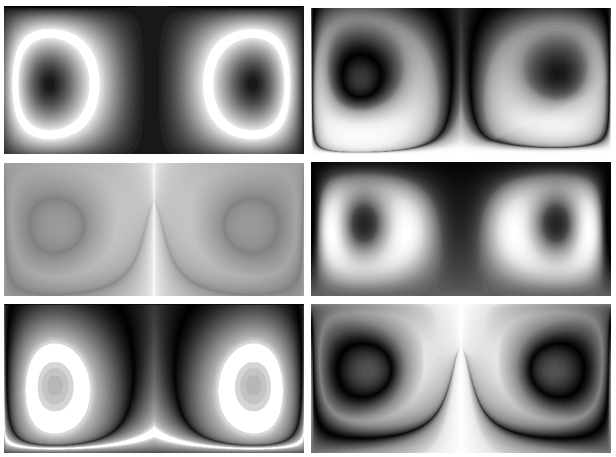


Figure 11: Examples of some of the most interesting images that have been evolved using a detector of breasts to assign fitness.

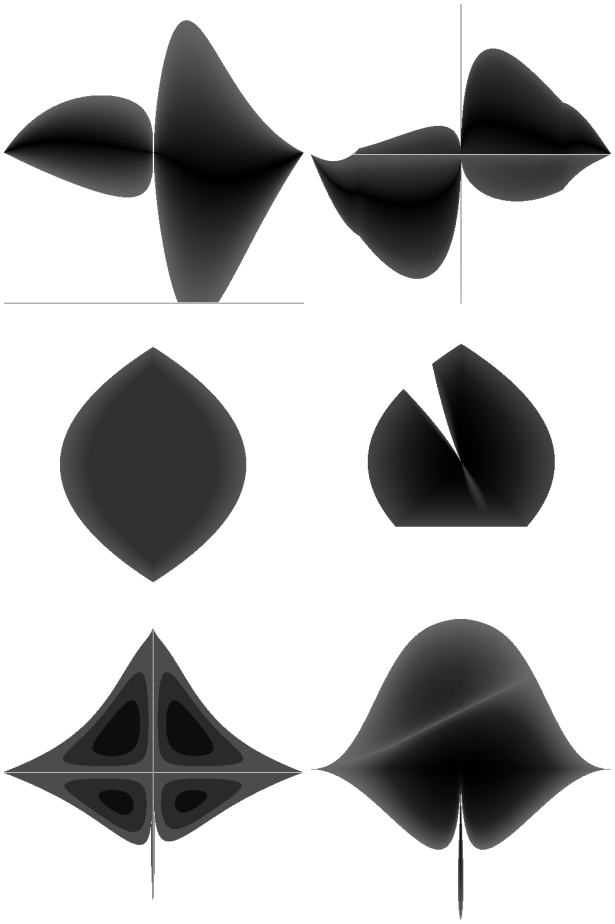


Figure 12: Examples of some of the most interesting images that have been evolved using a detector of leafs to assign fitness.

others a single leaf shape was detected.

In general, when the runs successfully evolve images that actually resemble the desired object, they tend to generate images that exaggerate the key features of the class. This is entirely consistent with the fitness assignment scheme that values images that are recognized with a high degree of certainty. This constitutes a valuable side effect of the approach, since the evolution of caricatures and logos fits our intention to further explore these images from a artistic and design perspective. The convergence to iconic, exaggerated instances of the class, may indicate the occurrence of the "Peak Shift Principle", but further testing is necessary to confirm this interpretation of the results.

Conclusions

The goal of this paper was to evolve different figurative images by evolutionary means, using a general-purpose expression based GP image generation engine and object detectors. Using the framework presented by Machado, Correia, and Romero (2012a), several object detectors were used to evolve images that resemble: faces, lips, breasts and leafs. The results from 30 independent runs per each classifier shown that is possible to evolve images that are detected as the corresponding objects and that also resemble that object from a human perspective. The images tend to depict an exaggeration of the key features of the associated object, allowing the exploration of these images in design and artistic contexts.

The paper makes 3 main contributions, addressing: (i) A well-known open problem in evolutionary art; (ii) The evolution of figurative images using a general-purpose expression based EC system; (iii) The generalization of previous results.

The open problem of finding a compact symbolic expression that matches a target image is addressed by generalization: instead of trying to match a target image we evolve individuals that match a given class. Previous results (see (Machado, Correia, and Romero 2012a)) concerned only the evolution of faces. Here we demonstrate that other classes of objects can be evolved. As far as we know, this is the first autonomous system that proved able to evolve different types of figurative images. Furthermore the experimental results show that this is attainable with off-the-shelf and purpose build classifiers, demonstrating that the approach is both generalizable and customizable.

Currently, we are performing additional tests with different object detectors in order to expand the types of imagery produced.

The next steps will comprise the following: combine, refine and explore the evolved images, using them in user-guided evolution and automatic fitness assignment schemes; combine multiple object detectors to help refine the evolved images (for instance use a face detector first and an eye or a lip detector next); use the evolved examples that are seen as shortcomings of the classifier to refine the training set and boost the existing detectors.

Acknowledgements

This research is partially funded by: the Portuguese Foundation for Science and Technology in the scope of project SBIRC (PTDC/EIA–EIA/115667/2009) and of the iCIS project (CENTRO-07-ST24-FEDER-002003), which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER; Xunta de Galicia Project XUGA?PGIDIT10TIC105008PR.

References

Baker, E. 1993. Evolving line drawings. Technical Report TR-21-93, Harvard University Center for Research in Computing Technology.

Baluja, S.; Pomerlau, D.; and Todd, J. 1994. Towards automated artificial evolution for computer-generated images. *Connection Science* 6(2):325–354.

DiPaola, S. R., and Gabora, L. 2009. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines* 10(2):97–110.

Freund, Y., and Schapire, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to

boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, 23–37. London, UK, UK: Springer-Verlag.

Frowd, C. D.; Hancock, P. J. B.; and Carson, D. 2004. EvoFIT: A holistic, evolutionary facial imaging technique for creating composites. *ACM Transactions on Applied Perception* 1(1):19–39.

Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology.

Johnston, V. S., and Caldwell, C. 1997. Tracking a criminal suspect through face space with a genetic algorithm. In Bäck, T.; Fogel, D. B.; and Michalewicz, Z., eds., *Handbook of Evolutionary Computation*. Bristol, New York: Institute of Physics Publishing and Oxford University Press. G8.3:1–8.

Lewis, M. 2007. Evolutionary visual art and design. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer Berlin Heidelberg. 3–37.

Lienhart, R., and Maydt, J. 2002. An extended set of haar-like features for rapid object detection. In *International Conference on Image Processing*, volume 1, I–900 – I–903 vol.1.

Lienhart, R.; Kuranov, E.; and Pisarevsky, V. 2003. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM 25th Pattern Recognition Symposium*, 297–304.

Machado, P., and Cardoso, A. 2002. All the truth about NEvAr. *Applied Intelligence, Special Issue on Creative Systems* 16(2):101–119.

Machado, P., and Romero, J. 2011. On evolutionary computer-generated art. *The Evolutionary Review: Art, Science, Culture* 2(1):156–170.

Machado, P.; Correia, J.; and Romero, J. 2012a. Expression-based evolution of faces. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012, Málaga, Spain, April 11-13, 2012. Proceedings*, volume 7247 of *Lecture Notes in Computer Science*, 187–198. Springer.

Machado, P.; Correia, J.; and Romero, J. 2012b. Improving face detection. In Moraglio, A.; Silva, S.; Krawiec, K.; Machado, P.; and Cotta, C., eds., *Genetic Programming - 15th European Conference, EuroGP 2012, Málaga, Spain, April 11-13, 2012. Proceedings*, volume 7244 of *Lecture Notes in Computer Science*, 73–84. Springer.

Machado, P.; Romero, J.; and Manaris, B. 2007. Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer Berlin Heidelberg. 381–415.

McCormack, J. 2005. Open problems in evolutionary music and art. In Rothlauf, F.; Branke, J.; Cagnoni, S.; Corne, D. W.; Drechsler, R.; Jin, Y.; Machado, P.; Marchiori, E.; Romero, J.; Smith, G. D.; and Squillero, G., eds., *EvoWorkshops*, volume 3449 of *Lecture Notes in Computer Science*, 428–436. Springer.

McCormack, J. 2007. Facing the future: Evolutionary possibilities for human-machine creativity. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer Berlin Heidelberg. 417–451.

Nishio, K.; Murakami, M.; Mizutani, E.; and N., H. 1997. Fuzzy fitness assignment in an interactive genetic algorithm for a cartoon face search. In Sanchez, E.; Shibata, T.; and Zadeh, L. A., eds., *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives*, volume 7. World Scientific.

Norton, D.; Darrell, H.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *Proceedings of the First International Conference Computational Creativity*, 26–35.

Papageorgiou, C. P.; Oren, M.; and Poggio, T. 1998. A general framework for object detection. In *Sixth International Conference on Computer Vision*, 555–562.

Romero, J.; Machado, P.; Santos, A.; and Cardoso, A. 2003. On the development of critics in evolutionary computation artists. In Günther, R., et al., eds., *Applications of Evolutionary Computing, EvoWorkshops 2003: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 2611 of *LNCS*. Essex, UK: Springer.

Santana, M. C.; Déniz-Suárez, O.; Antón-Canalís, L.; and Lorenzo-Navarro, J. 2008. Face and facial feature detection evaluation - performance evaluation of public domain haar detectors for face and facial feature detection. In Ranchordas, A., and Araújo, H., eds., *VISAPP (2)*, 167–172. INSTICC - Institute for Systems and Technologies of Information, Control and Communication.

Saunders, R., and Gero, J. 2001. The digital clockwork muse: A computational model of aesthetic evolution. In Wiggins, G., ed., *AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science*, 12–21.

Secretan, J.; Beato, N.; D'Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; Folsom-Kovarik, J. T.; and Stanley, K. O. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation* 19(3):373–403.

Sims, K. 1991. Artificial evolution for computer graphics. *ACM Computer Graphics* 25:319–328.

Ventrella, J. 2010. Self portraits with mandelbrot genetics. In *Proceedings of the 10th international conference on Smart graphics*, SG'10, 273–276. Berlin, Heidelberg: Springer-Verlag.

Viola, P., and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 1:511.

World, L. 1996. Aesthetic selection: The evolutionary art of steven Rooke. *IEEE Computer Graphics and Applications* 16(1).

Fitness Functions for Ant Colony Paintings

Penousal Machado and Hugo Amaro
CISUC, Department of Informatics Engineering
University of Coimbra
3030 Coimbra, Portugal
machado@dei.uc.pt, hamaro@student.dei.uc.pt

Abstract

A creativity-support tool for the creation of non-photorealistic renderings of images is described. It employs an evolutionary algorithm that evolves the parameters governing the behavior of ant species, and the paintings are produced by simulating the behavior of these artificial ants. The design of fitness functions, using both behavioral and image features is discussed, emphasizing the rationale and intentions that guided the design. The analysis of the experimental results obtained by using different fitness functions focuses on assessing if they convey the intentions of the fitness function designer.

Introduction

Machado and Pereira (2012) presented a non-photorealistic rendering (NPR) algorithm inspired on ant colony approaches: the trails of artificial ants were used to produce a rendering of an original input image. One of the novel characteristics of this algorithm is the adoption of scalable vector graphics, which contrasts with the pixel based approaches used in most ant painting algorithms, and enables the creation of resolution independent images. The trail of each ant is represented by a continuous line of varying width, contributing to the expressiveness of the NPRs.

In spite of the potential of this generative approach, the number of parameters controlling the behavior of the ants and their interdependencies was soon revealed to be too large to allow their tuning by hand. The results of these attempts revealed that only a small subset of the creative possibilities allowed by the algorithm was being explored.

To tackle this problem, Machado and Pereira (2012) presented a human-in-the-loop Genetic Algorithm (GA) to evolve the parameters, allowing the users to guide the algorithm according to their preferences and avoiding the need to understand the intricacies of the algorithm. Thus, instead of being forced to perform low-level changes, the users of this creativity-support tool become breeders of species of ants that produce results that they find valuable. The experimental results highlight the range of imagery that can be evolved by the system showing its potential for the production of large-format artworks.

This paper describes a further step in the automation of the space exploration process and departure from low-level modification and assessment. The users become designers of fitness functions, which are used to guide evolution, leading to results that are consistent with the user intentions. To this end, while the ants paint, statistics describing their behavior are gathered. Once each painting is completed image features are calculated. These behavioral and image features are the basis for the creation of the fitness functions.

Human-in-the-loop in evolutionary art systems are often used as creativity-support tools and thought to have the potential for exploratory creativity. Allowing the users to design fitness functions by specifying desired combinations of characteristics provides an additional level of abstraction, enabling the users to focus on their intents and overcoming the user fatigue problem. Additionally, this approach opens the door for evaluating the system by comparing the intents of the user with the outcomes of the process.

We begin with a short survey of related work. Next, in the third section, we describe the system, focusing on the behavior of the ants and on the evolutionary algorithm. In the fourth section we present experimental results, making a brief analysis. Finally, we draw some conclusions and discuss aspects to be addressed in future work.

State of the Art

In this section we make a survey of related works, focusing on systems that use artificial ants for image generation purposes and on systems where evolutionary computation is employed for NPR purposes.

Tzafestas (2000) presents a system where artificial ants pick-up and deposit food, which is represented by paint, and studies the self-regulation properties and complexity of the system and resulting images. Ramos and Almeida (2000) explore the use of ant systems for pattern recognition purposes. The artificial ants successfully detect the edges of the images producing stylized renderings of the originals and smooth transitions between different images. The artistic potential of these approaches is explored in later works (Ramos 2002) and thorough his collaboration with the artist Leonel Moura, resulting in several robotic swarm drawings (Moura 2002). Urbano (2005; 2007; 2011) presents several multi-agent systems based on artificial ants.

Aupetit et al. (2003) introduce an interactive GA for the creation of ant paintings. The algorithm evolves parameters of the rules that govern the behavior of the ants. The artificial ants deposit paint on the canvas as they move, thus pro-

Figure 1: Screenshot of the graphic user interface. Control panel on the left and current population of ant paintings on the right.

ducing a painting. In a later study, Monmarché et al. (2007) refine this approach exploring different rendering modes. Greenfield (2005) presents an evolutionary approach to the production of ant paintings and explores the use of behavioral statistics of the artificial ants to automatically assign fitness. Later Greenfield (2006) adopted a multiple pheromone model where ants' movements and behaviors are influenced (attracted or repelled) by both an environmentally generated pheromone and an ant generated pheromone.

The use of evolutionary algorithms to create image filters and NPRs of source images has been explored by several researchers. Focusing on the works where there was an artistic goal, we can mention the research of: Ross et al. (2006) and Neufeld et al. (2007), where Genetic Programming (GP), multi-objective optimization techniques, and an empirical model of aesthetics are used to automatically evolve image filters; Lewis (2004), evolves live-video processing filters through interactive evolution; Machado et al. (2002), use GP to evolve image coloring filters from a set of examples; Yip (2004) employs GAs to evolve filters that produce images that match certain features of a target image; Collomosse (2006; 2007) uses image salience metrics to determine the level of detail for portions of the image, and GAs to search for painterly renderings that match the desired salience maps; Hewgill and Ross (2003) use GP to evolve procedural textures for 3D objects; Machado and Graça (2008) employ GP to evolve assemblages of 3D objects that are an artistic representation of an input image.

The Framework

The system is composed of two main modules: the evolutionary engine and the painting algorithm. A graphic user interface gives access to these modules (see Fig. 1). Each genotype of the GA population encodes the parameters of a species of ants. These parameters determine how that ant species reacts to the input image. Each painting is produced by simulating the behavior of ants of a given species while they travel across the canvas, leaving a trail of varying width and transparency.

In the following sections we describe the framework. First, we present the painting algorithm. Next, we describe the evolutionary component. Finally, we detail the behavioral and image features that are gathered.

Figure 2: On the left, an ant with five sensory vectors. On the middle, the living canvas of an ant species. On the right, its painting canvas.

The Painting Algorithm

Our ants live on the 2D world provided by the input image and they paint on a painting canvas that is initially empty (i.e., black). Both living and painting canvas have the same dimensions and the ants move simultaneously on both canvas. The painting canvas is used exclusively for depositing ink and has no interference with the behavior of the ants. Each ant has a position, color, deposit transparency and energy; all the remaining parameters are shared by the entire species. If the energy of an ant is bellow a given threshold it dies, if is is above a given threshold it generates offspring.

The luminance of an area of the living canvas represents the available energy, i.e. food, at that point. Therefore, ants may gain energy by traveling through bright areas. The energy consumed by the ant is removed from the living canvas, as will be explained later in detail.

The ants' movement is determined by how they react to light. Each ant senses the environment by "looking" in several directions (see Fig. 2). We use 10 sensory vectors, each vector has a given direction relative to the current direction of the ant and a length. The sensory organs return the luminance value of the area where each vector ends. To update the position of an ant one performs a weighted sum, calculating the sum of the sensory vectors divided by their norms,

multiplied by the luminance of their end point and by the weight the ant gives to each sensor. The result of this operation is multiplied by a scaling scalar that represents the ant's base speed. Subsequently, to represent inaccuracy of movement and sensory organs, the direction is perturbed by the addition of Perlin (1985) noise to its angle.

The ant simulation algorithm is composed of the following steps:

1. Initialization: n ants are placed on the canvas on pre-established positions; Each ants assumes the color of the area where it was placed; Their energy and deposit transparencies are initialized using the species parameters;
2. For each ant:
   (a) Update the ant's energy;
   (b) Update the energy of the environment;
   (c) Place ink on the painting canvas;
   (d) If the ant's energy is bellow the death threshold remove the ant from the colony;
   (e) If the ant's energy is above the reproduction threshold generate an offspring; The offspring assumes the color of the position where it was created and a percentage of the energy of the progenitor (which loses this energy); The offspring inherits the velocity of the parent, but a perturbation is added to the angular velocity by randomly choosing an angle between $descvel_{min}$ and $descvel_{max}$ (both values are species' parameters); Likewise, the deposit transparency is inherited from the progenitor but a perturbation is included by adding a randomly choosen a value between $dtransp_{min}$ and $dtransp_{max}$;
   (f) Update ant's position;
3. Repeat from 2 until no living ants exist;

Steps (b) and (c) require further explanation. The consumption of energy is attained by drawing on the living canvas a black circle of size equal to $energy * cons_{rate}$ of a given transparency ($cons_{trans}$) . Ink is deposited on the paining canvas by drawing a circle of the color of the ant – which is attributed when the ant is born – with a size given by ($energy * deposit_{rate}$) and of given transparency ($deposit_{transp}$). Fig. 2 depicts the living and painting canvas of an ant species during the simulation process. It is important to notice that the color of an ant is determined at birth. Thus, the ants may carry this color to areas of the canvas that possess different colors in the original image. A detailed description of the painting algorithm can be found in Machado and Pereira (2012).

Evolutionary Engine

As previously mentioned, we employ a GA to evolve the ant species' parameters. The genotypes are tuples of floating point numbers which encode the parameters of the ant species. The size of the genotype depends on the experimental settings. Table 1 presents an overview of the encoded parameters. We use a two point crossover operator for recombination purposes and a Gaussian mutation operator. We employ tournament selection and an elitist strategy, the highest ranked individual proceeds – unchanged – to the next population.

Table 1: Parameters encoded by the genotype

Name	#	Comments
$gain$	1	scaling for energy gains
$decay$	1	scaling for energy decay
$cons_{rate}$	1	scaling for size of circles drawn on the living canvas
$cons_{trans}$	1	transparency of circles drawn on the living canvas
$deposit_{rate}$	1	scaling for size of circles drawn on the painting canvas
$deposit_{transp}$	1	base transparency of circles drawn on the painting canvas
$dtransp_{min}$	1	limits for perturbation of deposit transparency when offsprings are generated
$dtransp_{max}$	1	
$initial_{energy}$	1	initial energy of the starting ants
$death_{threshold}$	1	death energy treshold
$birth_{threshold}$	1	generate offspring energy threshold
$descvel_{min}$	1	limits for perturbation of angular velocity when offsprings are generated
$descvel_{max}$	1	
vel	1	base speed of the ants
$noise_{min}$	1	limits for the perlin noise generator function
$noise_{max}$	1	
$initial_{positions}$	$2 * n$	initial coordinates of the n ants placed on the canvas
$sensory_{vectors}$	$2 * m$	direction and length of the m sensory vectors
$sensory_{weights}$	m	weights of the m sensory vectors

The Features

During the simulation of each ant species the following behavioral statistics are collected:

$avg(ants)$ Average number of living ants;

$coverage$ Proportion of the living canvas visited by the ants; An area is considered to be visited if, at least, one ant consumed resources from that area;

$deposited_{ink}$ The total amount of "ink" deposited by the ants; This is calculated by multiplying the area of each circle drawn by the ants by the opacity (i.e. $1 - transparency$) used to draw it.

$avg(trail), std(trail)$ The average trail length and the standard deviation of the trail lengths, respectively;

$avg(life), std(life)$ The average life span of the ants and its standard deviation, respectively;

$avg(distance)$ The average euclidean distance between the position where the ant was born and the one where it died;

$avg(avg(width)), std(avg(width))$ For each trail we calculate its average width, then we calculate the average width of all trails, $avg(avg(width))$, and the standard deviation of the averages, $std(avg(width))$;

$avg(std(width)), std(std(width))$ For each trail we calculate the standard deviation of its width, then we calculate their average, $avg(std(width))$, and their standard deviation $std(std(width))$;

$avg(avg(av)), std(avg(av)), avg(std(av)), std(std(av))$ These statistics are analogous to the ones regarding trail width, but pertaining to the angular velocity of the ants;

When the simulation of each ant species ends we calculate the following image features:

$complexity$ the image produced by the ants, *I*, is encoded in $jpeg$ format, and its complexity estimated using the following formula:

$$complexity(I) = rmse(I, jpeg(I)) \times \frac{s(jpeg(I))}{s(I)},$$

where *rmse* stands for the root mean square error, $jpeg(I)$ is the image resulting from the $jpeg$ compression of I, and *s* is the file size function

$fract_{dim}, lac$ The fractal dimension of the ant painting estimated by the box-counting method and its λ lacunarity value estimated by the Sliding Box method (Karperien 2012), respectively;

$inv(rmse)$ The similarity between the ant painting and the original image estimated as follows:

$$inv(rmse) = \frac{1}{1 + rmse(I, O)},$$

where I is the ant painting and O is the original image;

Experimental results

The results presented in this section were obtained using the following experimental setup: Population Size = 25; Tournament size = 5; Crossover probability = 0.9; Mutation Probability = 0.1 (per gene); Initial Position of the ants = the image is divided in 3×3 rectangles of the same size and one ant is placed at the center of each of these rectangles; Initial number of ants = 9; Maximum number of ants = 250; Maximum number of simulation steps 1000. Thus, when the drawing stage starts each ant species is represented by nine ants. However, these ants may generate offspring during simulation, increasing the number of ants in the canvas.

Typically, interactive runs had 30 to 40 generations, although some were significantly longer. The runs conducted using explicit fitness functions lasted 50 generations. For each fitness function we conducted 10 independent runs.

User Guided Runs

Machado and Pereira (2012) describe and analyze results attained in the course of user guided runs. In Fig. 3 we depict some of the individuals evolved in those runs, with the goal of giving a flavor of the different types of imagery that were evolved.

Figure 3: Examples from user guided runs.

Using Features Individually

To test the evolutionary algorithm we performed runs where each feature, with the exception of $frac_{dim}$ and lac, was used as fitness function. Maximizing the values of fractal dimension and lacunarity would lead to results that we find uninteresting. Therefore, we established for these features by measuring the fractal dimension and lacunarity of one of our favorite ant paintings evolved in user guided runs, 1.5 and 0.95, respectively, and the maximum fitness is obtained when these values are reached. For these two features, fitness is assigned by the following formula:

$$fitness = \frac{1}{1 + |target_{value} - feature_{value}|}$$

In Fig. 4 we present the evolution of fitness across the evolutionary runs. To avoid clutter we only present a subset of the considered fitness functions. In general, the evolutionary algorithm was able to find, in all runs and for all features, individuals with high fitness in relatively few generations. Unsurprisingly, and although it is subjective to say it, the runs tended to converge to ant paintings that, at least in our eyes, are inferior to the ones created in the course of interactive runs. Fig. 5 depicts the individuals that obtained the maximum fitness value for the corresponding image features. These individuals are representative of the imagery evolved in the corresponding runs.

It worth to notice that high $complexity$ is obtained by evolving images with abrupt transitions from black to white. This results in high frequencies that make $jpeg$ compression inefficient, thus resulting in high complexity estimates. The results attained with $lacunarity$ yield paintings with "gaps" between lines, revealing the black background,

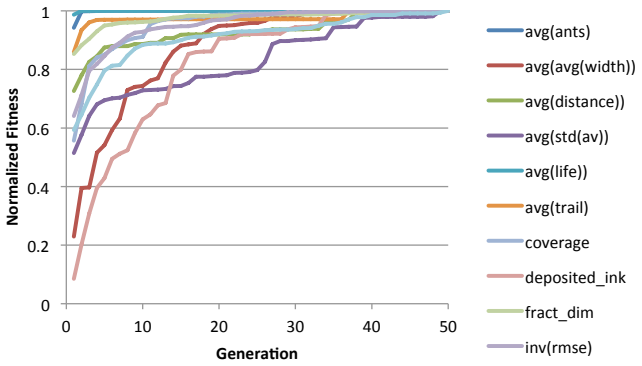


Figure 4: Evolution of the maximum fitness. The results are averages of 10 independent runs. The results have been normalized to allow the presentation of the results using distinct fitness functions in the same chart.

which matches the texture of the image from where the target $lacunarity$ value was collected. This contrasts with the results obtained using $fract_{dim}$, while the algorithm was able to match the target fractal dimension value, the images produced are radically different from the target's image. The $inv(rmse)$ runs revealed images that reproduce the original with some degree of fidelity, showing that this feature can promote similarity between the painting and the original.

The results obtained using a single behavioral feature are uninteresting in the context of NPR. They tend to fall in two categories, either they constitute "poor" variations of the original or they are unrecognizable versions of it.

Combining Behavioral and Image Features

From the beginning it was clear that it would be necessary to combine several features to attain our goals. To make the fitness function design process easy to understand, and thus allow inexperienced users to design their own fitness functions, we decided that all fitness functions should assume the form of a weighted sum.

Since different features have different ranges of values, it is necessary to normalize them, otherwise some features would outweigh the others. Additionally, angular velocity may be negative, so we should consider the absolute values. Considering these issues, normalization is attained by the following formula:

$$norm(feature) = abs\left(\frac{feature}{offlinemax(feature)}\right),$$

where $offlinemax$ returns the maximum value found in the course of the runs described in the previous section for the feature in question.

This modification is not sufficient to prevent the evolutionary algorithm to focus exclusively on a single feature. To minimize this problem, we consider a logarithmic scale so that the evolutionary advantage decreases as the feature value becomes higher, promoting the discovery of individuals that use all features employed in the fitness function. This is accomplished as follows:

$$lognorm(feature) = log(1 + norm(feature))$$

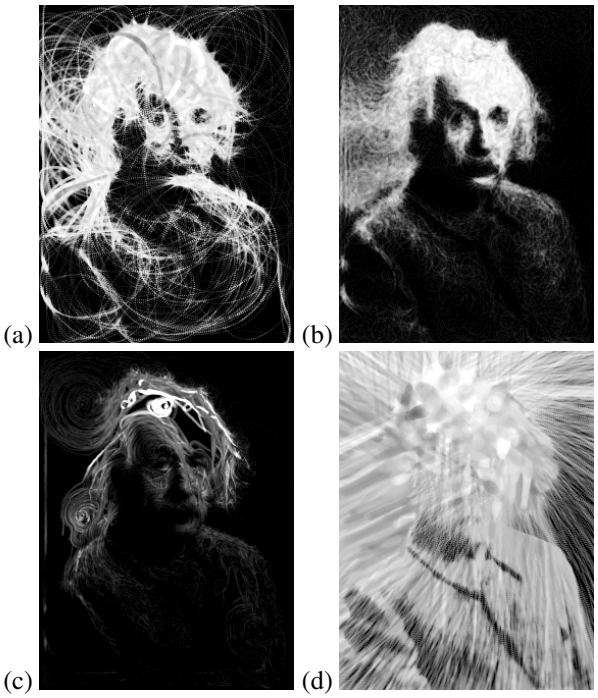


Figure 5: The individuals that obtained the maximum fitness value for: (a) $Complexity$; (b) $inv(rmse)$; (c) lac; (d) $fract_{dim}$.

All the fitness functions that combine several features are weighted sums of the $lognorm$ of each of the features used. However, for the sake of simplicity we will only mention the feature names when writing their formulas. From here onwards $feature$ should be read as $lognorm(feature)$.

Next we describe several fitness functions that combine a variable number of features. The analysis of the experimental results of evolutionary art systems is subjective by nature. As such, more than presenting measures of performance that would be meaningless when considering the goals of our system, we focus on describing the intentions behind the design of each fitness function, and make a subjective analysis of the results based on the comparison between the evolved paintings and our original design intentions.

f1: $coverage + complexity + lac$

The design of this fitness function was prompted by the results obtained in previous tests. The goal is to evolve ant paintings where the entire canvas is visited, with high $complexity$, and with a $lacunarity$ value of 0.95.

As it can be observed in Fig. 6 the evolved paintings successfully match these criteria. By comparing them with the ones presented in Fig. 5 one can observe how $lacunarity$ influences texture, $complexity$ leads high frequencies, and coverage promotes visiting most of the canvas.

f2: $inv(rmse) - 0.5 * complexity$

The rationale for this fitness function is obtaining a good approximation to the original image while keeping the complexity low. Thus, we wish to obtain a simplified version of

Figure 6: Two of the fittest images evolved using **f1**.

Figure 7: Two of the fittest images evolved using **f2**.

Figure 8: Two of the fittest images evolved using **f3**.

the original. Preliminary tests indicate the tendency of the algorithm to focus, exclusively, on minimizing $complexity$, which was achieved by producing images that were entirely black. Since this sort of image exists in the initial populations of the runs, this is a case of premature convergence. To circumvent it we decreased the weight given to $complexity$, which allowed the algorithm to escape this local optimum.

Although the results are consistent with the design (see Fig. 7) they do not depict the degree of abstraction and simplification we intended. As such, they should be considered a failure since they do not match our design intentions.

f3: $avg(std(width))+std(avg(width))-avg(avg(width))+inv(rmse)$

Here we focus on the width of the lines being drawn

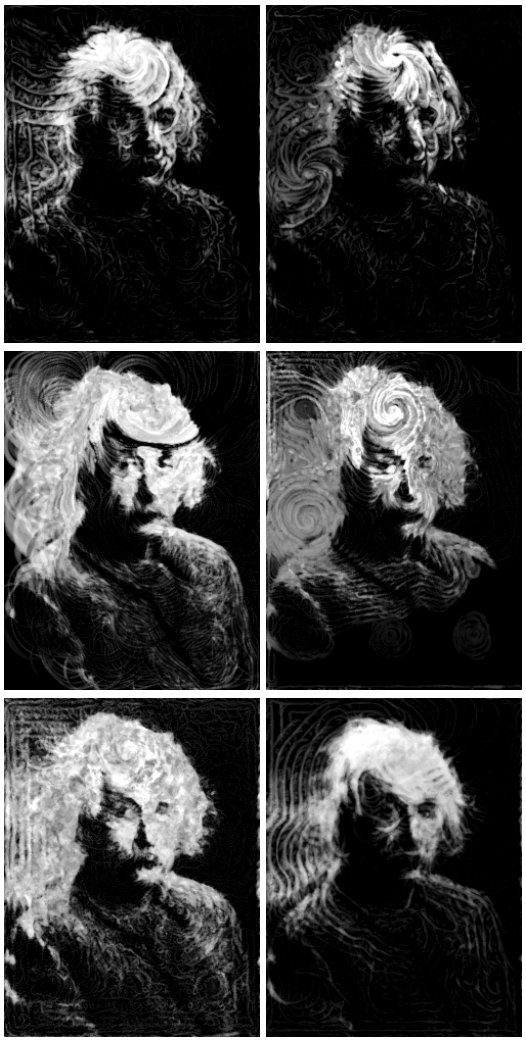


Figure 9: Two of the fittest images evolved using **f4** (first row), **f5** (second row) and **f6** (third row).

promoting the evolution of ant paintings with lines with high variations of width, $avg(std(width))$, heterogeneous widths among lines, $std(avg(width))$, and thin lines, $-avg(avg(width))$. To avoid radical deviations from the original drawing we also value $inv(rmse)$.

The experimental results, Fig. 8, depict these characteristics, however to fully observe the intricacies of the ant paintings a resolution higher than the space constraints of this paper allows would be required.

f4: $avg(std(av)) + inv(rmse) + coverage$
f5: $avg(avg(av)) - avg(std(av)) + inv(rmse) + coverage$
f6: $-avg(avg(av)) + avg(std(av)) + inv(rmse) + coverage$

When designing **f4-f6** we focused on controlling line direction. In **f4** we use $avg(std(av))$ to promote the appearance of lines that often change direction. In **f5** we use $avg(avg(av)) - avg(std(av))$ to encourage the appearance of circular motifs (high angular velocity and low variation of velocity). Finally, **f6** is a refinement of **f4** with

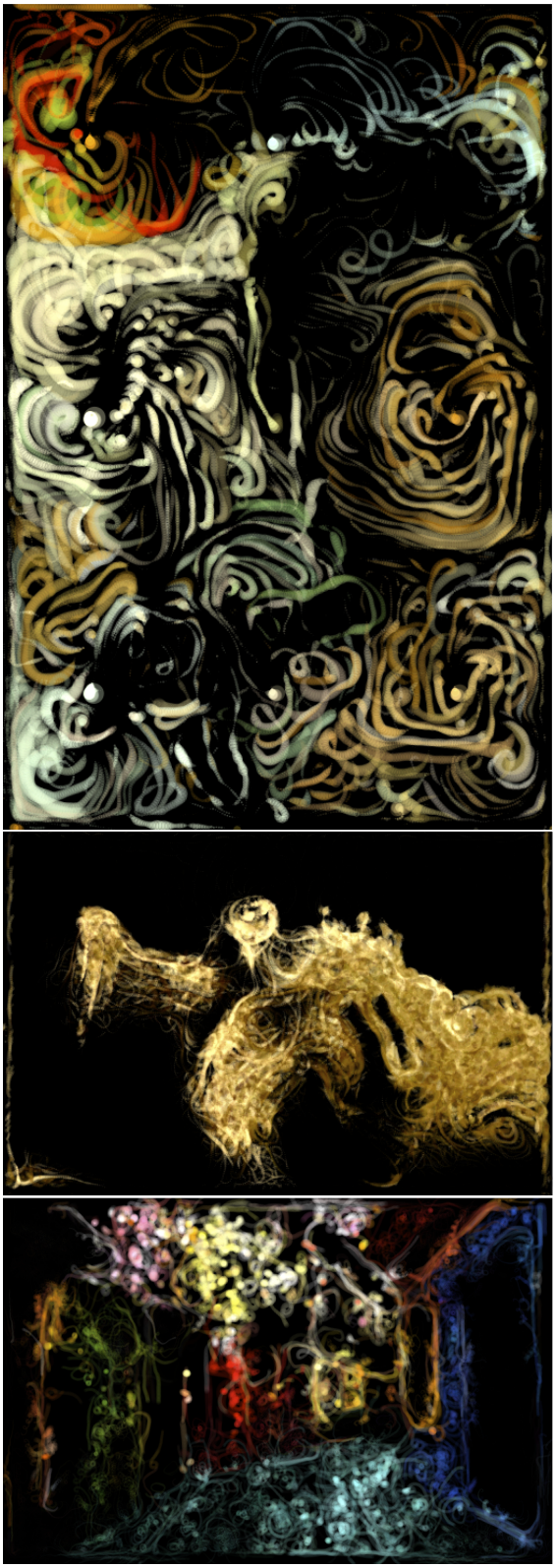


Figure 10: Results obtained by applying an individual from the **f4** runs to different input images.

$-avg(avg(av))$ preventing the appearance of circular patterns, valuing trails that curve in both directions, attaining an average angular velocity close to zero.

In all cases, the addition of $inv(rmse)$ and $coverage$ serves the goal of evolving ant paintings with some similarity to the original and that visit a large portion of the canvas.

In Fig 9 we present some of the outcomes of this experiences. As it can be observed the evolved images closely match our expectations and, as such, we consider them to be some of the most successful runs.

Once the individuals are evolved the ant species may be applied to different input images, hopefully resulting in ant-paintings that share the characteristics that we value. This is one of the key aspects of the system: although finding a valuable ant species may be time consuming, once it is found it can be applied with ease to other images producing large-scale NPR of them. In Fig. 10 we present ant paintings created by this method.

Conclusions

We presented a creativity-support tool that aids the users by providing a wide variety of paintings, which are arguably consistent with the intentions of the users, and which they would be unlikely to imagine on their own. While using this tool the users become designers of fitness functions, which are built using a combination of behavioral and image features. We reported the results obtained, focusing on the comparison between the evolved ant-paintings and the design intentions that led to the development of each fitness function.

Overall the results indicate that it is possible, to some extent, to convey design intention through fitness functions, leading to the discovery of individuals that match these intentions. This allows the users to operate at a higher level of abstraction than in user guided runs, circumventing the user-fatigue problem typically associated with interactive evolution. The analysis of the results also reveals the discovery of high-quality ant paintings that are radically different from the ones obtained through interactive evolution.

Although the system serves the user intents, different runs converge to different, and sometimes highly dissimilar, images. Each fitness function can be maximized in a multitude of ways, some of which are quite unexpected. As such, we argue that the system opens the realm of possibilities that are consistent with the intents expressed by the user, often surprising him/her in the process.

On the downside, as the **f2** runs reveal, in some cases the design intentions are not fully conveyed by the evolved ant paintings. It is also worth mentioning that interactive runs allow opportunistic reasoning, which may allow the discovery of unexpected and highly valued ant paintings.

The adoption of a semi-automatic fitness assignment scheme, such as the one presented by Machado et al. (2005), is one of the directions for further research. It also become obvious that we only began to scratch the vast number of possibilities provided by the design of fitness functions. In the future, we will invite users that are not familiar with the system to design their own fitness functions, which will allow us to assess the difficulty of the task for regular users.

Acknowledgements

This research is partially funded by the Portuguese Foundation for Science and Technology in the scope of project SBIRC (PTDC/EIA–EIA/115667/2009) and of the iCIS project (CENTRO-07-ST24-FEDER-002003), which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER.

References

Aupetit, S.; Bordeau, V.; Monmarché, N.; Slimane, C.; and Venturini, G. 2003. Interactive Evolution of Ant Paintings. In *IEEE Congress on Evolutionary Computation*, volume 2, 1376–1383.

Collomosse, J. P. 2006. Supervised genetic search for parameter selection in painterly rendering. In *Applications of Evolutionary Computing, EvoWorkshops 2006*, 599–610.

Collomosse, J. 2007. Evolutionary search for the artistic rendering of photographs. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer Berlin Heidelberg. 39–62.

Greenfield, G. 2005. Evolutionary methods for ant colony paintings. In *Applications of Evolutionary Computing, EvoWorkshops2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 3449 of *LNCS*, 478–487. Lausanne, Switzerland: Springer Verlag.

Greenfield, G. 2006. Ant Paintings using a Multiple Pheromone Model. In *Bridges*.

Hewgill, A., and Ross, B. J. 2003. Procedural 3d texture synthesis using genetic programming. *Computers and Graphics* 28:569–584.

Karperien, A. 2012. Fraclac for imagej, version 2.5. In *http://rsb.info.nih.gov/ij/plugins/fraclac/FLHelp/Introduction.htm*.

Lewis, M. 2004. Aesthetic video filter evolution in an interactive real-time framework. In *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 3005 of *LNCS*, 409–418. Coimbra, Portugal: Springer Verlag.

Machado, P., and Graça, F. 2008. Evolutionary pointillist modules: Evolving assemblages of 3d objects. In *Applications of Evolutionary Computing, EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoSTOC, and EvoTransLog, Naples, Italy, March 26-28, 2008. Proceedings*, volume 4974 of *Lecture Notes in Computer Science*, 453–462. Springer.

Machado, P., and Pereira, L. 2012. Photogrowth: non-photorealistic renderings through ant paintings. In Soule, T., and Moore, J. H., eds., *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012*, 233–240. ACM.

Machado, P.; Romero, J.; Cardoso, A.; and Santos, A. 2005. Partially interactive evolutionary artists. *New Generation Computing – Special Issue on Interactive Evolutionary Computation* 23(42):143–155.

Machado, P.; Dias, A.; and Cardoso, A. 2002. Learning to colour greyscale images. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour – AISB Journal* 1(2):209–219.

Monmarché, N.; Mahnich, I.; and Slimane, M. 2007. Artificial art made by artificial ants. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer Berlin Heidelberg. 227–247.

Moura, L. 2002. Swarm paintings – non-human. In *ARCHITOPIA Book, Art, Architecture and Science*. 1–24.

Neufeld, C.; Ross, B.; and Ralph, W. 2007. The evolution of artistic filters. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer Berlin Heidelberg. 335–356.

Perlin, K. 1985. An image synthesizer. In Cole, P.; Heilman, R.; and Barsky, B. A., eds., *Proceedings of the 12st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1985*, 287–296. ACM.

Ramos, V., and Almeida, F. 2000. Artificial ant colonies in digital image habitats - a mass behaviour effect study on pattern recognition. In *In Dorigo, M., Middendorf, M., Stuzle, T. (Eds.): From Ant Colonies to Artificial Ants - 2 nd Int. Wkshp on Ant Algorithms*, 113–116.

Ramos, V. 2002. On the implicit and on the artificial - morphogenesis and emergent aesthetics in autonomous collective systems. In *ARCHITOPIA Book, Art, Architecture and Science*. 25–57.

Ross, B. J.; Ralph, W.; and Hai, Z. 2006. Evolutionary image synthesis using a model of aesthetics. In Yen, G. G.; Lucas, S. M.; Fogel, G.; Kendall, G.; Salomon, R.; Zhang, B.-T.; Coello, C. A. C.; and Runarsson, T. P., eds., *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, 1087–1094. Vancouver, BC, Canada: IEEE Press.

Tzafestas, E. 2000. Integrating drawing tools with behavioral modeling in digital painting. In Ghandeharizadeh, S.; Chang, S.-F.; Fischer, S.; Konstan, J. A.; and Nahrstedt, K., eds., *ACM Multimedia Workshops*, 39–42. ACM Press.

Urbano, P. 2005. Playing in the pheromone playground: Experiences in swarm painting. In Rothlauf, F.; Branke, J.; Cagnoni, S.; Corne, D. W.; Drechsler, R.; Jin, Y.; Machado, P.; Marchiori, E.; Romero, J.; Smith, G. D.; and Squillero, G., eds., *EvoWorkshops*, volume 3449 of *Lecture Notes in Computer Science*, 527–532. Springer.

Urbano, P. 2007. Mimetic variations on stigmergic swarm paintings. In Monmarché, N.; Talbi, E.-G.; Collet, P.; Schoenauer, M.; and Lutton, E., eds., *Artificial Evolution*, volume 4926 of *Lecture Notes in Computer Science*, 62–72. Springer.

Urbano, P. 2011. The *t. albipennis* sand painting artists. In *EvoApplications (2)*, volume 6625 of *Lecture Notes in Computer Science*, 414–423. Springer.

Yip, C. 2004. Evolving Image Filters. Master's thesis, Imperial College of Science, Technology, and Medicine.

Adaptation of an Autonomous Creative Evolutionary System for Real-World Design Application Based on Creative Cognition

Steve DiPaola, Graeme McCaig, Kristin Carlson, Sara Salevati and Nathan Sorenson
School of Interactive Arts and Technology
Simon Fraser University
sdipaola@sfu.ca, gmccaig@sfu.ca, kca59@sfu.ca, sara_salevati@sfu.ca, nds6@sfu.ca

Abstract

This paper describes the conceptual and implementation shift from a creative research-based evolutionary system to a real-world evolutionary system for professional designers. The initial system, DarwinsGaze, is a Creative Genetic Programing system based on creative cognition theories. It generated artwork that 10,000's of viewers perceived as human-created art, during its successful run at peer-reviewed, solo shows at noted museums and art galleries. In an effort to improve the system for use with real-world designers, and with multi-person creativity in mind, we began working with a noted design firm exploring potential uses of our technology to support multi-variant creative design iteration. This second generation system, titled Evolver, provides designers with fast, unique creative options that expand beyond their habitual selections that can be inserted/extracted from the system process at any time for modular use at varying stages of the creative design process. We describe both systems and the design decisions to adapt our research system, whose goal was to incorporate creativity automatically within its algorithms, to our second generation system, which attempts to take elements of human creativity theories and populate them as tools back into the process. We report on our study with the design firm on the adapted system's effectiveness.

Introduction

Creativity is a complex set of cognitive process theorized to involve, among other elements, attention shifts between associative and analytical focus (Gabora, 2010), novel goals (Luo and Knoblich, 2007), and situated actions and difficult definitions of evaluation (Christoff et al, 2011). Computational creative systems strive to model a variety of creativity's aspects using computer algorithms from evolutionary 'small-step' modifications to intelligent autonomous composition and 'big-leap' innovation in an effort to better understand and replicate creative process (Boden, 2003). The focus by some researchers on replicating creativity in computational algorithms has been instrumental in learning more about human cognition (individual and collaborative) and how creative support tools might be used to enhance and augment human creative individuals and teams. All these aspects continue to evolve our perceptions of creativity and its role in computation in the current technology-saturated world.

Systems modeling creativity computationally have gained acceptance in the last two decades, situated mainly as artistic and research projects. Several researchers in computational creativity have addressed questions around such computational modeling by outlining different dimensions of creativity and proposing schema for evaluating a "level of creativity" of a given system, for example (Ritchie, 2007; Jennings, 2010; Colton, Pease and Charnley, 2011). While there is ongoing research and scholarly discourse about how a system is realized, how the results are generated, selected and adjusted and how the process and product are evaluated, there is less research about direct applications of creative cognitive support systems in real-world situations. Now that more autonomous, generative creative systems have been developed, we are re-evaluating the role of the human collaborator(s) when designing a creative system for real-world applications in an iterative creative design process environment (Shneiderman, 2007).

We explore creativity from theories of cognition that attempt to understand attentional shifts between associative and analytical focus. The existence of two stages of the creative process is consistent with the widely held view that there are two distinct forms of thought (Dartnell, 1993; Neisser, 1963; Piaget, 1926; Rips, 2001; Sloman, 1996). It has been proposed that creativity involves the ability to vary the degree of conceptual fluidity in response to the demands of any given phase of the creative process (Gabora, 2000; 2002a; 2002b; 2005). This dimension of variability in focus is referred to as contextual focus. Focused attention produces analytic thought, which is conducive to manipulating symbolic primitives and deducing laws of cause and effect, while defocused attention produces fluid or associative thought which is conducive to analogy and unearthing relationships of correlation. Thus, creativity is not just a matter of eliminating rules but of assimilating and then breaking free of them where warranted.

This paper focuses first on the implementation and applicability of contextual focus through our research system, DarwinsGaze, developed to use an automatic fitness function. Second, we present our effort to adapt this successful

but specific research system for more general use with real-world designers, and with multi-person creativity in mind. We worked with a noted design firm to examine potential uses of our technology for supporting multi-variant creative design iteration. Our analysis of their process combined with our knowledge of the cognitive aspects of creativity (gleaned from our early research), were used to completely rewrite the DarwinsGaze system to an interactive creativity support tool within a production pipeline. This 2nd generation system, Evolver, provides designers with fast, unique options that expand beyond their habitual selections that can be inserted and extracted from the system process at any time for modular use at varying stages of the creative design process. The changes focused firstly on usability needs, but became more important when we saw opportunities for affecting the shifts between contextual and analytical focus of the designer through the Evolver system. This process required evaluating the real-world iterative process of designers and testing various prototypes with designers from the firm Farmboy Fine Arts (FBFA) to see how they engaged with interactive creativity support. Lastly we evaluated with a user study the effectiveness of this conversion process and how non-technical designers appreciated and used this Creative Evolutionary System. We hope that our experience and evaluation can be a guide for other researchers to adapt creative research systems to more robust and user centric real world production tools.

The DarwinsGaze System

The DarwinsGaze system (DiPaola and Gabora, 2007) is a Creative Evolutionary System (CES) (Bentley and Corne, 2002) (see Figure 1) based on a variant of Genetic Programming (GP). Unlike typical Genetic Programming systems this system favors exploration over optimization, finding innovative or novel solutions over a preconceived notion of a specific optimal solution. It uses an automatic fitness function (albeit one specific to portrait painting) allowing it to function without human intervention between being launched and obtaining the final, often unanticipated and pleasing set of results; in this specific and limited sense we refer to DarwinsGaze as "autonomous". The inspiration for this work is to directly explore to what extent computer algorithms can be creative on their own (Gabora and DiPaola, 2012). Related work has begun to use creative evolutionary systems with automatic fitness functions in design and music (Bentley and Corne, 2002), as well as building of a creative invention machine (Koza, 2003). A contribution of the DarwinsGaze work is to model, in software, newly theorized aspects of human creativity, especially in terms of fluid contextual focus (see Figure 2).

DarwinsGaze capitalizes on recent developments in GP by employing a form of GP called Cartesian Genetic Programming (CGP) (Miller and Thomson, 2000; Walker and Miller, 2005). CGP uses GP techniques (crossover, mutation, and survival), but differs in certain key respects. The

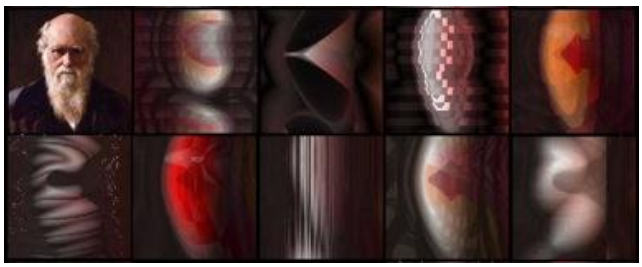


Figure 1. Source Darwin image with examples of evolved abstract portraits created using the DarwinsGaze autonomous creative system.

program is represented by a directed graph of indexed nodes. Each node has a number of inputs and a function that gives an output based on the inputs. The genotype is a list of integers determining the connectivity and functionality of the nodes, which can be mutated and mated to create new directed graphs.

CGP has several features that foster creativity including 1) its node based structure facilitates the creation of visual mapping modules, 2) its structure can represent complex computational input/output connectivity, thus accommodating our sophisticated tone and temperature-based color space model which enables designerly decision making, and most importantly 3) its component-based approach favors exploration over optimization by allowing different genotypes to map to the same phenotype. The last technique uses redundancy at the input, node, and functional levels, allowing the genotype to contain nodes that are not connected to the output nodes and so not expressed in the phenotype. Having different genotypes (recipes) map to the same phenotype (output) provides CGP with greater neutrality (Yu and Miller, 2005). Our work is based on Ashmore and Miller's (2004) CGP application to evolve visual algorithms for enhanced image complexity or circular objects in an image. Most of their efforts involve initializing a population and then letting the user take over. Our initial prototype was based upon their approach, but expanded it with a more sophisticated similarity and creativity function, and revised their system for a portrait painter process.

Since the advent of photography, portrait painting has not just been about accurate reproduction, but also about using modern painterly goals to achieve a creative representation of the sitter. We have created a fitness function that mainly rewards accurate representation, but given certain situations it also rewards visual painterly aesthetics using simple rules of art creation as well as a portrait knowledge space. Specifically, the painterly portion of our fitness function 1) weighs for face versus background composition, 2) uses tonal similarity over exact color similarity matched with a sophisticated artistic color space model which weighs for warm-cool color temperature relationships based analogous and complementary color harmony rules and 3) employs unequal dominate and subdominant tone and color rules and other artistic rules based on a portrait painter knowledge domain (DiPaola and Gabora, 2007) as illustrated in Figure 2. We mostly weight heavily

towards resemblance, which gives us a structured system, but can under the influence of functional triggers allow for artistic creativity. The approach gives us novelty and innovation from within, or better said, responding to a structured system -- a trait of human creative individuals.

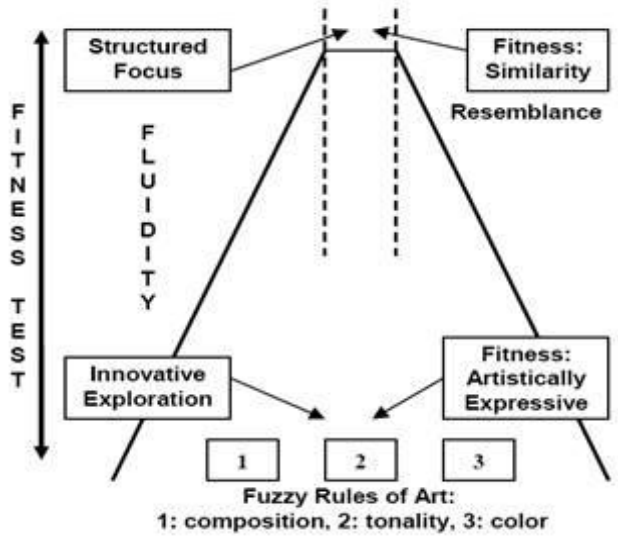


Figure 2. The DarwinsGaze fitness function mimics human creativity by moving between restrained focus (resemblance) to more unstructured associative focus (resemblance and more ambiguous art rules of composition, tonality and color theory).

Generated portrait programs in the beginning of the run will look less like the sitter but from an aesthetic point of view might be highly desirable, since the function set has been built with painterly rules. Specifically, the fitness function in the DarwinsGaze system calculates four scores (resemblance and the three painterly rules) separately and fluidly combines them in different ways to mimic human creativity by moving between restrained focus (resemblance) to more unstructured associative focus (3 rules of composition, tonality and color theory). In its default state the fitness function uses a ratio of 80% resemblance to 20% non-proportional scoring of our three painterly rules. Several functional triggers can alter this ratio in different ways. The system will also allow very high scoring of painterly rule individuals to be accepted into the next population. When a plateau or local minima is reached for a certain number of epochs, the fitness function ratio switches course where painterly rules are weighted higher than resemblance (on a sliding scale) and work in conjunction with redundancy at the input, node, and functional levels. Using this method, in the wider associative mode, high resemblance individuals are always part of the mix and when these individuals show a marked improvement in resemblance, a trigger is set to return to the more focused 80/20 resemblance ratio.

For CES used to create fine art paintings, the evaluation was based less on the process and more on the output. Could a closed process (that has no human intervention once the evolutionary process was started) produce artwork that was judged as creative using the methods by which real human artists are judged? Example pieces from the output over 30 days were framed and submitted to galleries as a related set of work. Care was taken by the author to select representational images of the evolved unsupervised process, however creative human bias obviously exists in the representational editing process. This is similar to how a curator chooses a subset of pieces from their artists, so it was deemed that is does not diminish the soft evaluation process.

The framed art work (darwinsgaze.com) was accepted and exhibited at six major galleries and museums including the TenderPixel Gallery in London, Emily Carr Galley in Vancouver, and Kings Art Centre at Cambridge University as well as the MIT Museum, and the High Museum in Atlanta, all either peer reviewed, juried or commissioned shows from institutions that typically only accept human art work. This gallery of abstract portraits of Darwin has been seen by tens of thousands of viewers who have commented with dated quotes in a gallery journal that they see the artwork as an aesthetic piece that ebbs and flows through creative ideas even though they were solely created by an evolutionary art computer program using contextual focus. Note that no attempt to create a formalized 'creativity Turing Test' was made. Most of the thousands of causal viewers assumed they were looking at human created art. The work was also selected for its aesthetic value to accompany an opinion piece in the journal Nature (Padian, 2008), and was given a strong critical review by the Harvard humanities critic, Browne (2009). While these are subjective measures, they are standards in the art world. The fact that the computer program produced novel creative artifacts, both as single art pieces and as a gallery collection of pieces with interrelated themes, is compelling evidence that the process passed a type of informal creativity Turing test.

The Shift from Autonomous Creative System to Creative Support Tool: the Evolver System

To move forward from the DarwinsGaze system we began looking to explore a real-world application of creativity in computation by leveraging concepts of contextual focus to integrate with collaborative process. The opportunity arose to work with FBFA, an international art consultancy firm that designs site-specific art collections for the luxury hotel and corporate sectors, to develop software that could complement and provoke their current iterative design processes. The focus on visual design for hotel decor was an interesting perspective that enabled us to consider what we had achieved with visual creative design in prior work, and how we could engage in the designer's intuitive yet visual (and hence somewhat parameterized) creative process.

In the effort to evaluate a CES within a Visual Design domain, we explored the use and adaptation of "Evolver". Evolver is a computational creative tool modified from the DarwinsGaze project structure. Evolver was created as a result of in-depth research and observations to support a specific design process at FBFA by automating some of the design tasks and restructuring the contextual search space. It provides a platform for brainstorming by generating various versions of original artwork provided by designers, through specific features such as controlling the color scheme or marrying different artworks together. It also offers some production capabilities by automating repeating tasks such as cropping for mass quantities of artworks traditionally performed by designers in programs such as Adobe Photoshop. Evolver incorporates a user-friendly GUI (see Figure 3) paired with a flexible internal image representation format for ease of use by the designer. The designer provides the seed material and selects preferred results while the system generates a population of artwork candidates, cross breeds and mutates the candidates under user control to generate new design products. The designer may select and extract any resulting candidate piece at any stage of the process for use in other areas or as generative fodder to later projects. System parameters of Evolver include shapes, colors, layers, patterns, symmetries and canvas dimensions.

Developing the Evolver System to Fit the Needs and Process of a Design Firm

FBFA takes design briefs from the hotel interior designers, and based on their extensive photo and graphic design database as source, designs specific art and design objects in a multitude of material (although typically wall hanging) often in unique sizes, shapes and multiples to specifically work with the hotel's (typically their large lobby and restaurants) needs. They do this by incorporating a number of designers who using digital systems like Adobe Illustrator significantly rework a source design to refit the space, shape and material specifics.

We began by demonstrating to them an interactive version of our DarwinsGaze system, which was mocked up on the darwinsgaze.com website, called 'Evolve It' to show what a potentially fully-interactive new system would look like. The designer's process to create a successful prototype for the client was a multi-step, iterative and somewhat inefficient process which relied on the designer's 'feel' of the problem context, the potential solution contexts and their intuitive exploration and selection process. In this particular situation designers would discuss a project with a client, then go to physical boxes or their digital database containing immense amounts of image material, find seed material that fits the feeling of the multiple contexts and then manipulate them to better fit the design problem in Adobe Illustrator. The designer's manipulation adjusts size, scale, shape, multiples and color in layers by hand. This process is highly labor-heavy and we felt it was most receptive to computational support because the designer had already defined the contextual focus for this problem through their own interpretation of the available options, constraints and aesthetic preference (which had already been confirmed by the client engaging with this company).

While the designers were reluctant to give up control of their intuitive, creative knowledge, they readily engaged with the Evolver system once they saw how CESs could support the restructuring of the designer's contextual space while also reducing the labor-intensive prior process. This shift freed up the designer's ability to creatively engage with the problem at hand. We strove to make the new systems flexible to different creative processes and paths that different designers might have.

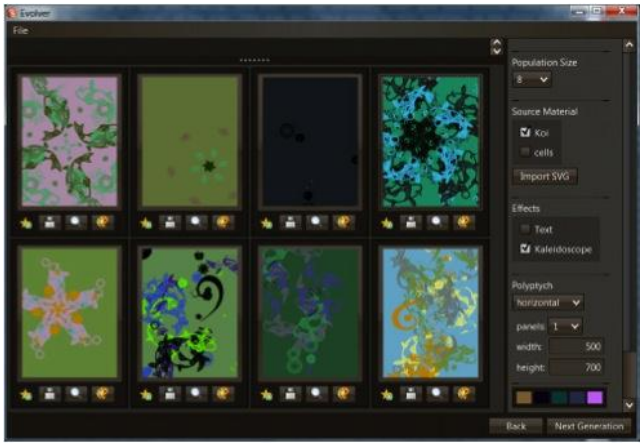


Figure 3. The Evolver Interface

Evolver's cognitive aspect provides designers with a platform to externalize and visualize their ideas. Artwork generated through Evolver can be used for different purposes in different phases of the design process, from conceptual design through to presentation. During the early phase of conceptual design, free-hand, non-rigid sketching techniques have an important role in the formation of creative ideas as designers externalize their ideas and interact with them spatially and visually (Suwa, Gero and Purcell, 1998). Evolver supports flexibility of ideas in this phase by enabling designers to easily produce an extensive range of alternatives. The ambiguous nature of the multiple generations produced supports the uncertain and fuzzy nature of conceptual design as they discover, frame out early ideas and brainstorm. The alternatives produced relieve cognitive load from the designer by separating them from the manual task of manipulating the design parameters, but do not separate them so far from the process that they cannot use their psychomotor and affective design knowledge.

Evolver is structured to support the shift between contextual and analytical focus by restructuring the contextual space users are working in. Users can choose to relinquish a degree of control while broadening their focus, gaining the ability to be inspired or provoked by novel generations from the system. On the other hand, it is possible to guide successive evolutions in a more deliberate, analytical way and the ability of Evolver to import/export individuals

to/from a precisely editable format (SVG - Adobe Illustrator) allows tightly focused design directions to be pursued. At later stages in the design process, artwork generated through Evolver can be used as mockups for clients and prototyping, and also as a communication tool in uses such as presentation at the very end of design process. The work produced by Evolver can be incorporated directly into the tool-chain leading to a finished piece.

Evolver Genetic Encoding: Moving to a More "Linear" Scheme

One of the most far-reaching design decisions involved in the construction of an evolutionary system is the specification of the genetic encoding. A particular choice of encoding delineates the space of possible images and dramatically influences the manner in which images can change during the course of evolution. The genotype induces a metric on the space of potential images: certain choices of representation will cause certain styles or images to be genetically related and others to be genetically distant. The related images will appear with much more probability, even if the distant images are technically possible to represent in the encoding system. For this reason, it is important that the genotype causes images that are aesthetically similar to be genetically related. Relevant aspects of the aesthetic merit of a work can then be successfully selected for and combined throughout the course of the evolutionary run. This property is referred to as gene linkage (Harik et al, 2006). We identified this property as especially important to an interactive creativity support tool, for designers who are used to exerting a high degree of creative control over their output and in a scenario where a certain sense of "high quality design" is to be maintained.

A genetic encoding can either be low level, representing extremely basic atomic elements such as pixels and color values, or high level, representing more complex structures such as shapes and patterns. A common low level encoding is to represent images as the composition of elemental mathematical functions (Bentley and Corne 2002). Though it is technically possible that any image can be conceivably represented as a composition of such functions, this encoding typically results in recognizable geometric patterns that readily signal the algorithmic nature of the process. A higher level encoding can be seen in shape grammars that represent not individual pixels but aggregates of primitive shapes (Machado et al, 2010). This approach can theoretically produce a much narrower range of images, but the images that are produced do not demonstrate the same highly-mathematical nature of lower-level encodings. Compared to the CGP genetic structure of DarwinsGaze, Evolver uses a list-based, tree-structure encoding that draws some inspiration from CGP but operates on higher-level components in order to maximize the property of gene linkage and user interpretability.

We viewed this new genetic representation as broadly "linear" in the sense that the genotype could be decomposed into elements and recombined, leading to a corresponding effect in the phenotype of recombining visually identifiable elements. The genetic representation is based on a collection of "design elements" (DEs), which are objects that denote particular aspects of the image. For example, a major component of our image representation is that of a symbol: a shape that can be duplicated and positioned on the canvas according to a position, rotation, and scaling parameter. DEs are defined in terms of atomic values and composite collections. The DE for a symbol, for example, is represented as a tuple consisting of two floats representing the x and y coordinates of the shape, a float representing the rotation, a float representing the scale, and an enumerable variable representing the particular shape graphic of the symbol. An image is then described by a list of these symbols. The genetic operations of mutation and crossover are derived from the structure of the DE definitions. Mutation is defined for the atomic values as a perturbation of the current value. Crossover is defined for the collection structures. The genotype is "strongly typed" so only genes of the same type can cross over. (For example, "position" may cross over with "position" of another other stamp's record, "color" may cross over with "color"; however "position" will never cross over with "color".) Figure 4 shows an example of Evolver system output.

Evolver User Interface: Optimizing Creative Support

To make the power of this flexible encoding system available to designers, we constructed an automatic import tool that analyzed existing images and parsed their structure into DEs that formed initial seed populations for the interactive evolution. This approach served to bootstrap the evolutionary search with images that are known to demonstrate artistic merit. Source artwork is converted to the SVG vector image format, which is a tree-based description of the shapes and curves that comprise a vector based image. The hierarchical grouping of art elements in the original work is preserved in the SVG format, and is used in determining which pieces are isolated to form symbol DEs. We also make use of heuristics that take into account the size of various potential groupings art elements and any commonly duplicated patterns to identify candidates for extraction.

The interactive evolution proceeds from a seed population constructed from these original parsed image elements. The user interface, by default, depicts a population of 8 pieces of generated art. These individuals can be selected from, to become the parents of the next generation, as is typical in interactive evolution. An added feature, which proved useful, was the ability to bookmark individuals, which placed them in a different collection that was separated from the evolutionary run. This collection of bookmarked individuals allowed users to store any

interesting images discovered during the run while proceeding to guide the evolution in a different direction.

Figure 4. Example Evolver Output Image

Evaluating Designers' Usage and Opinions of the Evolver System

Some months after the end of the project, with Evolver still being used and available for real world production at FBFA, we invited a small group of FBFA and associated designers to our labs, now under controlled study conditions. There we conducted a 45 minute questionnaire-based qualitative study that took place in 2 phases: it began with a uniform re-introduction and re-demonstration of Evolver and its functionalities, followed by a short session where the designer had the opportunity to re-explore the tool and answer a series of nine structured interview questions that concentrate on the adaptation of Evolver within their current and future work practices. The specific questions in phase two were:

1. What is your first impression of 'Evolver'?
2. How and in which stage would you use this tool in your current practice?
3. How does this tool change your design process? Can you provide an existing scenario of your current practice and how you envision Evolver would change that?
4. Which features of this tool do you find most interesting? Why?
5. What features would you like to change and/or add in the future? Why?
6. How would you use this tool apart of your design thinking stage in your process?
7. How does it help with the conceptualization of ideas?
8. What do you think of the role of computational tools such as Evolver within the Visual Design domain?
9. Do you have any further comments/suggestions for the future of this research?

The full qualitative study discursive results are beyond the scope of this paper; however we have included an exemplary set of these results, based on direct quotes from the designers and our assessment of the dominant themes in designer responses. Our main takeaways from this study were:

1. Designers saw Evolver as a creative partner that could suggest alternatives outside of the normal human cognitive capacity:
   "[The] Human brain is sometimes limited, I find Evolver to have this unlimited capacity for creativity." (KK, Interview)
   "Evolver introduces me to design options I never thought of before, it enhances my design thinking and helps me to produce abstract out of the norm ideas." (LA, Interview)
2. Evolver also enhanced the human user's ability to enter a more intuitive or associative mode of thought by easing some of the effort in manually visualizing alternative design concepts:
   "Sketching stuff out on paper takes more energy and tweaking - Evolver allows me to visualize easier, have a dialogue and collaborate with the design space." (RW, Interview)
3. Evolver could be used flexibly at different stages of the design process to support different tasks and modes of thought, including both generation and communication of ideas
   "The best part about the Evolver is that you can stop it at any stage of generation, edit and feed it back to the engine, also it is mobile and you can take it to meetings with clients and easily communicate various ideas and establish a shared understanding. It provides a frame of reference- what is in your head now." (RW, Interview)

Comparison and Discussion

We compare the details of the decisions made to shift from the autonomous DarwinsGaze system to the interactive Evolver system and describe their importance (see Table 1). One of the first changes was to shift the genetic representation (or the 'gene' structure). The DarwinsGaze system has genes which work together in a tree structure, to evolve output as a bitmap of the whole piece. The Evolver System genes were more linear and 'predictably recombinable' in order to minimize contextual focus within the system while prioritizing a variety of potentially successful solutions. DarwinsGaze used automatic fitness function-based Cartesian Genetic Programing while Evolver shifted to a simpler and interactive Genetic Algorithm in order to engage the designer in the system and support their intuitive decision-making process. In DarwinsGaze there is no control over pieces, layers or options for interaction involvement. The Evolver system has many layers and elements and is built on the standards based vector language (SVG). Using a design-shelf structure the user has more subtle control including feature navigation, text, symmetry and rotation. The user can either import many small SVG files as seed material or import a single large file and the

system will automatically separate and label the elements. With the user acting as the fitness function, the population size can be adjusted and desired results can be 'bookmarked' and set aside for manual iteration or can be re-inserted into the Evolver system's gene pool. So for instance, work that they create traditionally can be used as partial seed material, used fully at the start, output at any time from the system as raw inspiration results to be reworked traditionally or used as a final result. A careful effort was made to iteratively develop the graphical user-interface based on feedback from the designers about how they think within a creative process, what metaphors they use, and which perspectives and skills they rely on based on their backgrounds and experience. Finally we integrated additional post-processing options to give added novelty if needed (outside of the Genetic Algorithm) with effects such as kaleidoscope and multiple panels.

DarwinsGaze System	Evolver System
Genes specific to image resemblance & art rules	Genes linear, strong typed, focus on existing parameters
Automatic CGP: complex FF / functional triggering	Interactive Genetic Algorithm: simple structured forms
Bitmap, evolve-as-a-whole	SVG, evolve as labeled layers
Operates autonomously, no import/export material	Ability to import/export labeled semantic material – HCI based
Research system with specific evolve towards the sitter images goals	Communicates at any point of process with trad. design tools supporting wide creative styles
Innovative / complex auto functional triggers : analytical to associative & back	Simpler user-interaction: population size, bookmarks to support human creative triggers
One system : full process of creativity, no external communication	Integrated system: built to work w/ other tools, processes; supports creativity as an adaptive human process
Informed by creativity theory and simulates it internally in complex ways	Informed by creativity theory but uses it to support a real world meta system w/humans

Table 1. Comparison Between DarwinsGaze and Evolver Systems

The study of Evolver in use also made apparent an attitude shift of visual designers towards CESs, which change their role from sole creators to editors and collaborators. The designers became more receptive of tools such as Evolver as they came to view them not as replacing designers or automating the creative process; but rather as promoting new ways of design thinking, assisting and taking designer's abilities to the next level by providing efficiency and encouraging more 'aha' moments. The visual designers in the study described Evolver as an "invisible teammate", who they could collaborate with at any stage of their design process. Evolver became a center of dialogue among designers and helped them communicate their mental models and understanding of design situations to clients and other stakeholders.

Conclusions

Many significant research CES systems exist that are both innovative and useful. However as the field matures, there will be an increasing need to make CESs production worthy and work within a creative industry environment such as a digital design firm. To support others in this effort for production-targeted transformation, in this paper we described the shift from an autonomous fitness function based creative system, DarwinsGaze, to an interactive fitness function based creative support system, Evolver, for real-world design collaboration. DarwinsGaze operates using a complex automatic fitness function to model contextual focus as well as other aspects of human creativity simulated internally. In shifting to the Evolver project we found that the contextual focus perspective remained relevant, but now re-situated to overlay the collaborative process between designer and system. Four design principles developed on this basis were: 1) support analytic focus by providing tools tailored to the designer's specific needs and aesthetic preferences, 2) support associative or intuitive focus by relieving the designer's cognitive capacity, enabling a quick and serendipitous workflow when desired, and offering a large variety of parameterized options to utilize, 3) support a triggering of focus-shift between the designer and the system through options to 'bookmark' and save interesting pieces for later, as well as to move creative material from and to the system while retaining the work's semantic structure and editability, and 4) support a joint 'train of thought' between system and user by structuring a genotype representation compatible with human visual/cognitive intuition.

We found that the shift to a real-world design scenario required attention to the collaboration and creative processes of the designers who value their experience-developed expertise. The system design had to act as both a support tool engaging some cognitive load of the process, and a flexible, interactive repository of potentially successful options. Future real-world design considerations can explore methods for adapting intelligent operations to the cognitive processes and constraints of necessary situations, taking into account the expertise of collaborators.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada and Mitacs (Canada). We would like to thank the design firm Farmboy Fine Arts, Liane Gabora, Nahid Karimaghalou, Robb Lov-

ell and Sang Mah for agreeing to work on the industrial/academic partnership part of the work.

References

Ashmore, L., and Miller, J. 2004. Evolutionary Art with Cartesian Genetic Programming. Technical Online Report. http://www.emoware.org/evolutionary_art.asp.

Bentley, P., and Corne, D. eds. 2002. Creative Evolutionary Systems, San Francisco, CA.: Morgan Kaufmann.

Boden, M. 2003. The Creative Mind. London: Abacus.

Brown, J. 2009. Looking at Darwin: portraits and the making of an icon. Isis. Sept, 100(3):542–70.

Dartnell, T. 1993. Artificial intelligence and creativity: An introduction. Artificial Intelligence and the Simulation of Intelligence Quarterly 85.

DiPaola, S. and Gabora, L. 2007. Incorporating characteristics of human creativity into an evolutionary art algorithm. In (D. Thierens, Ed.), Proc Genetic and Evol Computing Conf , 2442– 2449. July 7-11, Univ College London.

Feist, G.J, 1999. The influence of personality on artistic and scientific creativity, in Handbook of Creativity, R.J. Sternberg, Ed. (Cambridge University Press, Cambridge, UK)

Gabora, L. 2000. Toward a theory of creative inklings. In (R. Ascott, Ed.) Art, Technology, and Consciousness, Intellect Press, Bristol, UK.

Gabora, L. 2002. The beer can theory of creativity, in (P. Bentley and D. Corne, Eds.) Creative Evolutionary Systems, 147-161. San Francisco, CA.: Morgan Kaufmann.

Gabora, L.2002. Cognitive mechanisms underlying the creative process. In (T. Hewett and T. Kavanagh, Eds.) Proceedings of the Fourth International Conference on Creativity and Cognition, Oct 13-16, UK, 126-133.

Gabora, L. 2005.Creative thought as a non-Darwinian evolutionary process. Journal of Creative Behavior, 39(4), 6587.

Gabora, L. 2010. Revenge of the "neurds": Characterizing creative thought in terms of the structure and dynamics of memory. Creativity Research Journal, 22(1), 1-13.

Gabora, L., and DiPaola, S. 2012. How did humans become so creative? Proceedings of the International Conference on Computational Creativity, 203-210. May 31 - June 1, Dublin, Ireland.

Harik, G. R., Lobo, F. G., and Sastry, K. 2006. Linkage Learning via Probabilistic Modeling in the Extended Compact Genetic Algorithm (ECGA). In D. M. Pelikan, K. Sastry, & D. E. CantúPaz (Eds.), Scalable Optimization via Probabilistic Modeling, 39–61. Springer Berlin Heidelberg.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. Minds and Machines, 1– 13.

Koza, J R., Keane, M A., and Streeter, M J. 2003. Evolving inventions. Scientific American. February. 288(2) 52 – 59.

Lawson, B. 2006. How designers think: the design process demystified (4th Edition). Oxford: Elsevier.

Luo, J., and Knoblich, G. 2007. Studying insight with neuroscientific methods. Methods, 42, 77-86.

Machado, P., Nunes, H., & Romero, J. 2010. Graph-Based Evolution of Visual Languages. In C. D. Chio, A. Brabazon, G. A. D. Caro, M. Ebner, M. Farooq, A. Fink, N. Urquhart (Eds.), Applications of Evolutionary Computation, 271–280. Springer Berlin Heidelberg.

Miller, J. 2011. Cartesian Genetic Programming, Springer.

Miller, J., and Thomson, P. 2000. Cartesian Genetic Programming. Proceedings of the 3rd European Conference on Genetic Programming, 121-132. Edinburgh, UK.

Neisser, U. 1963. The multiplicity of thought. British Journal of Psychology, 54, 1-14.

Padian, K. 2008. Darwin's enduring legacy. Nature, 451(7179), 632–634.

Piaget, J. 1926. The Language and Thought of the Child. Routledge and Kegan Paul, London.

Pease, A. and Colton, S. 2011. On Impact and Evaluation in Computational Creativity: A Discussion of the Turing Test and an Alternative Proposal. In Proceedings of the AISB symposium on AI and Philosophy .

Rips, L. J. 2001. Necessity and natural categories. Psychological Bulletin, 127(6), 827-852.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. Minds and Machines 17.

Shneiderman, B. 2007. Creativity support tools: accelerating discovery and innovation. Commun. ACM, 50(12), 20–32.

Schön, D. 1988. Designing: Rules, Types, and Worlds. Design Studies, 9/3 181-4.

Sloman, S. 1996. The empirical case for two systems of reasoning. Psychol. Bull. 9(1), 3–22.

Suwa, M., Gero, J. S., and Purcell, T. 1998. The roles of sketches in early conceptual design processes. In Proceedings of Twentieth Annual Meeting of the Cognitive Science Society, 1043-1048.

Walker, J. and Miller, J. 2005. Improving the Evolvability of Digital Multipliers Using Embedded Cartesian Genetic Programming and Product Reduction. Evolvable Systems: From Biology to Hardware, 6th International Conference, ICES 2005, Proceedings, Sitges, Spain. Springer.

Yu, T., and Miller, J. 2001. Neutrality and the Evolvability of Boolean function landscape. Proceedings of the Fourth European Conference on Genetic Programming, 204-217. Berlin Springer-Verlag.

A Computational Model of Analogical Reasoning in Dementia Care

Konstantinos Zachos and Neil Maiden
Centre for Creativity in Professional Practice
City University London
Northampton Square, London EC1V 0HB, UK
{k.zachos, N.A.M.Maiden}@city.ac.uk

Abstract

This paper reports a practical application of a computational model of analogical reasoning to a pressing social problem, which is to improve the care of older people with dementia. Underpinning the support for carers for people with dementia is a computational model of analogical reasoning that retrieves information about cases from analogical problem domains. The model implements structure-mapping theory adapted to match source and target domains expressed in unstructured natural language. The model is implemented as a computational service invoked by a mobile app used by carers during their care shifts.

Dementia Care and Creativity

Dementia is a condition related to ageing. After the age of 65 the proportion of people with dementia doubles for every 5 years of age so that one fifth of people over the age of 85 are affected (Alzheimers Society 2010). This equates to a current total of 750,000 people in the UK with dementia, a figure projected to double by 2051 when it is predicted to affect a third of the population either as a sufferer, relative or carer (Wimo and Prince 2010). Dementia care is often delivered in residential homes. In the UK, for example, two in three of all home residents have some form of dementia (e.g. Wimo and Prince 2010), and delivering the required care to them poses complex and diverse problems carers that new software technologies have the potential to overcome. However, this potential is still to be tapped.

The prevailing paradigm in dementia care is person-centered care. This paradigm seeks an individualized approach that recognizes the uniqueness of each resident and understanding the world from the perspective of the person with dementia (Brooker 2007). It can offer an important role for creative problem solving that produces novel and useful outcomes (Sternberg 1999), i.e. care activities that both recognize a sense of uniqueness and are new to the care of the resident and/or carer. However, there is little explicit use of creative problem solving in dementia care, let alone with the benefits that technology can provide. Therefore, the objective of our research was to enable more creative problem solving in dementia care through new software technologies.

This paper reports two computational services developed to support carers to manage challenging behaviors in person-centered dementia care – a computational analogical matching service that retrieves similar challenging behavior cases in less-constrained domains, and a second service that automatically generates creativity prompts based on the computed analogical mappings. Both are delivered to carers through a mobile software app. The next two sections summarize results from one pre-design study that motivates the role of analogical matching in managing challenging behavior in dementia care then describe the two computational creativity services.

A Pre-Design Study

Creative problem solving is not new to care work. Osborn (1965) reported that creative problem solving courses were introduced in nursing and occupational therapy programs in the 1960s. Le Storti et al. (1999) developed a program that fostered the personal creative development of student nurses, challenging them to use creativity techniques to solve nursing problems. This required a shift in nursing education from task- to role-orientation and established a higher level of nursing practice – a level that treated nurses as creative members of health care teams. There have been calls for creative approaches to be used in the care of people with dementia. Successful creative problem solving was recognized to counteract the negative and stressful effects that are a frequent outcome of caring for people with dementia (Help the Aged, 2007). Several current dementia care learning initiatives can be considered creative in their approaches. These include the adoption of training courses in which care staff are put physically into residents' shoes, and exercises to encourage participants to experience life mentally through the eyes of someone with dementia (Brooker 2007). Caring for people with late stage dementia is recognized to require more creative approaches, and a common theme is the need to deliver care specific to each individual's behavioral patterns and habits.

To discover the types of dementia care problem more amenable to this model of creative problem solving, we observed care work and interviews with carers at one UK residential home revealed different roles for creative problem solving in dementia care. One of these roles was to

reduce the instances of challenging behavior in residents. Challenging behavior defined as "*culturally abnormal behavior(s) of such an intensity, frequency or duration that the physical safety of the person or others is likely to be placed in serious jeopardy, or behavior which is likely to seriously limit use of, or result in the person being denied access to, ordinary community facilities*" (Bromley and Emerson 1995). Examples include the refusal of food or medication, and verbal aggression.

Interviews with carers revealed that creative problem solving has the potential to generate possible solutions to reduce instances of challenging behavior. For example, if a resident is uncooperative with carers when taking medication, one means to reduce it might be to have a carer wear a doctor's coat when giving the medication. The means is creative because it can be useful, novel to the resident if not applied to him before, and novel to the care team who have not applied it before. Therefore, with carers in the pilot home, we explored the potential of different creativity techniques to reduce challenging behavior.

During one half-day workshop with 6 carers we explored the effectiveness and potential of different creativity techniques to manage a fictional challenging behavior. During a three-stage process the carers were presented with the fictional resident and challenging behavior, generated ideas to reduce the behavior, then prepared to implement these ideas. They used different creativity techniques, presented to them as practical problem solving techniques, to reduce the fictional challenging behavior. The carers demonstrated the greatest potential and appetite for the other exploratory creativity technique, called *Other Worlds* (Innovation Story 2002). During the workshop, the carers sought to generate ideas to reduce the challenging behavior in four different, less constrained domains - *social life*, *research*, *word of mouth* and *different cultures*. These ideas were then transferred to the care domain to explore their effectiveness in it. *Other Worlds* was judged to be the most effective as well as the most interesting to carers. It created more ideas than any of the other techniques, and two of the ideas from the session were deemed sufficiently useful to implement in the pilot home immediately. Carers singled out the technique because, unlike others, it purposefully transferred knowledge and ideas via similarity-based reasoning from sources outside of the immediate problem spaces – the resident, residential home and dementia care domain.

The Carer App

To implement *Other Worlds* in care work we decided to develop a mobile software app, called *Carer*, which carers can use during their work. In the place of human facilitation, the software retrieves then guides carers to explore other worlds that are retrieved by the app, and in place of face-to-face communication, the software was to support asynchronous communication between carers who would digitally share information about care ideas and practices via the software.

The Carer app accesses a digital repository to retrieve natural language descriptions of cases of good care practice in XML based on the structure of dementia care case studies reported by the Social Care Institute for Excellence (Owen and Meyer 2009) as well as challenging behavior cases in non-care domains such as *teen parenting*, *student mentoring* and *prison life*. Each case has two main parts of up to 150 words of prose each – the situation encountered and the care plan enhancement applied – and is attributed to one class of domain to which the case belongs. The current version of the repository contains 115 case descriptions.

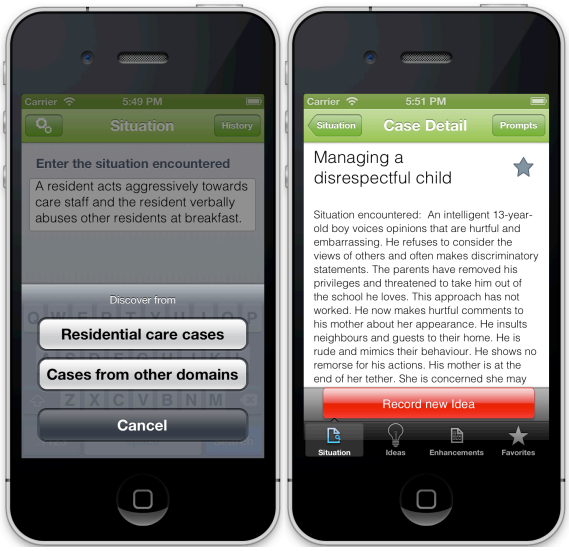


Figure 1. The Carer mobile app showing how carers describe challenging behaviors (on the left-hand side) and a detailed description of one of these cases (on the right-hand side)

Carer app automatically retrieves the previous cases using different services in response to natural language entries typed and/or spoken by a carer into the app. One supports case-based reasoning with literally similar cases based on information retrieval techniques, similar to strategies applied to people with chronic diseases (Houts et al. 1996). A second supports the other worlds technique more generally by automatically generating different domains such as *traveling* or *cooking* in which to generate care plan enhancements to a current situation without the constraints of the care domain (Innovation Company 2002). The user is encouraged to think about how to solve the aggression situation *in the school playground*. A simple flick of the screen will generate a different other world, such as *parachuting from an aircraft*. A third service automatically generates creativity prompts from retrieved case content. Lastly, the *Carer* app invokes AnTiQue, an *analogical reasoning* discovery service that matches the description of a challenging behavior situation to descriptions in the repository of challenging behavior cases in non-care domains. To do this, the service implements a computational analogical reasoning algorithm based on the Structure-

Mapping Theory (Gentner 1983; Falkenhainer et al. 1989) with natural language parsing techniques and a domain-independent verb lexicon called VerbNet (Kipper et al. 2000). A carer can then record new ideas resulting from creative thinking in audio form, then reflect on them by playing them back to change them, generate further ideas, compose them into a care plan and share the plan with other carers. Some of these features are depicted in Figure 1. The right-hand side of Figure 1 shows one retrieved analogical case description – *Managing a disrespectful child* – as it is presented to a carer using the app. The *Carer* app is described at length in Maiden (2012). The next section describes two of the computational creativity services – the analogical reasoning discovery service and the creativity prompt generation service.

The Analogical Reasoning Discovery Service

This service (called AnTiQue) matches a description of challenging behavior in dementia care to descriptions of challenging behavior problems and resolutions in other domains, for example *good policing practices to manage disorderly revelers* and good *teaching practices to manage disruptive children*. AnTiQue's design seeks to solve 2 research problems: (i) match incomplete and ambiguous natural language descriptions of challenging behaviour in dementia care and challenging behaviour problems and resolutions in other domains using different lexical terms; (ii) compute complex analogical matches between descriptions without a priori classification of the described domains.

Analogical service retrieval can increase the number of cases that are useful to the care staff by retrieving descriptions of cases solved successfully in other domains, for example *good policing practices to manage disorderly revelers* and good *teaching practices to manage disruptive children*. The problem and solution description of each case might have aspects that, through analogical reasoning, can trigger discovery of new ideas on the current challenging behaviour. For example, a description of good policing practice to manage disorderly revellers can provide analogical insights with which to manage challenging behaviour in dementia care. AnTiQue seeks to leverage these new sources of knowledge in dementia care.

Analogical retrieval in AnTiQue uses a similarity model called the Structure Mapping Theory (SMT) (Gentner 1983) which seeks to transfer a network of related facts rather than unrelated one (Gentner 1983) from a source to a target domain. To enable structure-matching AnTiQue transforms natural language queries and case descriptions into predicates that express prepositional networks of nodes (objects) and edges (predicate values). Attributional predicates state properties of objects in the form PredicateValue(Object) such as *asleep(resident)* and *absent(relative)*. Relational predicates express relations between objects as PredicateValue (Object1,Object2) such as *abuse(resident, care-staff)* and *remain(resident,room)*. According to the SMT, a literal similarity is a comparison in which attributional and relational predicates can both be mapped from a source to a target. In contrast an analogy is a comparison in which relational predicates but few or no attributional predicates can be mapped. Therefore AnTiQue retrieves cases with high match scores for relational predicates and low match scores for attributional predicates, for example a match with the predicate *abuse(detainee,police-officer)* but no match with the predicate *drunk(detainee)*.

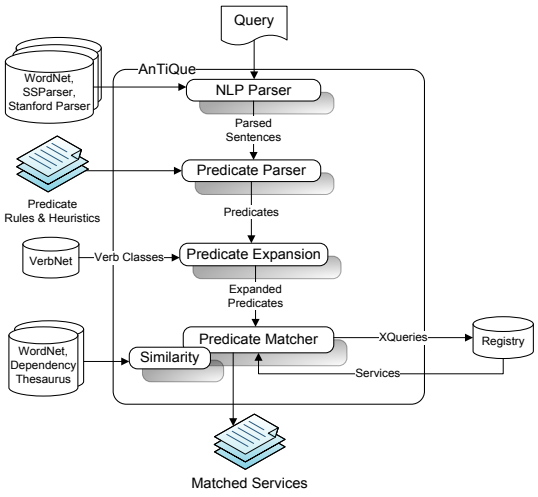


Figure 2. Internal structure of AnTiQue

Figure 2 depicts AnTiQue's 5 components. When invoked the service first divides query and case problem description text into sentences, then part-of-speech tagged, shallow parsed to identify sentence constituents and chunked in noun phrases. It then applies 21 syntax structure rules and 7 lexical extraction heuristics to identify predicates and extract lexical content in each sentence. Natural language sentences are presented as predicates in the form *PredicateValue(Object1, Object2)*. The service then expands each query predicate with additional predicate values that have similar meaning according to verb classes found in VerbNet to increase the likelihood of a match with predicates describing each case. For example the predicate value *abuse* is in the same verb class as *attack*. The service then matches all expanded predicates to a similar set of predicates that describe the problem description of each case in the repository. This is achieved using XQuery text- searching functions to discover an initial set of cases that satisfy global search constraints. Finally it applies semantic and dependency-based similarity measures to refine the candidate case study set. The service returns an ordered set of analogical cases based on the match score with the query.

The components use WordNet, VerbNet, and the Dependency Thesaurus to compute attributional and relational similarities. WordNet is a lexical database inspired by psycholinguistic theories of human lexical memory (Miller

1993). Its word senses and definitions provide the data with which to disambiguate terms in queries and case problem descriptions. Its semantic relations link terms to other terms with similar meanings with which to make service queries more complete. For example a service query with the term *car* is expanded with other terms with similar meaning, such as *automobile* and *vehicle*, to increase matches with web service descriptions.

VerbNet (Kipper et al. 2000) is a domain independent verb lexicon. It organizes terms into verb classes that refine Levin classes (Levin 1993) and add sub-classes to achieve syntactic and semantic coherence among members of a verb class. AnTiQue uses it to expand query predicate values with different members from the same verb class. For example, queries with the verb *abuse* are expanded with other verbs with similar meaning such as *attack*.

The Dependency Thesaurus supports dependency-based word similarity matching to detect similar words from text corpora. Lin (1998) used a 64-million word corpus to compute pair-wise similarities between all of the nouns, verbs, adjectives and adverbs in the corpus using a similarity measure. Given an input word the Dependency Thesaurus can retrieve similar words and group them automatically into clusters. AnTiQue used the Dependency Thesaurus to compute the relational similarity between 2 sets of predicates.

In the remainder of this section we demonstrate the AnTiQue components using text from the following example challenging behaviour situation:

> *A resident acts aggressively towards care staff and the resident verbally abuses other residents at breakfast. Suspect underlying insecurities to new people.*

Natural Language Processing

This component prepares the structured natural language (NL) service query for predicate parsing and expansion. In the first step the text is split into sentences. In the second a part-of-speech tagging process is applied that marks up the words in each sentence as corresponding to a particular lexical category (part-of-speech) using its definition and context. In the third step the algorithm applies a NL processing technique called *shallow parsing* that attempts to provide some machine understanding of the structure of a sentence without parsing it fully into a parsed tree form. The output is a division of the text's sentences into a series of words that, together, constitute a grammatical unit. In our example the tagged sentence *a resident acts aggressively towards care staff and the resident verbally abuses other residents at breakfast* is shown in Figure 3. Tags that follow a word with a forward slash (e.g. *driver/NN*) correspond to lexical categories including noun, verb, adjective and adverb. For example, the *NN* tag means "noun singular or mass", *DT* means "determinant" and *VBZ* means "verb, present tense, 3rd person singular". Tags attached to each chunk (e.g. *[The/DT driver/NN]NP*) correspond to phrasal categories. For instance, the *NP* tag denotes a "noun phrase", *VP* a "verb phrase", *S* a "simple declarative clause", *PP* a "prepositional phrase" and *ADVP* a "adverb phrase".

```
[A/DT resident/NN]NP [acts/VBZ]VP [aggressive-
ly/RB]ADVP [towards/]PP [care staff/NN]NP.
```

Figure 3. The sentence *a resident acts aggressively towards care staff* after performing part-of-speech tagging and chunking

The component then decomposes each sentence into its *phrasal categories* used in the next component to identify predicates in each sentence structure.

Predicate Parsing

This component automatically identifies predicate structures within each annotated NL sentence based on *syntax structure rules* and *lexical extraction heuristics*. Syntax structure rules break down a pre-processed NL sentence into sequences of phrasal categories where each sequence contains 2 or more phrasal categories. Lexical extraction heuristics are applied on each identified sequence of phrasal categories to extract its lexical content used to generate one or more predicates.

Firstly the algorithm applies 21 syntax structure rules. Each rule consists of a phrasal category sequence of the form *Ri* → *[Bj]*, meaning that the rule *Ri* consists of a phrasal category sequence *B1, B2,…, Bj*. For example the rule *R4* → *[NP, VP, S, VP, NP]* reads: rule *R1* consists of a *NP* followed by a *VP*, a *S*, a *VP*, and a *NP*, where *NP*, *VP* and *S* mean a *noun phrase*, a *verb phrase* and a *simple declarative clause* respectively. The method takes a phrasal category list as input and returns a list containing each discovered syntax structure rule and its starting point in the corresponding phrasal category list, e.g. *{(R1,3), (R5,1)}*. In our example, the input for the pre-processed sentence shown in Figure 3 corresponds to a list *Input = (NP, VP, ADVP, PP, NP)*. Starting from the first list position the method recursively checks whether there exists a sequence within the phrasal category list that matches one of the syntax structure rules. The output after applying the algorithm on list *Input* is a list of only one matched syntax structure rule, i.e. *Output = {(R2,1)}*.

Secondly the algorithm applies lexical extraction heuristics on a syntax structure rule-tagged sentence to extract content words for generating one or more predicates. For each identified syntax structure rule in a sentence the algorithm: (1) determines the position of both noun and verb phrases within the phrasal category sequence; (2) applies the heuristics to extract the content words (verbs and nouns) from each phrase category; (3) converts each verb and noun to its morphological root (e.g. *abusing* to *abuse*); and (4) generates the corresponding predicate p in the form *PredicateValue(Object1, Object2)* where *PredicateValue* is the verb and *Object1* and *Object2* the nouns. To illustrate this the algorithm identified rule *R2+* for our example sen-

tence in Figure 3. According to one heuristic *{R2+}* corresponds to the following phrasal category sequence *[NP, VP, ADVP, PP, NP]*. Therefore the algorithm determines the position of both noun and verb phrases within this sequence, i.e. noun phrases in *{NP,1}* and *{NP,5}* and verb phrases in *{VP,2}*. Lexical extraction heuristics are applied to extract the content words from each phrase category, i.e. *{NP,1}* → *resident*, *{NP,5}* → *care staff*, *{VP,2}* → *act*. Returning to our example, the algorithm generates two predicates for the sentence *a resident acts aggressively towards care staff and the resident verbally abuses other residents at breakfast*, namely *act(resident,care_staff)* and *abuse(resident,resident)*.

Predicate Expansion

Predicate expansion and matching are key to the service's effectiveness. In AnTiQue queries are expanded using words with similar meaning. AnTiQue uses ontological information from VerbNet to extract semantically related verbs for verbs in each predicate.

VerbNet classes are organised to ensure syntactic and semantic coherence among members, for example the verb *abuse* as *repeatedly treat a victim in a cruel way* is one of 24 members of the *judgement* class. Other members include *attack, assault* and *insult* and 20 other verbs as potential expansions. Thus VerbNet provides 23 verbs as potential expansions for the verb *abuse*. Although classes group together verbs with similar argument structures, the meanings of the verbs are not necessarily synonymous. For instance, the degree of attributional similarity between *abuse* and *reward* is very low, whereas the similarity between *abuse* and *assault* is very high. The service constrains use of expansion to verb members that achieve a threshold on the degree of attributional similarity computed with WordNet-based similarity measurements (Simpson and Dao 2005). Given 2 sets of text, $T1$ and $T2$, the measurement determines how similar the meaning of $T1$ and $T2$ is scored between 0 and 1. For example, for the verb *abuse*, the algorithm computes the degree of attributional similarity between *abuse* and each co-member within the *judgement* class. In our example verbs such as *attack*, *assault* and *insult* but not *honour* and *doubt* are used to generate additional predicates in the expanded query.

Predicate Matching

Coarse-grained Matching The expanded query is fired at problem descriptions of cases in the repository as an XQuery. Prior to executing the XQuery we pre-process all problem descriptions of cases in the registries using the Natural Language Processing and Predicate Parsing components and store them locally. The XQuery includes functions to match each original and expanded predicate value to equivalent representations of candidate problem descriptions of cases. The service retrieves an initial set of matched cases.

Fine-grained Matching The Predicate Matcher applies semantic and dependency-based similarity measures to assess the quality of the candidate case set. It computes relational similarity between the query and each case retrieved during coarse-grain matching. To compute relational similarities that indicate analogical matches between service and query predicate arguments the Predicate Matcher uses the Dependency Thesaurus to select web services that are relationally similar to mapped predicates in the service query.

In our example the case *Managing a disrespectful child*, which describes a good childcare practice to manage a disrespectful child, is one candidate case retrieved during coarse-grained matching. Figure 4 shows the problem and solution description of the case.

Name	Managing a disrespectful child
Problem	An intelligent 13-year-old boy voices opinions that are hurtful and embarrassing. The child refuses to consider the views of others and often makes discriminatory statements. The parents have removed his privileges and threatened to take him out of the school he loves. This approach has not worked. He now makes hurtful comments to his mother about her appearance. The child insults neighbours and guests at their home. He is rude and mimics their behaviour. The child shows no remorse for his actions. His mother is at the end of her tether.
Solution	The son needs very clear boundaries set. The parents are going to set clear rules on acceptable behaviour. They will state what they are not prepared to tolerate. They will highlight rude comments in a firm tone with the boy. He will receive an explanation as to why the comments are hurtful. Both parents will agree punishments for rule breaking that are realistic. They will work as a team and follow through on punishments. The son can then regain his privileges as rewards for consistent good behaviours.

Figure 4. A retrieved case describing a good childcare practice to manage a disrespectful child

The algorithm receives as inputs a pre-processed sentence list for the query and problem description of the case. It compares each predicate in the pre-processed query sentence list $Pred(j)_{Query}$ with each predicate in the pre-processed problem description sentence list $Pred(k)_{Case}$ to calculate the relevant match value, where

$$Pred(j)_{Query} = PredVal_{Query}(Arg1_{Query;}\ Arg2_{Query})$$

and

$$Pred(k)_{Case} = PredVal_{Case}\ (Arg1_{Case;}\ Arg2_{Case}).$$

The following conditions must be met in order to accept a match between the predicate pair:

1. $PredVal_{Case}$ exists in list of expanded predicate values of $PredVal_{Query}$;
2. $Arg1_{Query}$ and $Arg1_{Case}$ (or $Arg2_{Query}$ and $Arg2_{Case}$ respectively) are not the same;
3. $Arg1_{Case}$ (or $Arg2_{Case}$) exists in the Dependency Thesaurus result set when using $Arg1_{Query}$ (or $Arg2_{Query}$) as the query to the Thesaurus;
4. the resulting attributional similarity value from step 3 is below a specified threshold.

If all conditions are met, $Pred_{Case}$ is added to the list of matched predicates for the current case. If not the algorithm rejects $Pred_{Case}$ and considers the next list item.

AnTiQue queries the Dependency Thesaurus to retrieve a list of dependent terms. Terms are grouped automatically according to their dependency-based similarity degree. Firstly the algorithm checks whether the case predicate argument exists in this list. If so, it uses the semantic similarity component to further refine and assess the quality of the case predicate with regards to relational similarity.

Using this 2-step process AnTiQue returns an ordered set of analogical cases based on the match score with the query. In our example consider $Pred(j)_{Query}$ = *abuse(resident,residents)* extracted from the sentence *the resident verbally abuses other residents at breakfast,* and the $Pred(k)_{Case}$ = *insult(child,neighbours)* from the sentence *The child insults neighbours and guests at their home* taken from the description of the *Managing a disrespectful child* good childcare practice case in Figure 4. In this example all conditions for an analogical match are met: the predicate values *abuse* and *insult* are semantically equivalent whilst the object names *resident* and *child* and *residents* and *neighbours* are not the same. According to the Dependency thesaurus *child* is similar based on dependencies to *resident, and neighbour* is similar based on dependencies to *resident*. Finally the attributional similarity value of *resident* and *child* is 0.33, for *resident* and *neighbour* 0.25 – both below the specified threshold. As a result the predicate *insult(child,neighbours)* is added to the list of matched predicates for the predicate *abuse(resident,resident)*.

At the end of each invocation, the service returns an ordered set of the descriptions of the highest-scoring cases for the app component to display to the care staff.

The Creativity Trigger Generation Service

Although care staff can generate new resolutions directly from retrieved case descriptions, formative usability testing with the app revealed that users were often overwhelmed by the volume of text describing each case and uncertain how to start idea generation. Therefore we developed an automated service that care staff can invoke to generate creative triggers that extract content from the retrieved descriptions to conjecture new ideas that care staff can consider for the resident. Each trigger expresses a single idea that care staff can use to initiate creative thinking. The service uses the attributional predicates generated by the analogical matching discovery service to generate prompts that encourage analogical transfer of knowledge using the object-pair mappings identified in each predicate. It has the form *Think about a new idea based on the*, followed by mapped subject and object names in the target domain. To illustate, referring back to the *Managing a disrespectful child* good practice case retrieved from the childcare domain shown in Figure 1, Figure 5 shows how they are presented in the Carer mobile app while Figure 6 lists all creativity prompts that the service generates for the analogical case.

Figure 5. The Carer mobile app showing creativity prompts generated for the *Managing a disrespectful child* case

Think about a new idea based on the boundaries
Think about a new idea based on the clear rules
Think about a new idea based on the acceptable behaviour
Think about a new idea based on the rude comments
Think about a new idea based on the firm tone
Think about a new idea based on the explanation
Think about a new idea based on the comments
Think about a new idea based on the punishment
Think about a new idea based on the rule breaking
Think about a new idea based on the rewards
Think about a new idea based on the privileges
Think about new idea based on the consistent good behaviour

Figure 6. Creativity prompts generated for the *Managing a disrespectful child* case

Discovering Novel Ideas

Our design of the Carer app builds on Kerne et al. (2008)'s notion of human-centered creative cognition, in which information gathering and idea discovery occur concurrently, and information search and idea generation reinforce each other. The computational model of analogical reasoning searches for and retrieves information from analogical domains, and the creativity trigger generation service manipulates this information to support more effective idea generation from information, however the generation of new ideas remains a human cognitive activity undertaken by carers, supported by bespoke features implemented in the app.

For example, a carer can audio-record a new idea at any time in response to retrieved analogical cases and/or presented creativity triggers by pressing the red button visible in Figures 1 then verbalizing and naming the idea. Recorded ideas can be selected and ordered to construct a new care enhancement plan that can be extended with more

ideas and comments at any time. The carer can also play back the audio-recorded ideas and care enhancement plans to reflect and learn about them, inspired by similar use of the audio channel in digitally supported creative brainstorming (van Dijk et al. 2011). Reflection about an idea is supported with guidance from the app to reflect on why the idea is needed, what the idea achieved, and how and when the idea should be implemented. Reflection about a care enhancement plan is more sophisticated. A carer can drag-and-drop ideas in and out of the plan and into different sequences in it. Then, during play back of the plan, the app concatenates the individual idea audio files and plays the plan as a single recording, allowing the carer to listen to and reflect on each version of the plan as a different narrated story. Moreover, s/he can reflect collaboratively with colleagues using the app to share the plan as e-mail attachments, thereby enabling asynchronous communication between carers.

Formative Evaluation of the Carer App

The *Carer* app was made available for evaluation over prolonged periods with carers in a residential home. At the start of the evaluation, 7 nurses and care staff in the residential home were given an iPod Touch for their individual use during their care work over a continuous 28-day period. All 7 carers received face-to-face training in how to use the device and both apps before the evaluation started. A half-day workshop was held at the residential home to allow them to experiment with all of both apps' features. The carers were also given training and practice with the 3 forms of *Other Worlds* creativity technique through practice and facilitation to demonstrate how it can lead to idea generation. We deemed this training in the creativity technique an essential precondition for successful uptake of the app.

Even though it only lasted 4 weeks, the reported evaluation of the *Carer* app in one residential home provided valuable data about the use of mobile computing and creativity techniques in dementia care. Figure 7 depicts the results.

	Residential cases	Analogical domain cases	Ideas generated	Enhancement plans generated
Totals	27	5	14	10

Figure 7. Situations, ideas and care enhancement plans generated by care staff using Carer app

The focus group revealed that the nurses and carers implemented at least one major change to the care of one resident based on ideas generated using the app.

However, most of this success was not based on the analogical cases retrieved by the computational model. Whilst carers using the app did use the analogical matching service, and the service did retrieve relevant cases from analogical domains such as childcare and student management, the carers were unable to map and transfer knowledge from each of these source domains to the current dementia-related challenging behavior. The log data recorded only 5 uses of the *analogical reasoning* service to retrieve descriptions of cases of challenging behaviors from non-care domains. Rather, the carers appeared to use the *case-based reasoning* service to retrieve descriptions of challenging behavior cases from the care domain – the log data recorded 28 uses of this service, and most of the 114 recorded uses of the *creativity prompt generation* service were generated from these same-domain dementia cases. The focus group revealed that the carers did not use retrieved non-care domain cases because they were unable to recognize analogical similarities between them and the challenging behavior situation. We identified two possible reasons for this. Firstly, AnTiQue implements an approach that *approximates* analogical retrieval, hence there is always the possibility of computing seemingly "wrong" associations and retrieve cases that do not have analogical similarities. Previous evaluations of AnTiQue with regards to the precision and recall (Zachos & Maiden, 2008) revealed a recall score of 100% and a precision score of 66,6% highlighting one potential limitation of computing the attributional similarity using WordNet-based similarity measures.

Secondly, the results suggests that carers will require more interactive support based on results generated by the computational model to support cognitive analogical reasoning, consistent with previously reported empirical findings (e.g. Gick 1983). Examples of such increased interactive support include explicitly reporting each computed analogical mapping to the carer, use of graphical depictions of structured knowledge to transfer from the source to the target domain, and more deliberate analogical support prompts, for example based on the form *A is to B as C is to D*. We are extending Carer app with such features and look forward to reporting these extensions in the near future.

Related Work

Since the 1980s, the efforts of many Artificial Intelligence researchers and psychologists have contributed to an emerging agreement on many issues relating to analogical reasoning. In various ways and with differing emphases, all current computational analogical reasoning techniques use underlying structural information about the sources and the target domains to derive analogies. However, at the algorithmic level, they achieve the computation in many different ways (Keane et al. 1994).

Based on the Structure Mapping Theory (SMT), Gentner constructed a computer model of this theory called Structure Mapping Engine (SME) (Gentner 1989). The method assumes that both target and source situations are represented using a certain symbolic representation. The SME also only uses syntactic structures about the two situations as the main input knowledge — it has no knowledge of any kind of semantic similarity between various descriptions and relations in the two situations. All processing is based on syntactic structural features of the two given representations.

The application of analogical reasoning to software reuse is not new. For example, Massonet and van

Lamsweerde (1997) applied analogy-making techniques to complete partial requirements specifications using a rich, well-structured ontology combined with formal assertions. The method was based on query generalization for completing specifications. The absence of effective ontologies and taxonomies would expose the weaknesses of the proposed approach due to the reliance on ontologies. Pisan (2000) tried to overcome this weakness by applying the SME to expand semi-formal specifications. The idea was to find mappings from specifications for problems similar to the one in hand and use the mappings to adapt an existing specification without requiring domain specific knowledge. The research presented in this paper overcomes limitations of the above-mentioned approaches by using additional knowledge bases to extent the mapping process with semantic similarity measures.

Conclusion and Future Work

This paper reports a practical application of a computational model of analogical reasoning to a pressing social problem, which is to improve the care of older people with dementia. The result is a mobile app that is capable technically of accepting spoken and typed natural language input and retrieving analogical domain cases that can be presented with creativity triggers to support analogical problem solving.

The evaluation results reported revealed that our model of creative problem solving in dementia care did not describe all observed carer behavior, so we are currently repeating the rollout and evaluation of *Carer* in other residential homes to validate this finding. *Carer* is being extended with new creativity support features that include web images that match generated creativity prompts, and more explicit support for analogical reuse of cases from non-dementia care domains. We are extending the repository with new cases that are semantically closer to dementia care and, therefore, easier to recognize analogical similarities with.

Acknowledgment

The research reported in this paper is supported by the EU-funded MIRROR integrated project 257617, 2010-14.

References

Alzheimers Society, 2010. Statistics. http://www.alzheimers.org.uk/site/scripts/documents_info.php?documentID=341

Wimo A. & Prince M., 2010, 'World Alzheimer Report 2010: The Global Economic impact of Dementia, http://www.alz.co.uk/research/worldreport/

Brooker, D., 2007, 'Person-centred dementia care: Making Services Better, Bradford Dementia Group Good Practice Guides. Jessica Kingsley Publishers London and Philadephia

Osborn A., 1965. The Creative Trend in Education. In: Source Book For Creative Problem Solving: A Fifty Year Digest of Proven Innovation Processes. Creative Education Foundation Press, New York.

Le Storti A., J., Cullen P., A., Hanzlik E., M., Michiels, J., M., Piano L., A., Lawless Ryan P., Johnson W., 1999. Creative Thinking In Nursing Education: Preparing for Tomorrow's Challenges. Nursing Outlook. Vol. 47, no. 2 62–66.

Help The Aged, 2007, 'My Home Life: Quality of Life In Care Homes; A Review of the Literature London. [online] http://myhomelifemovement.org/ downloads/mhl_review.pdf [Accessed 5 Jan 2011].

Bromley J. and Emerson E., 1995, 'Beliefs and Emotional Reactions of Care Staff working with People with Challenging Behaviour', Journal of Intellectual Disability Research 39(4), 341-352

Innovation Company, 2002. 'Sticky Wisdom (?WhatIf!)'. Capstone Publishing Company Limited, Chichester.

Owen T. and Meyer J., 2009, 'Minimizing the use of 'restraint' in care homes: Challenges, dilemmas and positive approaches', Adult Services Report 25, Social Care Institute of Excellence, http://www.scie.org.uk/publications/reports/report25.pdf.

Houts, P.S., Nezubd, A.M., Magut Nezubd, C, Bucherc, J.A., 1996, 'The prepared family caregiver: a problem- solving approach to family caregiver education', Patient Education and Counselling, 27,1, 63-73.

Gentner D., 1983, 'Structure-Mapping: A Theoretical Framework for Analogy', Cognitive Science 5, 121-152.

Falkenhainer B., Forbus K.D. & Gentner D., 1989, 'The Structure-Mapping Engine: Algorithm and Examples', Artificial Intelligence 41, 1-63.

Kerne, A., Koh, E., Smith, S. M., Webb, A., Dworaczyk, B., 2008, 'combinFormation: Mixed-Initiative Composition of Image and Text Surrogates Promotes Information Discovery', ACM Transactions on Information Systems, 27(1), 1-45.

Kipper K., Dang H.T. and Palmer M., 2000, 'Class-based Construction of a Verb Lexicon', Proceedings AAAI/IAAI Conference 2000, 691–696.

Miller K., 1993, 'Introduction to WordNet: an On-line Lexical Database' Distributed with WordNet software.

Levin B., 1993, 'English Verb Classes and Alternations: A Preliminary Investigation', University Chicago Press.

Lin D., 1998, 'Automatic retrieval and clustering of similar words', In COLINGACL, 768-774.

Maiden N.A.M., 2012, 'D5.2 Deliverable: Techniques and Software Apps for Integrated Creative Problem Solving and Reflective Learning Version 1', Technical Report. Available at http://www.mirror-project.eu/showroom-a-publications/deliverables.

Simpson, T. and Dao, T. (2005). Wordnet-based semantic similarity measurement. codeproject.com/cs/library/semanticsimilaritywordnet.asp

van Dijk J., van der Roest J., van der Lugt R. & Overbeeke K., 'NOOT: A Tool for Sharing Moments of Reflection during Creative Meetings', Proceedings 10th ACM Creativity and Cognition Conference, Atlanta Georgia, Nov 2011, ACM Press.

Gick M.L., 1989, 'Two Functions of Diagrams in Problem Solving by Analogy', Knowledge Acquisition from Text and Pictures', ed. M. Handi & J.R. Levin, Elsevier Publishers B.V. North-Holland, 215-231.

Keane, M.T., Ledgeway, T. & Duff, S. (1994). Constraints on analogical map- ping: A comparison of three models. Cognitive Science, 18, 387–438.

Massonet, P. and van Lamsweerde, A. (1997). Analogical reuse of requirements framework. 3rd IEEE International Symposium on Requirements Engineering.

Pisan, Y. (2000). Extending requirement specifications using analogy. 22nd International Conference on Software Engineering (ICSE), limerick, Ireland.

Zachos K. & Maiden N.A.M., 2008, 'Inventing Requirements from Software: An Empirical Investigation with Web Services', Proceedings 16th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, 145-154.

Transforming Exploratory Creativity with DeLeNoX

Antonios Liapis[1], Héctor P. Martínez[2], Julian Togelius[1] and Georgios N. Yannakakis[2]
1: Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
2: Institute of Digital Games
University of Malta
Msida, Malta
anli@itu.dk, hector.p.martinez@um.edu.mt, juto@itu.dk, georgios.yannakakis@um.edu.mt

Abstract

We introduce DeLeNoX (Deep Learning Novelty Explorer), a system that autonomously creates artifacts in constrained spaces according to its own evolving interestingness criterion. DeLeNoX proceeds in alternating phases of *exploration* and *transformation*. In the exploration phases, a version of novelty search augmented with constraint handling searches for maximally diverse artifacts using a given distance function. In the transformation phases, a deep learning autoencoder learns to compress the variation between the found artifacts into a lower-dimensional space. The newly trained encoder is then used as the basis for a new distance function, transforming the criteria for the next exploration phase. In the current paper, we apply DeLeNoX to the creation of spaceships suitable for use in two-dimensional arcade-style computer games, a representative problem in procedural content generation in games. We also situate DeLeNoX in relation to the distinction between exploratory and transformational creativity, and in relation to Schmidhuber's theory of creativity through the drive for compression progress.

Introduction

Within computational creativity research, many systems have been designed that create artifacts automatically through search in a given space for predefined objectives, using evolutionary computation or some similar stochastic global search/optimization algorithm. Recently, the novelty search paradigm has aimed to abandon all objectives, and simply search the space for a set of artifacts that is as diverse as possible, i.e. for maximum novelty (Lehman and Stanley 2011). However, no search is without biases. Depending on the problem, the search space often contains constraints that limit and bias the exploration, while the mapping from genotype space (in which the algorithm searches) and phenotype space (in which novelty is calculated) is often indirect, introducing further biases. The result is a limited and biased novelty search, an incomplete exploration of the given space.

But what if we could characterize the bias of the search process as it unfolds and counter it? If the way space is being searched is continuously transformed in response to detected bias, the resulting algorithm would more thoroughly search the space by cycling through or subsuming biases. In applications such as game content generation, it would be particularly useful to sample the highly constrained space of useful artifacts as thoroughly as possible in this way.

In this paper, we present the Deep Learning Novelty Explorer (DeLeNoX) system, which is an attempt to do exactly this. DeLeNoX combines phases of exploration through constrained novelty search with phases of transformation through deep learning autoencoders. The target application domain is the generation of two-dimensional spaceships which can be used in space shooter games such as *Galaga* (Namco 1981). Automatically generating visually diverse spaceships which however fulfill constraints on believability addresses the "content creation" bottleneck of many game titles. The spaceships are generated by pattern-producing networks (CPPNs) via augmenting topologies (Stanley 2006). In the exploration phases, DeLeNoX finds the most diverse set of spaceships possible given a particular distance function. In the transformation phases, it characterizes the found artifacts by obtaining a low-dimensional representation of their differences. This is done via autoencoders, a novel technique for nonlinear principal component analysis (Bengio 2009). The features found by the autoencoder are orthogonal to the bias of the current CPPN complexity, ensuring that each exploratory phase has a different bias than the previous. These features are then used to derive a new distance function which drives the next exploration phase. By using constrained novelty search for features tailored to the concurrent complexity, DeLeNoX can create content that is both *useful* (as it lies within constraints) and *novel*.

We will discuss the technical details of DeLeNoX shortly, and show results indicating that a surprising variety of spaceships can be found given the highly constrained search space. But first we will discuss the system and the core idea in terms of exploratory and transformational creativity, and in the context of Schmidhuber's theory of creativity as an impulse to improve the compressibility of growing data.

Between exploratory and transformational creativity

A ubiquitous distinction in creativity theory is that between exploratory and transformational creativity. Perhaps the most well-known statement of this distinction is due to Boden (1990) and was later formalized by Wiggins (2006) and others. However, similar ideas seem to be present in al-

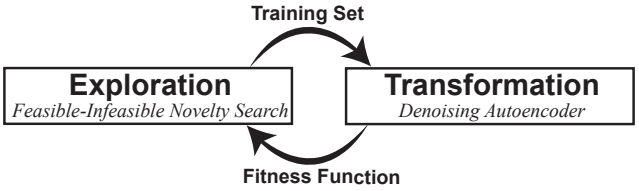


Figure 1: Exploration transformed with DeLeNoX: the flowchart includes the general principles of DeLeNoX (bold) and the methods of the presented case study (italics).

most every major discussion of creativity such as "thinking outside the box" (De Bono 1970), "paradigm shifts" (Kuhn 1962) etc. The idea requires that creativity is conceptualized as some sort of search in a space of artifacts or ideas. In Boden's formulation, exploratory creativity refers to search within a given search space, and transformational creativity refers to changing the rules that bind the search so that other spaces can be searched. Exploratory creativity is often associated with the kind of pedestrian problem solving that ordinary people engage in every day, whereas transformational creativity is associated with major breakthroughs that redefine the way we see problems.

Naturally, much effort has been devoted to thinking up ways of modeling and implementing transformational creativity in a computational framework. Exploratory creativity is often modeled "simply" as objective-driven search, e.g. using constraint satisfaction techniques or evolutionary algorithms (including interactive evolution).

We see the distinction between exploratory and transformative creativity as a matter quantitative rather than qualitative. In some cases, exploratory creativity is indeed limited by hard *constraints* that must be broken in order to transcend into unexplored regions of search space (and thus achieve transformational creativity). In other cases, exploratory creativity is instead limited by *biases* in the search process. A painter might have a particular painting technique she defaults to, a writer a common set of plot devices he returns to, and an inventor might be accustomed to analyze problems in a particular order. This means that some artifacts are in practice never found, even though finding them would not break any constraints — those artifacts are contained within the space delineated by the original constraints. Analogously, any search algorithm will over-explore some regions of search space and in practice never explore other areas because of particularities related to e.g. evaluation functions, variation operators or representation (cf. the discussion of search biases in machine learning (Mitchell 1997)). This means that some artifacts are never found in practice, even though the representation is capable of expressing them and there exists a way in which they could in principle be found.

DeLeNoX and Transformed Exploration

As mentioned above, the case study of this paper is two-dimensional spaceships. These are represented as images generated by Compositional Pattern-Producing Networks (CPPNs) with constraints on which shapes are viable spaceships. Exploration is done through a version of novelty search, which is a type of evolutionary algorithm that seeks to explore a search space as thoroughly as possible rather than maximizing an objective function. In order to do this, it needs a measure of difference between individuals. The distance measure inherently privileges some region of the search space over others, in particular when searching at the border of feasible search space. Additionally, CPPNs with different topologies are likely to create specific patterns in generated spaceships, with more complex CPPNs typically creating more complex patterns. Therefore, in different stages of this evolutionary complexification process, different regions of the search space will be under-explored. Many artifacts that are expressible within the representation will thus most likely not be found; in other words, there are limitations to creativity because of search biases.

In order to alleviate this problem and achieve a fuller coverage of space, we algorithmically *characterize* the biases from the search process and the representation. This is what the autoencoders do. These autoencoders are applied on a set of spaceships resulting from an initial exploration of the space. A trained autoencoder is a function from a complete spaceship (phenotype) to a relatively low-dimensional array of real values. We then use the output of this function to compute a new distance measure, which differs from previous ones in that it better captures typical patterns at the current representational power of the spaceship-generating CPPNs. Changing the distance function amounts to changing the exploration process of novelty search, as novelty search is now in effect searching along different dimensions (see Fig. 1). We have thus *transformed* exploratory creativity, not by changing or abandoning any constraints, but by adjusting the search bias. This can be seen as analogous to changing the painting technique of a painter, the analysis sequence of an inventor, or introducing new plot devices for a writer. All of the spaceships that are found by the new search process could in principle have been found by the previous processes, but were very unlikely to be.

Schmidhuber's theory of creativity

Schmidhuber (2006; 2007) advances an ambitious and influential theory of beauty, interestingness and creativity that arguably holds explanatory power at least under certain circumstances. Though the theory is couched in computational terms, it is meant to be applicable to humans and other animals as well as artificial agents. In Schmidhuber's theory, a beautiful pattern for a curious agent A is one that can successfully be compressed to much smaller description length by that agent's compression algorithm. However, perfect beauty is not interesting; an agent gets bored by environments it can compress very well and cannot learn to compress better, and also by those it cannot compress at all. Interesting environments for A are those which A can compress to some extent but where there is potential to improve the compression ratio, or in other words potential for A to learn about this type of environment. This can be illustrated by tastes in reading: beginning readers like to read linguistically and thematically simple texts, but such texts are seen by advanced readers as "predictable" (i.e. compressible), and the curious advanced readers therefore seek out more

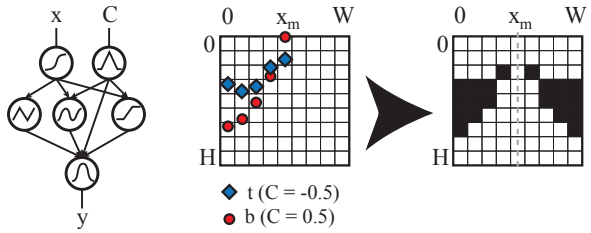
complex texts. In Schmidhuber's framework, creative individuals such as artists and scientists are also seen as a curious agents: they seek to pose themselves problems that are on the verge of what they can solve, learning as much as possible in the process. It is interesting to note the close links between this idea and the theory of flow (Csikszentmihalyi 1996) but also theories of learning in children (Vygotsky et al. 1987) and game-players (Koster and Wright 2004).

The DeLeNoX system fits very well into Schmidhuber's framework and can be seen as a novel implementation of a creative agent. The system proceeds in phases of exploration, carried out by novelty search which searches for interesting spaceships, and transformation, where autoencoders learn to compress the spaceships found in the previous exploration phase (see Fig. 1) into a lower-dimensional representation. In the exploration phases, "interesting" amounts to far away from existing solutions according to the distance function defined by the autoencoder in the previous transformation phase. This corresponds to Schmidhuber's definition of interesting environments as those where the agent can learn (improve its compression for the new environment); the more distant the spaceships are, the more they force the autoencoder to change its compression algorithm (the weights of the network) in the next transformation phase. In the transformation phase, the learning in the autoencoder directly implements the improvement in capacity to compress recent environments ("compression progress") envisioned in Schmidhuber's theory.

There are two differences between our model and Schmidhuber's model of creativity, however. In Schmidhuber's model, the agent stores all observations indefinitely and always retrains its compressor on the whole history of previous observations. As DeLeNoX resets its archive of created artifacts in every exploration phase, it is a rather forgetful creator. A memory could be implemented by keeping an archive of artifacts found by novelty search in all previous exploration phases, but this would incur a high and constantly increasing computational cost. It could however be argued that the dependence of each phase on the previous represents an implicit, decaying memory. The other difference to Schmidhuber's mechanism is that novelty search always looks for the solution/artifact that is most different to those that have been found so far, rather than the one predicted to improve learning the most. Assuming that the autoencoder compresses *relatively* better the more diverse the set of artifacts is, this difference vanishes; this assumption is likely to be true at least in the current application domain.

A case study of DeLeNoX: Spaceship Generation

This paper presents a case study of DeLeNoX for the creation of spaceship sprites, where exploration is performed via constrained novelty search which ensures a believable appearance, while transformation is performed via a denoising autoencoder which finds typical features in the spaceships' current representation (see Fig. 1). Search is performed via neuroevolution of augmenting topologies, which changes the representational power of the genotype and warrants the transformation of features which bias the search.



(a) CPPN (b) Spaceship representation

Figure 2: Fig 2a shows a sample CPPN using the full range of pattern-producing activation functions available. Fig. 2b shows the process of spaceship generation: the coordinates 0 to x_m, normalized as 0 to 1 (respectively) are used as input x of the CPPN. Two C values are used for each x, resulting in two points, top (t) and bottom (b) for each x. CPPN input x and output y are treated as the coordinates of t and b; if t has a higher y value than that of b then the column is empty, else the hull extends between t and b. The generated hull is reflected vertically along x_m.

Domain Representation

Spaceships are stored as two-dimensional sprites; the spaceship's hull is shown as black pixels. Each spaceship is encoded by a Compositional Pattern-Producing Network (CPPN), which is able to create complex patterns via function composition (Stanley 2006). A CPPN is ideal for visual representation as it can be queried with arbitrary spatial granularity (infinite resolution); however, this study uses a fixed resolution for simplicity. Unlike standard artificial neural networks where all nodes have the same activation function, each CPPN node may have a different, pattern-producing function; six activation functions bound within $[0, 1]$ are used in this study (see Fig. 2a). To generate a spaceship, the sprite is divided into a number of equidistant columns equal to the sprite's width (W) in pixels. In each column, two points are identified as top (t) and bottom (b); the spaceship extends from t to b, while no hull exists if t is below b (see Fig. 2b). The y coordinate of the top and bottom points is the output of the CPPN; its inputs are the point's x coordinate and a constant C which differentiates between t and b (with $C = -0.5$ and $C = 0.5$, respectively). Only half of the sprites' columns, including the middle column at $x_m = \lceil \frac{W}{2} \rceil$, are used to generate t and b; the remaining columns are derived by reflecting vertically along x_m.

A sufficiently expanded CPPN, as a superset of a multilayer perceptron, is theoretically capable of representing any function. This means that any image could in principle be produced by a CPPN. However, the interpretation of CPPN output we use here means that images are severely limited to those where each column contains at most one vertical black bar. Additionally, the particularities of the NEAT complexification process, of the activation functions used and of the distance function which drives evolution make the system heavily biased towards particular shapes. It is this latter bias that is characterized within the transformation phase.

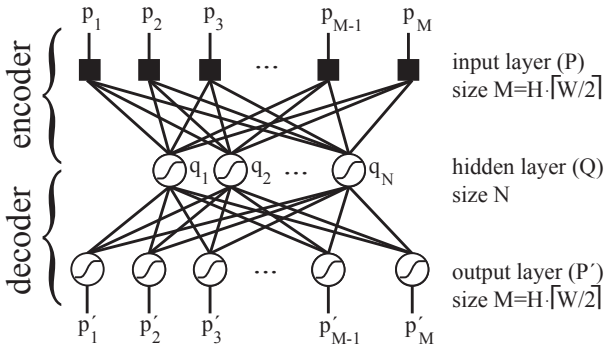


Figure 3: The autoencoder architecture used for DeLeNoX, consisting of the encoder where $Q = f^{\mathbf{w}}(P)$ and the decoder where $P' = g^{\mathbf{w}}(Q)$. The higher-level representation in $q_1, q_2, \ldots, q_N$ is used to calculate the difference between individuals for the purposes of novelty search.

Transformation Phase: Denoising Autoencoder

The core innovation of DeLeNoX is the integration of autoencoders (AEs) in the calculation of the novelty heuristic (described in the next section), which is used to explore the search space according to the current representational power of the encoding CPPNs. AEs (Hinton and Zemel) are non-linear models that transform an input space P into a new distributed representation Q by applying a deterministic parametrized function called the *encoder* $Q = f^{\mathbf{w}}(P)$. This encoder, instantiated in this paper as a single layer of logistic neurons, is trained alongside a *decoder* (see Fig. 3) that maps back the transformed into the original representation ($P' = g^{\mathbf{w}}(Q)$) with a small reconstruction error, i.e. the original and corresponding decoded inputs are similar. By using a lower number of neurons than inputs, the AE is a method for the lossy compression of data; its most desirable feature, for the purposes of DeLeNoX, is that the compression is achieved by exploiting typical patterns observed in the training set. In order to increase the robustness of this compression, we employ *denoising autoencoders* (DAs), an AE variant that corrupts the inputs of the encoder during training while enforcing that the original uncorrupted data is reconstructed (Vincent et al. 2008). Forced to both maintain most of the information from the input and undo the effect of corruption, the DA must "capture the main variations in the data, i.e. on the manifold" (Vincent et al. 2008), which makes DAs far more powerful tools than linear models for principal component analysis.

For the purposes of detecting the core visual features of the generated spaceships, DeLeNoX uses DAs to transform the spaceship's sprite to a low-dimensional array of real values, which correspond to the output of the encoder. Since spaceships are symmetrical along x_m, the training set consists of the left half of every spaceship sprite (see Fig. 4d). The encoder has $H \cdot \lceil \frac{W}{2} \rceil$ inputs (P), which are assigned a corrupted version of the spaceship's half-sprite; corruption is accomplished by randomly replacing pixels with 0, which is the same as randomly removing pixels from the spaceship (see Fig. 4e). The encoder has N neurons, corresponding to

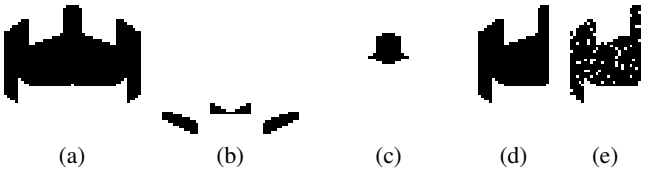


Figure 4: Sample spaceships of 49 by 49 pixels, used for demonstrating DeLeNoX. Fig. 4a is a feasible spaceship; Fig. 4b and 4c are infeasible, as they have disconnected pixels and insufficient size respectively. The autoencoder is trained to predict the left half of the spaceship in Fig. 4a (Fig. 4d) from a corrupted version of it (Fig. 4e).

the number of high-level features captured; each feature q_i is a function of the input P as $sig(W_i \cdot P + b_i)$ where $sig(x)$ the sigmoid function and $\{W_i, b_i\}$ the feature's learnable parameters (weight set and bias value, respectively). The output P' of the decoder is an estimation of the uncorrupted half-sprite derived from $Q = [q_1, q_2, \ldots, q_N]$ via $P' = sig(W' \cdot Q + B')$; in this paper the DA uses tied weights and thus W' is the transpose of $W = [W_1, W_2, \ldots, W_N]$. The parameters $\{W, B, B'\}$ are trained via backpropagation (Rumelhart 1995) according to the mean squared error between pixels in the uncorrupted half-sprite with those in the reconstructed sprite.

Exploration Phase: Constrained Novelty Search

The spaceships generated by DeLeNoX are expected to be useful for a computer game; spaceships must have a believable appearance and sufficient size to be visible. Specifically, spaceships must not have disconnected pixels and must occupy at least half of the sprite's height and width; see examples of infeasible spaceships in Fig. 4b and 4c. In order to optimize feasible spaceships towards novelty, content is evolved via a *feasible-infeasible novelty search* (FINS) (Liapis, Yannakakis, and Togelius 2013). FINS follows the paradigm of the feasible-infeasible two-population genetic algorithm (Kimbrough et al. 2008) by maintaining two separate populations: a *feasible population* of individuals satisfying all constraints and an *infeasible population* of individuals failing one or more constraints. Each population selects individuals among its own members, but feasible offspring of infeasible parents are transferred to the feasible population and vice versa; this form of interbreeding increases the diversity of both populations. In FINS, the feasible population selects parents based on a novelty heuristic (ρ) while the infeasible population selects parents based on their proximity to the feasible border (f_{inf}), defined as:

$$f_{inf} = 1 - \frac{1}{3}\left[\max\{0, 1 - \tfrac{2w}{W}\} + \max\{0, 1 - \tfrac{2h}{H}\} + \tfrac{A_s}{A}\right]$$

where w and h is the width and height of the spaceship in pixels; W and H is the width and height of the sprite in pixels; A is the total number of black pixels on the image and A_s the number of pixels on all disconnected segments.

For the feasible population, the paradigm of novelty search is followed in order to explore the full spectrum of

the CPPNs' representational power. The fitness score $\rho(i)$ for a feasible individual i amounts to its average difference with the k closest feasible neighbors within the population or in an archive of past novel individuals (Lehman and Stanley 2011). In each generation, the l highest-scoring feasible individuals are inserted in an archive of novel individuals. In DeLeNoX, the difference used to calculate ρ is the Euclidean distance between the high-level features discovered by the denoising autoencoder; thus $\rho(i)$ is calculated as:

$$\rho(i) = \frac{1}{k} \sum_{m=1}^{k} \sqrt{\sum_{n=1}^{N} [q_n(i) - q_n(\mu_m)]^2}$$

where μ_m is the m-th-nearest neighbor of i (in the population or the archive of novel individuals); N is the number of hidden nodes (features) of the autoencoder and $q_n(i)$ the value of feature n for spaceship i. As with the training process of the denoising autoencoder, the left half of spaceship i is used as input to $q_n(i)$, although the input is not corrupted.

In both populations, evolution is carried out via neuroevolution of augmenting topologies (Stanley and Miikkulainen 2002) using only mutation; an individual in the population may be selected (via fitness-proportionate roulette wheel selection) more than once for mutation. Mutation may add a hidden node (5% chance), add a link (10% chance), change the activation function of an existing node (5% chance) or modify all links' weights by a small value.

Experimentation

DeLeNoX will be demonstrated with the iteratively transformed exploration of spaceships on sprites of 49 by 49 pixels. The experiment consists of a series of *iterations*, with each iteration divided into an exploration phase and a transformation phase. The exploration phase uses constrained novelty search to optimize a set of diverse spaceships, with "diversity" evaluated according to the features of the previous iteration; the transformation phase uses the set of spaceships optimized in the exploration phase to create new features which are better able to exploit the regularities of the current spaceship complexity. Each exploration phase creates a set of 1000 spaceships, which are generated from 100 independent runs of the FINS algorithm for 50 generations; the 10 fittest feasible individuals of each run are inserted into the set. Given the genetic operators used in the mutation scheme, each exploration phase augments the CPPN topology by roughly 5 nodes. While the first iteration starts with an initial population consisting of CPPNs with no hidden nodes, subsequent iterations start with an initial population of CPPNs of the same complexity as the final individuals of the previous iteration. The total population of each run is 200 individuals, and parameters of novelty search are $k = 20$ and $l = 5$. Each evolutionary run maintains its own archive of novel individuals; no information regarding novelty is shared from previous iterations or across runs. Forgetting past visited areas of the search space is likely to hinder novelty search, but using a large archive of past individuals comes with a huge computational burden; given that CPPN topology augments in each iteration, it is less likely that previous novel individuals will be re-discovered, which makes "forgetting" past breakthroughs an acceptable sacrifice.

Each transformation phase trains a denoising autoencoder with a hidden layer of 64 nodes, thus creating 64 high-level features. The weights and biases for these features are trained in the 1000 spaceships created in the exploration phase. Training runs for 1000 epochs, trying to accurately predict the real half-sprite of the spaceship (see Fig. 4d) from a corrupted version of it (see Fig. 4e); corruption occurs by replacing any pixel with a white pixel (with 10% chance).

We observe the progress of DeLeNoX for 6 iterations. For the first iteration, the features driving the exploration phase are trained on a set of 1000 spaceships created by randomly initialized CPPNs with no hidden nodes; these spaceships and features are identified as "initial". The impact of transformation is shown via a second experiment, where spaceships evolve for 6 iterations using the initial set of features trained from simple spaceships with no transformation phases between iterations; this second experiment is named "static" (contrary to the proposed "transforming" method).

The final spaceships generated in the exploration phase of each iteration are shown in Fig. 5 for the transforming run and in Fig. 6 for the static run. For the purposes of brevity, the figures show six samples selected based on their diversity (according to the features on which they were evolved); Fig. 5 and 6 therefore not only showcase the artifacts generated by DeLeNoX, but the sampling method demonstrates the shapes which are identified as "different" by the features.

In Fig. 5, the shifting representational power of CPPNs is obvious: CPPNs with no hidden nodes tend to create predominantly V-shaped spaceships, while larger networks create more curved shapes (such as in the 2nd iteration) and eventually lead to jagged edges or "spikes" in later iterations. While CPPNs can create more elaborate shapes with larger topologies, Fig. 5 includes simple shapes even in late iterations: such an example is the 6th iteration, where two of the sample spaceships seem simple. This is likely due to the lack of a "long-term memory", since there is no persistent archive of novel individuals across iterations.

In terms of detected features, Fig. 8 displays a random sample of the 64 features trained in each transformation phase of the transforming run; the static run uses the "initial" features (see Fig. 8a) in every iteration. The shape of the spaceships directly affects the features' appearance: for instance, the simple V-shaped spaceships of the initial training set result in features which detect diagonal edges. The features become increasingly more complex, and thus difficult to identify, in later iterations: while in the 1st iteration straight edges are still prevalent, features in the 5th or 6th iterations detect circular or vertical areas.

Comparing Fig. 6 with Fig. 5, we observe that despite the larger CPPN topologies of later iterations, spaceships evolved in the static run are much simpler than their respective ones in the transforming run. Exploration in the static run is always driven by simple initial features (see Fig. 8a), showing how the features used in the fitness function ρ bias search. On the contrary, the transformation phase in each iteration counters this bias and re-aligns exploration towards more visually diverse artifacts.

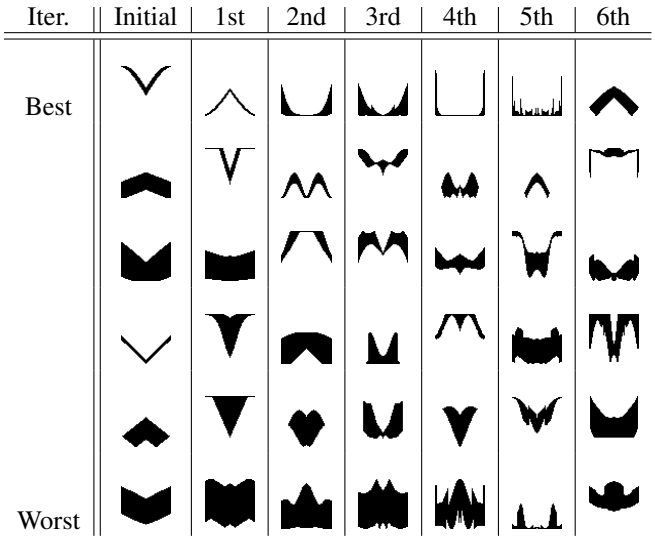


Figure 5: Sample spaceships among the results of each iteration of exploration; such spaceships comprise the training set for detecting the next iteration's features (transforming run). The best and worst spaceship in terms of difference (using the previous iteration's features) is included, along with spaceships evenly distributed in terms of difference.

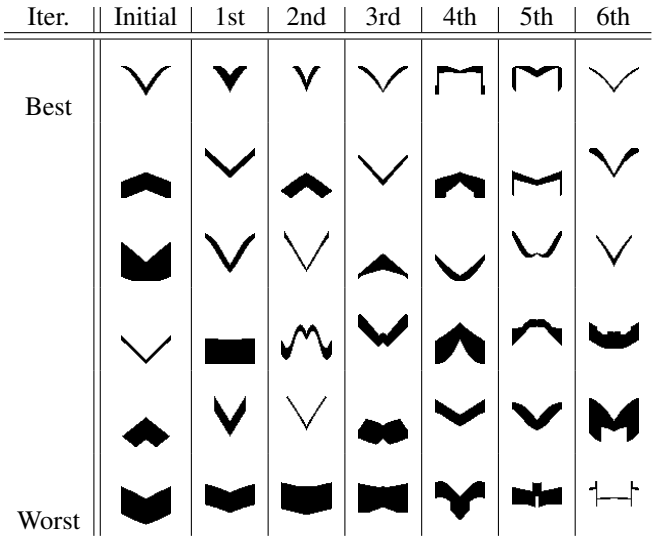


Figure 6: Sample spaceships (sorted by difference) among the results of each iteration of exploration driven by static features trained on the initial spaceship set (static run).

The diversity of spaceships and the quality of detected features can be gleaned from Fig. 7, in which features trained in different iterations of the transforming run generate distance metrics which evaluate the diversity of every iteration's training set, both for the transforming and for the static run. Diversity is measured as the Euclidean distance averaged from all spaceship pairs of the training set of an iteration. In the transforming run, the highest diversity score for a feature set is usually attained in the training set of the following iteration (e.g. the initial features score the highest diversity in the 1st iteration's spaceships). This is expected, since the features of the previous iteration are used in the distance function driving novelty search in the next iteration. This trend, however, does not hold in the last 3 iterations, possibly because patterns after the 3rd iteration become too complex for 64 features to capture, while the simpler patterns of earlier iterations are more in tune with what they can detect. It is surprising that features of later iterations, primarily those of the 3rd and 6th iteration, result in high diversity values in most training sets, even those of the static run which were driven by the much simpler initial features. It appears that features trained in the more complicated shapes of later iterations are more general — as they can detect patterns they haven't actually seen, such as those in the static run — than features of the initial or 1st iteration which primarily detect straight edges (see Fig. 8).

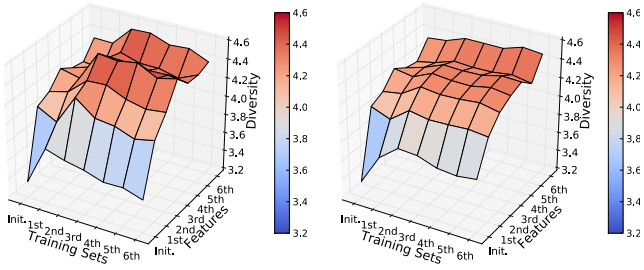
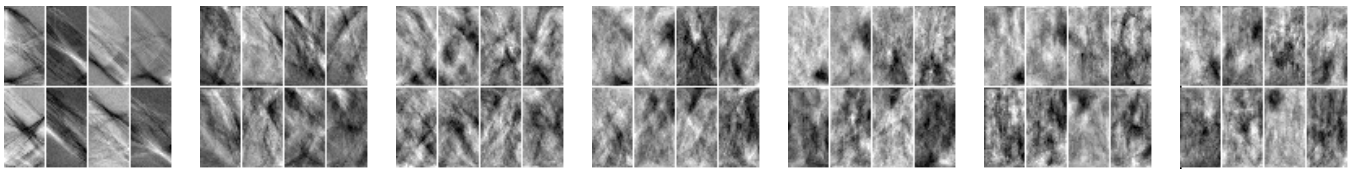


Figure 7: Diversity scores of the training sets at the end of each iteration's exploration phase, derived from the feature sets trained in the transformation phases of the transforming run. The training sets of the transforming run are evaluated on the left figure, and those of the static run on the right.

Discussion

This paper has presented DeLeNoX as a system which transforms exploration of the search space in order to counter the biases of the representation and the evolutionary process. While short, the included case study demonstrates the potential of DeLeNoX in several distinct but complementary ways. The shifting representation of augmenting CPPNs benefits from the iterative transformations of the novelty heuristic which is used to evolve it, as demonstrated by early features which detect straight lines versus later features which focus on areas of interest. Using the early, simple features for evolving complex CPPNs is shown to hinder exploration since the representational bias which caused those features to be prevalent has been countered by augmenting topologies. On the other hand, the iterative exploration guided by features tailored to the representation creates a more diverse training set for the autoencoder, resulting in an overall improvement in the features detected as shown by the increased diversity scores of later features on the same data. This positive feedback loop, where the exploration phase benefits from the transformation phase, which in turn benefits from the improved divergent search of exploration is the core argument for DeLeNoX. It should be noted,



(a) Initial (b) 1st (c) 2nd (d) 3rd (e) 4th (f) 5th (g) 6th

Figure 8: A sample of the 64 trained features at the end of each iteration. The visualization displays the weights of each pixel of the input (i.e. the left half of the spaceship's sprite). Weights are normalized to black (lowest) and white (highest).

however, that for this case study DeLeNoX is not without its own biases, as the increasingly diverse training set eventually challenges the feature detector's ability to capture typical patterns in the latest of presented iterations; suggestions for countering such biases will be presented in this section.

The case study presented in this paper is an example of exploration via high-level features derived by compressing information based on their statistical dependencies. The number of features chosen was arguably arbitrary; it allows for a decent compression (980 pixels to 64 real values) and measuring the Euclidean distance for novelty search is computationally manageable. At the same time, it is large enough to capture the most prevalent features among generated spaceships, at least in the first iterations where spaceships and their encoding CPPNs are simple. As exploration becomes more thorough — enhanced both by the increased representational power of larger CPPNs and by more informed feature detectors — typical patterns become harder to find. It could be argued that as exploration results in increasingly more diverse content, the number of features should increase to counter the fewer dependencies in the training set; for the same reasons, the size of the training set should perhaps increase. Future experiments should evaluate the impact of the number of features and the size of the training set both on the accuracy of the autoencoder and on the progress of novelty search. Other experiments should explore the potential of adjusting these values dynamically on a per-iteration basis; adjustments can be made via a constant multiplier or according to the quality of generated artifacts.

It should be pointed out that the presented case study uses a single autoencoder, which is able to discover simple features such as edges. These simple features are easy to present visually, and deriving the distance metric is straightforward based on the outputs of the autoencoder's hidden layer. For a simple testbed such as spaceship generation, features discovered by the single autoencoder suffice — especially in early iterations of novelty search. However, the true potential of DeLeNoX will be shown via stacked autoencoders which allow for truly *deep* learning; the outputs from the upper layers of such a deep belief network (Bengio 2009) represent more "abstract" concepts than those of a single autoencoder. Using such robust features for deriving a novelty value is likely to address current limitations of the feature extractor in images generated by complex CPPNs, and can be applied to more complex problems.

The case study presented in this paper is ideal for demonstrating DeLeNoX due to the evolutionary complexification of CPPNs; the indirect mapping between genotype and phenotype and the augmenting topologies both warrant the iterative transformation of the features which drive novelty search. A direct or static mapping would likely find the iterative transformation of the search process less useful, since representational bias remains constant. However, any indirect mapping between genotype and phenotype including neuroevolution, grammatical evolution or genetic programming can be used for DeLeNoX.

Related Work

DeLeNoX is indirectly linked to the foci of a few studies in automatic content generation and evolutionary art. The creation of artifacts has been the primary focus of evolutionary art; however, the autonomy of art generation is often challenged by the use of interactive evolution driven by human preferences. In order to create closed systems, an art appreciation component is used to automatically evaluate generated artifacts. This *artificial art critic* (Machado et al. 2003) is often an artificial neural network pre-trained to simulate user ratings in a collection of generated content (Baluja, Pomerleau, and Jochem 1999) or between man-made and generated images (Machado et al. 2007). Image compression has also been used in the evaluation of generated artifacts (Machado et al. 2007). While DeLeNoX essentially uses an artificial neural network to learn features of the training set, it does not simulate human aesthetic criteria as its training is unsupervised; moreover, the learned features are used to diversify the generated artifacts rather than converge them towards a specific art style or aesthetic. This same independence from human aesthetics, however, makes evaluating results of DeLeNoX difficult. Finally, while the autoencoder compresses images to a much smaller size, this compression is tailored to the particularities of the training set, unlike the generic compression methods such as *jpeg* used in NEvAr (Machado et al. 2007). Recent interest in dynamically extracting features targeting deviation from previously evolved content (Correia et al. 2013) has several similarities to DeLeNoX; the former approach, however, does not use novelty search (and thus exploration of the search space is limited) while features are extracted via supervised learning on a classification task between newly (and previously) generated artifacts and man-made art pieces.

The potential of DeLeNoX is demonstrated using the generation of spaceship sprites as a testbed. Spaceship gener-

ation is representative of the larger problem of automatic game content creation which has recently received considerable academic interest (Yannakakis 2012). Search-based techniques such as genetic algorithms are popular for optimizing many different properties of game content; for a full survey see (Togelius et al. 2011). Procedurally generated spaceships have been optimized, via neuroevolution, for performance measures such as speed (Liapis, Yannakakis, and Togelius 2011a) or for predefined aesthetic measures such as symmetry (Liapis, Yannakakis, and Togelius 2012; 2011b). Similarly to the method described in this paper, these early attempts use CPPN-NEAT to generate a spaceship's hull. This paper, however, describes a spaceship via top and bottom points and uses a sprite-based representation, both of which are more likely to generate feasible content; additionally, the spaceship's thrusters and weapons are not considered.

Acknowledgments

The research is supported, in part, by the FP7 ICT project SIREN (project no: 258453) and by the FP7 ICT project C2Learn (project no: 318480).

References

Baluja, S.; Pomerleau, D.; and Jochem, T. 1999. Towards automated artificial evolution for computer-generated images. *Musical networks* 341–370.

Bengio, Y. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2(1):1–127.

Boden, M. 1990. *The Creative Mind*. Abacus.

Correia, J.; Machado, P.; Romero, J.; and Carballal, A. 2013. Feature selection and novelty in computational aesthetics. In *Proceedings of the International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design*, 133–144.

Csikszentmihalyi, M. 1996. *Creativity-flow and the psychology of discovery and invention*. Harper perennial.

De Bono, E. 1970. *Lateral thinking: creativity step by step*. Harper & Row.

Hinton, G. E., and Zemel, R. S. Autoencoders, minimum description length, and helmholtz free energy. In *Advances in Neural Information Processing Systems*.

Kimbrough, S. O.; Koehler, G. J.; Lu, M.; and Wood, D. H. 2008. On a feasible-infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. *European Journal of Operational Research* 190(2):310–327.

Koster, R., and Wright, W. 2004. *A Theory of Fun for Game Design*. Paraglyph Press.

Kuhn, T. S. 1962. *The Structure of Scientific Revolutions*. University of Chicago Press.

Lehman, J., and Stanley, K. O. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation* 19(2):189–223.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2011a. Neuroevolutionary constrained optimization for content creation. In *Proceedings of IEEE Conference on Computational Intelligence and Games*, 71–78.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2011b. Optimizing visual properties of game content through neuroevolution. In *Artificial Intelligence for Interactive Digital Entertainment Conference*.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2012. Adapting models of visual aesthetics for personalized content creation. *IEEE Transactions on Computational Intelligence and AI in Games* 4(3):213–228.

Liapis, A.; Yannakakis, G.; and Togelius, J. 2013. Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proceedings of Genetic and Evolutionary Computation Conference*.

Machado, P.; Romero, J.; Manaris, B.; Santos, A.; and Cardoso, A. 2003. Power to the critics — A framework for the development of artificial art critics. In *IJCAI 2003 Workshop on Creative Systems*.

Machado, P.; Romero, J.; Santos, A.; Cardoso, A.; and Pazos, A. 2007. On the development of evolutionary artificial artists. *Computers & Graphics* 31(6):818 – 826.

Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill.

Rumelhart, D. 1995. *Backpropagation: theory, architectures, and applications*. Lawrence Erlbaum.

Schmidhuber, J. 2006. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science* 18(2):173–187.

Schmidhuber, J. 2007. Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity & creativity. *Lecture Notes in Computer Science* 4755:26–38.

Stanley, K. O., and Miikkulainen, R. 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10(2):99–127.

Stanley, K. O. 2006. Exploiting regularity without development. In *Proceedings of the AAAI Fall Symposium on Developmental Systems*. Menlo Park, CA: AAAI Press.

Togelius, J.; Yannakakis, G.; Stanley, K.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* (99).

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine learning*, 1096–1103.

Vygotsky, L.; Rieber, R.; Carton, A.; Wollock, J.; and Glick, J. 1987. *The Collected Works of L.S. Vygotsky: Scientific legacy*. Cognition and Language. Plenum Press.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Yannakakis, G. N. 2012. Game AI revisited. In *Proceedings of ACM Computing Frontiers Conference*.

A Discussion on Serendipity in Creative Systems

Alison Pease, Simon Colton, Ramin Ramezani, John Charnley and Kate Reed
Computational Creativity Group
Department of Computing
Imperial College, London
ccg.doc.ic.ac.uk

Abstract

We investigate serendipity, or happy, accidental discoveries, in CC, and propose computational concepts related to serendipity. These include a *focus-shift*, a breakdown of serendipitous discovery into *prepared mind*, *serendipity trigger*, *bridge* and *result* and three dimensions of serendipity: *chance*, *sagacity* and *value*. We propose a definition and standards for computational serendipity and evaluate three creative systems with respect to our standards. We argue that this is an important notion in creativity and, if carefully developed and used with caution, could result in a valuable new discovery technique in CC.

Introduction and motivation

A serendipitous discovery is one in which chance plays a crucial role and which results in a surprising, and often unsought, useful finding. This may result in a new product, such as Viagra, which was found when researching a drug for angina; an idea, such as acid rain, which was found when investigating consequences of tree clearance; or an artefact, such as the Rosetta Stone, discovered when demolishing a wall in Egypt.

In this paper we describe serendipitous discovery firstly in a human, and secondly in a computational context, and propose a series of associated computational concepts. We follow a modified version of Jordanous's evaluation guidelines for CC (Jordanous 2012), and consider three computational case studies in terms of our concepts and standards for serendipity. We finish by discussing whether serendipity in computers is either possible or desirable, and placing our ideas in the context of related work.

Eminent scientists have emphasised the role of chance in scientific discoveries: for instance, in 1679 Robert Hooke claimed: "The greatest part of invention being but a lucky bitt (*sic*) of chance" (cited in (Van Andel 1994, p. 634)), and, in 1775, Joseph Priestly said: "That more is owing to what we call *chance* ... than to any proper design, or preconceived theory in the business" (cited in (Merton and Barber 2004, p. 162)). In 1854, Louis Pasteur made what Merton and Barber refer to as "one of the most famous remarks of all time on the role of chance" (Merton and Barber 2004, p. 162) in his opening speech as Dean of the new Faculté des Sciences at Lille: "Dans les champs de l'observation le hasard ne favorise que les esprits préparés" (cited in (Van Andel 1994, p. 634 – 635)) ("In the fields of observation, fortune favours prepared minds"). Contemporary writers on serendipity include the psychologists Nickerson: "serendipity is widely acknowledged to have played a significant role in many scientific discoveries" (Nickerson 1999, 409) and Simonton: "Serendipity is a truly general process for the origination of new ideas" (Simonton 1995, p. 469); scientific journalist Singh: "The history of science and technology is littered with serendipity" (Rond and Morley 2010, p. 66); cognitive scientist and popular CC writer Boden remarks that "Chance is held to be a prime factor in many creative acts" (Boden 1990, p. 233). Equating serendipity with unexpected findings, Dunbar and Fugelsang used observational studies of scientists "in the wild" and brain imaging studies of scientific thinking to show that over half of scientists' findings are unexpected (Dunbar and Fugelsang 2005).

The word *serendipity* was coined in 1754 by Horace Walpole, as describing a particular kind of discovery. He illustrated the concept by reference to a Persian folk tale, *The Travels and Adventures of Three Princes of Serendip*, in which the princes go travelling and together make various observations and Holmesian inferences: "They were always making discoveries, by accidents and sagacity, of things which they were not in quest of" (cited in (Merton and Barber 2004, p. 2)) One such example occurs when a camel driver asks if they have seen his lost camel, and they display such detailed knowledge of the camel that the driver accuses them of stealing it. They justify their knowledge based on their observations and abductive inferences. In the last 260 years (and the last 60 in particular), this notion of a *happy, accidental discovery* has gone from being an arcane word and concept, to being part of commonplace language.

Serendipity is a value-laden concept, and has been considered both to depreciate and enhance a scientist's achievement, leading to accounts in which the role of serendipity in a discovery is either under or overrated. Despite this difficulty, there are numerous examples of serendipity in scientific discovery, some of which have been gathered into collections ((Roberts 1989) contains over 70 examples, (Rond and Morley 2010) contains examples in cosmology, astronomy, physics and other domains, and (Van Andel 1994) claims to have over 1000 (unpublished) examples). Examples from these sources include numerous medical discoveries, when a side effect was found to be more useful

than the original goal; Kekulé's 1865 dream-inspired discovery of the structure of the benzine ring; the discovery of a Quechua man with maleria, drinking water which happened to be tainted from the bark of cinchona trees, that quinine (found in the bark) can cure maleria; Goodyear's discovery of vulcanised rubber, when trying to make a rubber resistent to temperature changes, after accidentally leaving a mixture of rubber and sulfer on a hot stove and finding that it charred, rather than melted; Penzias and Wilson's discovery of the *echoes of the Big Bang*, in which they were testing for the source of noise that a radio telescope was picking up, discovering eventually from a physicist that these were echoes from the Big Bang; and the Rosetta Stone, which was found by a soldier who was demolishing a wall in order to clear ground. We consider three examples below:

1. In 1928, while researching influenza, Fleming noticed an unusual clear patch in a petri dish of bacteria cultures. Subsequent examination revealed that the lid of the petri dish had fallen off (thus invalidating the experiment) and mould had fallen into the dish, killing the bacteria – resulting in the discovery of penicillin.

2. In 1948, on returning home from a walk, de Mestral found cockleburs attached to his jacket. While trying to pick them off, he became interested in what made them stick so tightly, and started to think about uses for a system designed on similar principles – resulting in the discovery of *Velcro*.

3. In 1974, Fry was struggling to use pieces of paper to mark pages in his choir book, when he recalled of a colleague's failed attempts to develop superglue. The colleague had accidentally made a glue so weak that two glued pieces of paper could be pulled apart – this resulted in the discovery of *Post-it* notes.

We are fortunate in that the sociologist Merton and historian Barber have written a detailed account of the word "serendipity", tracing its meaning from its coinage in 1754 to 1954 (and an extended afterword on its usage from 1954 - 2004) (Merton and Barber 2004). This is a tremendous resource for those who require an algorithmic level of detail of a hard-to-grasp concept. By basing our computational interpretation on this book we can claim that we are using the word in the same way as is used in common parlance. They highlight three things of particular interest: firstly, while Walpole was unambiguous that serendipity referred to an *unsought* finding, this criterion has dropped from dictionary definitions (only 5 out of 30 English language dictionaries from 1909 - 2000 explicitly say "not sought for" (Roberts 1989, pp. 246–249); secondly, while serendipity originally described an event (a type of discovery), it has since been reconceptualised as a psychological attribute (of the discoverer); thirdly, they argue that the psychological perspective needs to be integrated with a sociological one.[1]

[1]Serendipity is usually discussed within the context of discovery, rather than creativity: in this paper we assume an association between the two.

Serendipitous discovery in a computational context

We identify characteristics of serendipitous discovery and propose corresponding computational concepts.

The focus-shift.

Serendipitous discovery often (perhaps always) involves a shift of focus. In our examples we see focus-shifts in the context of an unsuccessful (but valid) experiment (Viagra); a mistake (leaving the lid off a petri dish, thus invalidating an experiment); previously discarded refuse (weak glue); an accident (letting rubber touch hot stove); an object which is being removed (the Rosetta Stone); and something which was considered to be a nuisance (the noise in the Big Bang example, the burs on jacket), unimportant (side effects in medical drugs), or irrelevant (a dream). In all of these cases there is a radical change in the discoverer's evaluation of what is interesting: we can think of this as a reclassification of signal-to-noise (literally, in Penzias and Wilson's case).

There is not always a main focus: for instance, de Mestral was out walking when he came across the seeds of his discovery. In cases where there is a focus, this might be abandoned in favour of a more interesting or promising direction, or may be achieved alongside the shift in focus. In computational terms we could model a focus-shift by enabling a system to "change its mind" that is, to re-evaluate an object as interesting, which it had previously judged to be uninteresting.

Components.

We break down the components in serendipitous discovery as follows:

Prepared Mind: This is the discoverer's previous experiences, background knowledge, store of unsolved problems, skills and current focus. It corresponds to the set of background knowledge, unsolved problems, current goal, and so on in a system.

Serendipity Trigger: This is the part of the examples discussed which arises immediately prior to the discovery. Examples include a dream, a petri dish with a clear area, cockleburs attached to a jacket and discarded glue. It corresponds to the example or concept in a system, which precedes the discovery.

Bridge: The techniques which enables one to go from the trigger to the result. These include reasoning techniques such as abduction (Fleming uses abductive inference to explain the surprising observation of the clear patch in petri dish); analogical reasoning (de Mestral constructed a target domain from the source domain of burs hooked onto fabric); and conceptual-blending (Kekulé blended molecule structure with a vision of a snake biting its tail and invented the concept of benzine ring). In AI, some reasoning techniques are more associated with creativity than others. For instance, analogical reasoning, conceptual-blending, genetic algorithms and automated theory formation techniques have featured heavily in CC publications. This is a good start for

the reasoning techniques we identify here. Another key attribute is the ability to perform a focus-shift at an opportune time.

Result: This is the discovery itself. This may be a new product (such as Velcro), artefact (such as the Rosetta Stone), process (vulcanisation of rubber), hypothesis (such as "penicillium kills staphylococcus bacteria"), use for an object (such as quinine), and so on. The discovery may be an example of a *sought* finding (classified by Roberts as *pseudoserendipity* (Roberts 1989, p. x)), in which case the solution arises from an unknown, unlikely, coincidental or unexpected source.

Three dimensions of serendipity.

1. **Chance:** The *serendipity trigger* is unlikely, unexpected, unsought, accidental, random, surprising, coincidental, arises independently of, and before, the result. The value of carefully controlled randomness in CC and AI systems is well-established. For instance, GA systems, which are popular in CC, employ a user-defined mutation probability, usually set to around 5-10%. Introducing randomness into search has also proved profitable in other systems. Likewise, the role that surprise plays in CC is well explored.
2. **Sagacity:** This dimension describes the attributes, or skill, on the part of the discoverer (the *bridge* between the *trigger* and the *result*). In many of these examples others had been in the same position and not made the discovery. This skill involves an *open mind* (an ability to take advantage of the unpredictable); ability to focus-shift; appropriate reasoning techniques; and ability to recognise value in the discovery.
3. **Value:** The *result* must be happy, useful (evaluated externally). Measuring the value of a system's results is a well-known problem in CC, and can be evaluated independently of the programmer and system or (as is more common) by the programmer alone.

A discovery does not have to score highly on each axis to be considered serendipitous. The chances of an unanticipated use being found for a drug under development may be quite high (i.e., the role that chance plays in such a discovery is low), and the sagacity needed to discern that quinine-infused water has cured malaria may be low. While the discoveries that Walpole describes were not always important, the examples given today (in (Roberts 1989; Rond and Morley 2010; Van Andel 1994)) describe valuable, often domain-changing, discoveries. Arguably, the discovery of penicillin is the most serendipitous of our examples, since two improbable events were involved: the combination of penicillium mould and staphylococcus bacteria, and the accident of the petri dish lid falling off; it took great skill to recognise the importance of the observation, and – having saved millions of lives – it is clearly of great value.

Environmental factors.

As Merton and Barber point out, serendipitous discovery is not achieved in isolation. The discoverer is operating in a messy world and engaged in a range of activities and experiences. We propose the following characteristics of the discoverers' environments, and computational analogs:

1. **Dynamic world:** Data was presented in stages, not as a complete, consistent whole. This corresponds to streaming from live media such as the web.
2. **Multiple contexts:** Information from one context, or domain was used in another. This is a common notion in analogical reasoning.
3. **Multiple tasks:** Discoverers were often involved in multiple tasks. This corresponds to threading, or distributed computing.
4. **Multiple influences:** All discoveries took place in a social context, and in some examples the "unexpected source" was another person. This corresponds to systems such as agent architectures, in which different software agents with different knowledge and goals interact.

The three-step model of SPECS.

Jordanous summarises her evaluation guidelines in three steps; to identify a definition of creativity, state evaluation standards, and apply the standard to your creative system (Jordanous 2012). Here we apply these steps to the notion of *serendipity*.

Step 1: Identify a definition of serendipity that your system should satisfy to be considered serendipitous. We propose the following definition of computational serendipitous discovery:

Computational serendipitous discovery occurs when a) within a system with a prepared mind, a previously uninteresting serendipity trigger arises partially due to chance, and is reclassified as interesting by the system; and b) when the system, by processing this re-evaluated trigger and background information together with abductive, analogical or conceptual-blending reasoning techniques, obtains a new result that is considered useful both by the system and by external sources.

Step 2: Using Step 1, clearly state what standards you use to evaluate the serendipity of your system. With our definition in mind, we propose the following standards for computational serendipity:

Evaluation standard 1: (i) The system has a prepared mind, consisting of previous experiences, background knowledge, a store of unsolved problems, skills and (optionally) a current focus or goal. (ii) The serendipity trigger arises partially as a result of chance factors such as randomness, independence of the end result, unexpectedness, or surprisingness.

Evaluation standard 2: The system: (i) uses reasoning techniques associated with serendipitous discovery: abduction, analogy, conceptual-blending; (ii) performs a focus-shift; (iii) evaluates its discovery as useful.

Evaluation standard 3: As a consequence of the focus-shift, a result which is evaluated as useful by an external source is found.

Step 3: Test your serendipitous system against the standards stated in Step 2 and report the results. In the following section we evaluate three systems against our standards.

Computational Case Studies

Armed with an analysis of serendipity in computational settings, we investigate here the value of these insights with respect to past, present and future creative systems. In particular, we describe and evaluate from a serendipity perspective: (a) an abductive reasoning system which has already been employed in a different context (b) a series of experiments with the HR automated theory formation system aimed at promoting serendipitous discovery, and (c) a proposed an extension to a framework for creative currently under development.

The GH system

Our first system models the sort of reasoning initially described by Walpole in the Princes of Serendip story. As described in (Ramezani and Colton 2010), *Dynamic Investigation Problems* (DIPs) are a type of hybrid AI problem specifically designed to model real life situations where a guilty party has to be chosen from a number of suspects, with the decision depending on a changing (dynamic) set of facts and constraints about the current case and a changing set of case studies of a similar nature to the current case. Such situations occur in criminal or medical investigations, for instance, and the GH solver has been named after the fictional medical investigator Gregory House, although his namesake of Sherlock Holmes would equally suffice. DIPs have been designed to be unsolvable either by machine learning rules from the case studies or solving the constraints as a Constraint Satisfaction Problem, hence requiring a hybrid learning and constraint solving approach.

The GH system is given facts about a current investigation, in the form of predicates known to be true which relate various attributes of the guilty suspect but do not identify it. The problems are noisy in that only some of these facts are pertinent to finding the guilty suspect and (optionally) some facts which are required are missing. GH is also given similar facts about a number of previous cases which are related in nature to the current case, with the facts given again in predicate form. The facts of the current case and those of the case studies are given in blocks at discrete time steps, and the software solves the partial problems at each time step. To find the solutions, the facts of the current case are interpreted as a CSP to be solved by the CLPFD solver in Sicstus Prolog. Before it attempts to find a solution, GH maps the attributes of the previous cases onto those of the current case, and then uses association rule mining via the Weka machine learning package to find empirically true relationships between the attributes described in the facts. These relationships are selectively added to the CSP in order to find a more precise solution. The DIPs are set up so that the CSP without the extra constraints can be solved by multiple suspects, while – if the correct extra constraints are mined from the case studies – there is only one correct solution. Presenting further details of DIPs or the GH system is beyond the scope of this paper, but suffice to say, we performed a series of experiments to explore the nature DIPs and the solutions that GH can find. For instance, when the DIPs have 4 pertinent constraints of arity five or less, and 100% of the constraints are available either in the current case or hidden in the case studies, GH has an error rate (i.e., choosing the wrong subject) of 10%. When only 50% of the pertinent facts can be found, the error rate rises to 31%.

Standard 1: (i) The system has a *prepared mind* consisting of past cases, background knowledge and an unsolved problem. (ii) the *serendipity trigger* corresponds to a new piece of data which means that a previous case is now relevant. *Chance* factors arise in the order and which data the system receives.

Standard 2: (i) The system uses induction, abduction and constraint solving as reasoning techniques; its abductive procedures are of particular interest. (ii) *Focus-shifts* can occur if a previous case is re-evaluated by the system as relevant to the current case. (iii) The *result* is the diagnosis or identification of the guilty party, and is judged by the system to be correct.

Standard 3: As a consequence of the previously irrelevant case being re-evaluated as relevant, the diagnosis is achieved. *Value* consists in external evaluations of whether the system has reached the correct solution.

Additionally, the environmental factors are partially well represented: the system operates in a *dynamic world*; and we can see reasoning about different cases as operating in *multiple contexts*. However, it only solves one *task* at a time, and there are not currently *multiple influences*.

Experiments in model generation

The HR program (Colton 2002) is an automated theory formation system which, starting with background knowledge describing concepts and examples of those concepts, uses production rules iteratively to construct new concepts from old ones. It forms conjectures empirically which relate one or more concepts, and evaluates concepts and conjectures using a number of measures of interestingness, which in turn drives a best-first heuristic search whereby the most interesting old concepts are used to produce new concepts. For instance, the *complexity* of a concept is the number of production rule steps that were used in its production, and the complexity of a conjecture is the average of the complexity of the concepts it relates. When working in domains of pure mathematics for which axioms are given, HR can interface with the Davis-Putnam style model generator MACE and the resolution theorem prover Otter to attempt to disprove/prove empirical conjectures respectively. Working in domains of finite algebra, we started HR with only the axioms of the domain, and the background concepts required to express those axioms. In particular, HR was given no example algebras, and hence each algebra introduced to the theory came as a counterexample to a false conjecture the software made due to lack of data. In all sessions, we used modest time resources for using MACE (5 secs) and Otter (3 secs).

HR was enhanced so that whenever it found a counterexample to a new false conjecture, it tested to see whether that counterexample broke any previously unsolved *open* conjecture (i.e., for which MACE could previously find no counterexample and Otter could find no proof). We found that such occurrences were very rare. In the three test domains of group theory (associativity, identity and inverse axioms), monoid theory (associativity, identity) and semigroup theory (associativity), when run in breadth first mode, i.e., with no heuristic search, we never observed this behaviour during sessions with tens of thousands of production rule steps. This is because the search strategy means that usually the simplest concepts and hence the simplest conjectures were made early on during the session, and as became increasingly harder to find counterexamples to the progressively more difficult false conjectures, it was never the case that a later conjecture was disproved with a counterexample that also disproved an earlier one.

To attempt to encourage the re-use of counterexamples, we ran random search strategies, whereby the next concept to use in production rule steps was chosen randomly, subject to a complexity limit of 10. This strategy worked for monoids and semigroups, but not for group theory. As an example, in monoid theory, after 1532 steps, this conjecture:

$\forall b, c, d(((b * c = d \wedge b * d = c \wedge d * b = c \wedge c * d = b \wedge (d \neq id)) \leftrightarrow$
$(b * c = d \wedge d * b = c \wedge c * d = b \wedge (d \neq id))))$

was disproved by MACE finding a counterexample. The counterexample also broke this previous open conjecture:

$\forall b, c, d(((b * c = d \wedge c * b = d \wedge c * d = b \wedge (\exists e(e * c = d \wedge e * d = c))) \leftrightarrow (b * c = d \wedge (\exists f(b * c = f)) \wedge (\exists g(g * c = b)) \wedge d * b = c \wedge c * d = b)))$

This was the sole example we saw in 2000 theory formation steps in monoid theory. In semigroup theory, such events were more common: there were three times when a new counterexample was used to solve a single open conjecture, and on one occasion ten open conjectures were disproved by one counterexample.

Standard 1: (i) In these experiments HR develops a *prepared mind* during the run. The background knowledge is user-given concepts, the examples which have arisen during the run and all of the developed concepts and conjectures. The open conjectures constitute the store of unsolved problems, the skills are the production rules and other procedural mechanisms. At the point just before the *serendipity trigger*, the counterexample which arose in the context of the low complexity conjecture, the current focus is to prove or disprove the current conjecture. (ii) While there is no randomness in the way that MACE generates the *serendipity trigger*, in the random runs there is randomness in the way that the conjecture which prompted the new example was generated. In addition, the example was generated independently of the end result.

Standard 2: (i) The system did not use any of the three reasoning techniques. (ii) It did re-evaluate the previously unsolved conjecture, once it was solved, but this was not the reason that focus shifted.

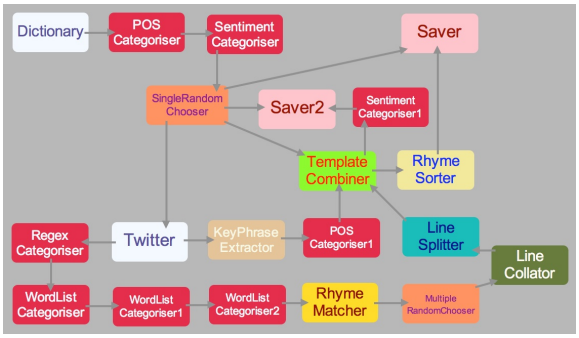


Figure 1: A poetry generating flowchart.

Standard 3: The *result* was the now-solved (previously open) conjecture. Apart from the fact that a theorem generally has higher status in mathematics than an open conjecture, we cannot claim that the solved conjectures were interesting. (None of them would appear in a textbook on group theory.) However, we can claim that, in this mode, if it was not for the example arising in a different context, the system would not have been able to solve the 18 open conjectures. We know this since it had already attempted to and failed within the time limits.

A flowcharting framework

In a project separate from our work on serendipity, we are building a flowcharting system to be used for Computational Creativity projects. Each node in the flowcharts undertakes a particular task on data types such as text and images, and the task can be generative or evaluative, or it could bring back data from websites or local databases. Without going into detail, the example flowchart in figure generates poems by compiling tweets mined from Twitter using a single adjective W as a search term, employing sentiment analysis and a rhyming dictionary along the way. The following is a stanza from a poem generated by the flowcharting system using this flowchart, where W was *malevolent*:

I hear the souls of the damned wailing in hell.
I feel a malevolent spectre hovering just behind me.
It must be his birthday.
Is God willing to prevent evil, but not able?
Then he is not omnipotent.
Is he able, but not willing?
Then he is malevolent.
It's only when his intelligence grows and he understands the laws of man that
He becomes malevolent and violent.
I don't find it malevolent, I find it affectionate.
Geeks do weird things and that can be hilarious for different reasons.

One of the purposes of the flowchart project is to have a platform for the development of creative systems that the whole Computational Creativity community to contribute to and benefit from. Our aim is to have a number of people developing nodes locally at various sites worldwide, then uploading them for everyone to share in building their own

flowcharts via a GUI. We are specifically aiming for a domain independent framework, and to this end, our *inspiring examples* in building the system are the theory formation abilities of the HR system (Colton 2002), the painting abilities of The Painting Fool system (Colton 2013) and the poetry generation abilities described in (Colton, Goodwin, and Veale 2012). We currently have flowcharts which approximate the functioning of the original systems in all three cases.

Another main purpose of the project is to explore ways in which the software can automatically construct flowcharts itself - so that it can innovate at the process level. It is beyond the scope of this paper to describe how this will be done in detail, but one fact is pertinent: if/when such automated construction is possible, we will situate a version of the software on a server, constantly generating, testing and evaluating the flowcharts it produces, and making the artefacts it produces available, along with framing information (Charnley, Pease, and Colton 2012) about the process and the product. As new nodes are developed, they will be automatically made available to the system, and flowcharts will immediately be formed which utilise the new node.

The dynamic nature of this framework is clear: nodes will be accessing web services, so the data being used will be constantly changing; existing nodes will be updated and new nodes will be uploaded regularly; and new flowcharts will be created rapidly. In fact, we aim to increase this dynamic nature by having multiple such systems residing on various servers around the world, swapping nodes, flowcharts, outputs and meta-level information at regular intervals. We believe that this will increase the likelihood of chance encounters occurring to expect serendipity to follow. Moreover, the framework is not domain specific, and we will encourage the building of nodes which transfer information, say, from visual arts outputs to textual inputs, and vice versa. Thus, the environmental factors are extremely well-represented: the system operates in a *dynamic world* as it brings back data from websites or local databases, such as streaming from Twitter; the domain-independent aspect ensures that is can operate in *multiple contexts* (these will be concurrent, as in the example given in which the contexts are theory formation, painting and poetry). At any time-point there will be multiple *tasks* being undertaken by the various nodes, and, by feeding into each other these will provide *multiple influences*. We believe this will increase the likelihood of results/ideas/processes in one domain being serendipitously applied in another domain, hopefully with happy consequences.

Standard 1: (i) If we view the flowcharting system as a whole, then the *prepared mind* will be constructed via the nodes, consisting of the knowledge in the system at any time and the generative and evaluative procedures which the nodes are able to perform. Current goals will be the particular tasks that each node is involved in. (ii) The *serendipity trigger* to a particular node will arise via new information (for instance, from streaming such as Twitter) or sharing from other nodes. The sharing and updating could have a random element to it, but the main factor relating to *chance* will be that new information will arise in independent contexts, and thus will be independent of final *results*.

Standard 2: (i) As a platform for the development of creative systems that the whole CC community will contribute to and benefit from, the system as a whole will perform a variety of techniques, in particular those associated with creativity. Therefore, we expect that it will be able to perform abduction, analogy and conceptual-blending. (ii) The task that each node undertakes can be evaluative, and, if the system can perform automated construction of the flowcharts itself, it will constantly be evaluating the flowcharts it produces. Thus, *focus-shifts* should be possible; (iii) likewise, nodes will evaluate their own *results* (the artefacts that they produce).

Standard 3: The artefacts produced, such as the poem above, will be evaluated by external sources to determine the success of the whole project.

Discussion

With respect to the dynamic investigation problem and the model generation experiments described above, we can say that the former is realistic but not particularly serendipitous, while the latter is more serendipitous, but more artificial - in fact, we had to willingly make the system less effective to encourage incidents which onto which we might project the word serendipity. This raises the question of whether it is indeed possible to set up a computational situation within which such incidents genuinely occur. The flowchart system is the most promising in terms of making serendipitous discoveries. Of course, the evaluation standards themselves should be subject to evaluation, to make sure that they both reflect our intuitive notion of serendipity and are practical to apply to our CC systems.

We assume that in CC we are aiming to develop software which can surprise us, generate culturally valuable artefacts, and produce a good story about how it constructed the artefacts. There is tension between systematicity and serendipity, and it may be the case that incorporating serendipity into a creative system inhibits its ability to produce the desired artefacts. We take seriously the concern that modelling serendipity in CC may be either impossible or undesirable.

One can argue that, given the role of chance in serendipity, it is impossible to program such discoveries. Like have-a-go heroes, serendipity in our systems should be cherished but not encouraged. In response to such arguments, we have tried to characterise the sorts of environments which enhance the likelihood of making a chance discovery, and we have suggested computational analogs. Serendipity is not "mere chance" – the axes of sagacity and useful results are equally important. That serendipity-facilitating skills can be taught to people is not a new argument – much work written by scientists on serendipity is designed to teach others what skills are involved (see also (Lenox 1985)). Many (perhaps all) of the skills are standard skills of a scientist, and it may be argued that relevant machine learning techniques, such as anomaly detection and outlier analysis, already exist. We suggest that such techniques will be extremely useful, but

probably not sufficient, for computational serendipitous discovery.

One might also argue that the same characteristics which aid serendipity would also aid negative serendipity. A system which allowed itself to be derailed from a task at hand might not achieve as much as one which maintains focus. Negative serendipity can be defined in various ways: Pek defines it as when: "A surprising fact or relation is seen but not (optimally) investigated by the discoverer", giving Columbus' lifelong belief that he had found a new route to Asia, rather than a new continent, as an example (Van Andel 1994, p. 369). We can also define it as a discovery which is *prevented* due to chance factors: this would be very hard to demonstrate, but relates to the "Person from Porlock" syndrome, where creative flow is interrupted due to an unwanted interruption. As well as negative serendipity, one might argue that a reliance on serendipity contrasts intelligence, and a system which uses a random search may exhibit less intelligent behaviour than one which follows a well developed heuristic search. Thus, in our HR experiment, enhancing its serendipity was a retrograde step for the system. We certainly would not advocate that all CC developers add serendipitous functionality to their existing software, which might detract from other functionality. Despite this, we suggest that serendipity is a feature which can be both possible and useful to model in future creative systems.

The examples of human serendipity all describe groundbreaking discoveries. In CC, we have learned that we must not aim to build systems which perform domain-changing acts of creativity, before systems which can perform everyday, mundane creativity (distinguished as "Big C" and "little c" creativity.) Similarly, we must expect to model "little s" serendipity before we are able to model "Big S" serendipity. The dimension which this affects the most is the third one – we must not expect the discoveries to be rated too highly with our embryonic models of computational serendipity. A useful intermediate way of evaluating the results might be with respect to other, non-serendipitous, results.

Related work

Many of the aspects we have identified as inherent in serendipitous discovery are already widespread computational techniques, and there are large bodies of work which will be particularly relevant. For instance, research into the role of problem reformulation in problem-solving, such as (Griffith, Nersessian, and Goel 2000), is relevant to the *focus-shift* aspect in that reformulation can trigger new solutions and re-evaluations. Our notion of focus-shift differs from problem reformulation, in that the focus may be on examples, artefacts, etc rather than problems, and the result of a focus-shift is a re-evaluation rather than re-representation. Problem-shift, where a problem evolves alongside possible solutions (see, for instance, (Helms and Goel 2012)), is also relevant.

Wills and Kolodner (Wills and Kolodner 1994) have analysed the processes involved in serendipitous recognition of solutions to suspended design problems, where the solutions overcome both functional fixedness and fixation on standard solutions. They propose a computational model which is based on the hypothesis that recognition arises from interaction between the processes of problem evolution and assimilation of proposed ideas into memory. Their analysis fits into our *sagacity* dimension as they elaborate skills needed to recognise value in unexpected places, and in particular ways in which the *focus-shift* can work.

There is related work on chance. For instance, Campbell's model of creativity, "blind variation and selective retention" (described in (Simonton 1999)), in which he draws an analogy between biological evolution and creativity, seems to be particularly pertinent for serendipity, with its emphasis on "blind" (Campbell elaborates his use of the term and discusses other candidates, including: chance, random, aleatory, fortuitous, haphazard, unrestricted, and spontaneous). This corresponds to our notion of *chance*.

Serendipity was formalised by Figueiredo and Campos in their paper 'The Serendipity Equations' (Figueiredo and Campos 2001). This paper used logical equations to describe the subtle differences between some of the many forms of serendipity. In practice none of the implemented examples rely on the computer to be the prepared mind. It is the user that is expected to have the 'aha' moment and thus the creative step. The computer is used to facilitate this by searching outside of the normal search parameters to engineer potentially serendipitous (or at least pseudo–serendipitous) encounters. One example of this is 'Max' created by Figueiredo and Campos (Campos and Figueiredo 2002). Here the user emails Max with a list of interests and Max finds a webpage that may be of interest to the user. Max expands the search parameters by using WordNet[2] to generate synsets for words of interest. Max also has the ability to wander; taking information from the first set of results and using these to find further pages. Other search examples include searching for analogies (Donoghue and Crean July 2002) and content (Iaquinta et al. 2008). These all use different strategies to provide new and potentially serendipitous information to the user (who must be the "prepared mind").

Further work and conclusions

The notion of serendipitous discovery is a popular and rather romantic one. Thus, when scientists or artists are framing their work for public consumption, they might tell a backstory about the role that serendipity played, which might enhance our perception of the value of the discovery or discoverer. In (Charnley, Pease, and Colton 2012), we outline the importance of producing framing information in CC. While the account of a discovery can be fictional (and thus could refer to a serendipity which did not happen), incorporating it into discovery mechanisms could result in richer framing information.

Challenging the idea that only humans can be serendipitous is a problem which is familiar to CC researchers. In the case of serendipity this may be even greater, since the notion of designing for serendipity can appear to be oxymoronic. Our message in this paper is that we should *proceed with caution* in this intriguing area.

[2] http://wordnet.princeton.edu/

Acknowledgements

We would like to thank our three reviewers who gave particularly thorough reviews and useful references. This research was funded by EPSRC grant EP/J004049.

References

Boden, M. 1990. *The Creative Mind: Myths and Mechanisms*. London: Weidenfield and Nicholson.

Campos, J., and Figueiredo, A. D. 2002. Programming for serendipity. In *Proc. of the AAAI Fall Symposium on Chance Discovery – The Discovery and Management of Chance Events*.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proc. of the 3rd ICCC*, 77–81.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. In *Proc of the 3rd ICCC*.

Colton, S. 2002. *Automated Theory Formation in Pure Mathematics*. Springer-Verlag.

Colton, S. 2013. The Painting Fool: Stories from building an automated painter. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity (forthcoming)*. Springer-Verlag.

Donoghue, D., and Crean, B. July, 2002. Searching for Serendipitous Analogies. In *European Conference on Artifical Intelligence (ECAI), Workshop on Creative Systems*.

Dunbar, K., and Fugelsang, J. 2005. Causal thinking in science: How scientists and students interpret the unexpected. In Gorman, M. E.; Tweney, R. D.; Gooding, D.; and Kincannon, A., eds., *Scientific and technical thinking*. Mahwah, NJ: Erlbaum. 57–79.

Figueiredo, A. D., and Campos, J. 2001. The Serendipity Equations. In Weber, R., and von Wangenheim, C. G., eds., *Proc. of ICCBR-4*.

Griffith, T. W.; Nersessian, N. J.; and Goel, A. 2000. Function-follows-form transformations in scientific problem solving. In *Prc. of the 22nd Annual Conference of the Cognitive Science Society*, 196–201. Cognitive Science Society.

Helms, M. E., and Goel, A. K. 2012. Analogical problem evolution in biologically inspired design. In *Design Computing and Cognition DCC'12 (J.S. Gero (ed))*. Springer.

Iaquinta, L.; Gemmis, M.; Lops, P.; Semeraro, G.; Filannino, M.; and Molino, P. 2008. Introducing Serendipity in a Content-Based Recommender System. *8th Int. Conf. on Hybrid Intelligent Systems* 168–173.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Lenox, R. S. 1985. Educating for the serendipitous discovery. *Journal of Chemical Education* 62(4):282–285.

Merton, R. K., and Barber, E. 2004. *The Travels and Adventures of Serendipity: A study in Sociological Semantics and the Sociology of Science*. New Jersey, USA: Princeton University Press.

Nickerson, R. S. 1999. Enhancing creativity. In Sternberg, R. J., ed., *Handbook of Creativity*. Cambridge, UK: Cambridge University Press. 392–430.

Ramezani, R., and Colton, S. 2010. Automatic generation of dynamic investigation problems. In *Proc of ARW*.

Roberts, R. M. 1989. *Serendipity: Accidental Discoveries in Science*. USA: John Wiley and Sons, Inc.

Rond, M. d., and Morley, I. 2010. *Serendipity: Fortune and the Prepared Mind (Darwin College Lectures)*. Cambridge University Press.

Simonton, D. K. 1995. Foresight in insight? In Sternberg, R. J., and Davidson (Eds.), J. E., eds., *The nature of insight*. MIT. 465–494.

Simonton, D. K. 1999. Creativity as blind variation and selective retention: Is the creative process darwinian? *Psychological Inquiry* 10(4):309–328.

Van Andel, P. 1994. Anatomy of the unsought finding. *The British Journal for the Philosophy of Science* 45(2):pp. 631–648.

Wills, L. M., and Kolodner, J. L. 1994. Explaining serendipitous recognition in design. In *Proc. of the 16th Annual Conference of the Cognitive Science Society*. Atlanta, GA.

Considering Vertical and Horizontal Context in Corpus-based Generative Electronic Dance Music

Arne Eigenfeldt
School for the Contemporary Arts
Simon Fraser University
Vancouver, BC
Canada

Philippe Pasquier
School of Interactive Arts and Technology
Simon Fraser University
Surrey, BC
Canada

Abstract

We present GESMI (Generative Electronica Statistical Modeling Instrument) – a computationally creative music generation system that produces Electronic Dance Music through statistical modeling of a corpus. We discuss how the model requires complex interrelationships between simple patterns, relationships that span both time (horizontal) and concurrency (vertical). Specifically, we present how context-specific drum patterns are generated, and how auxiliary percussion parts, basslines, and drum breaks are generated in relation to both generated material and the corpus. Generated audio from the system has been accepted for performance in an EDM festival.

Introduction

Music consists of complex relationships between its constituent elements. For example, a myriad of implicit and explicit rules exist for the construction of successive pitches – the rules of melody (Lerdahl and Jackendoff 1983). Furthermore, as music is time-based, composers must take into account how the music unfolds: how ideas are introduced, developed and later restated. This is the concept of musical form – the structure of music in time. As these relationships are concerned with a single voice, and thus monophonic, we can consider them to be horizontal[1].

Similarly, relationships between multiple voices need to be assessed. As with melody, explicit production rules exist for concurrent relationships – harmony – as well as the relationships between melodic motives: polyphony. We can consider these relationships to be vertical (see Figure 1).

[1] The question of whether melody is considered a horizontal or vertical relationship is relative to how the data is presented: in traditional music notation, it would be horizontal; in sequencer (list) notation, it would be vertical. For the purposes of this paper, will assume traditional musical notation.

Music has had a long history of applying generative methods to composition, due in large part to the explicit rules involved in its production. A standard early reference is the *Musikalsches Würfelspiel* of 1792, often attributed to Mozart, in which pre-composed musical sections were assembled by the user based upon rolls of the dice (Chuang 1995); however, the "Canonic" compositions of the late 15th century are even earlier examples of procedural composition. In these works, a single voice was written out, and singers were instructed to derive their own parts from it by rule: for example, singing the same melody delayed by a set number of pulses, or at inversion (Randel 2003).

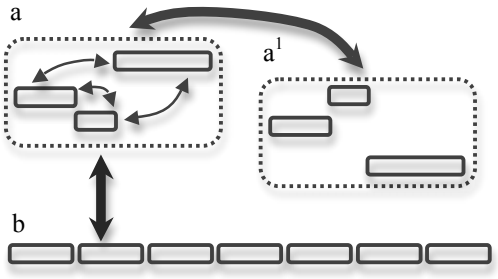


Figure 1. Relationships within three musical phrases, *a, a[1], b*: melodic (horizontal) between pitches within *a*; formal (horizontal) between *a* and *a[1]*; polyphonic (vertical) between *a* and *b*.

Exploring generative methods with computers began with some of the first applications of computers in the arts. Hiller's *Illiac Suite* of 1956, created using the Illiac computer at the University of Champaign-Urbana, utilized Markov chains for the generation of melodic sequences (Hiller and Isaacson 1979). In the next forty years, a wide variety of approaches were investigated – see (Papadopoulos and Wiggins 1999) for a good overview of early uses of computers within algorithm composition. However, as the authors suggest, "most of these systems deal with algorithmic composition as a problem solving task rather than a creative and meaningful process". Since that time, this separation has continued: with a few exceptions (Cope 1992, Waschka 2007, Eigenfeldt and Pasquier 2012), contemporary algorithmic systems that employ AI methods

remain experimental, rather than generating complete and successful musical compositions.

The same cannot be said about live generative music, sometimes called interactive computer music due to its reliance upon composer or performer input during performance. In these systems (Chadabe 1984, Rowe 1993, Lewis 1999), the emphasis is less upon computational experimentation and more upon musical results. However, many musical decisions – notably formal control and polyphonic relationships – essentially remain in the hands of the composer during performance.

Joel Chadabe was the first to interact with musical automata. In 1971, he designed a complex analog system that allowed him to compose and perform *Ideas of Movement at Bolton Landing* (Chadabe 1984). This was the first instance of what he called interactive composing, "a mutually influential relationship between performer and instrument." In 1977, Chadabe began to perform with a digital synthesizer/small computer system: in *Solo*, the first work he finished using this system, the computer generated up to eight simultaneous melodic constructions, which he guided in realtime. Chadabe suggested that *Solo* implied an intimate jazz group; as such, all voices aligned to a harmonic structure generated by the system (Chadabe 1980).

Although the complexity of interaction increased between the earlier analog and the later digital work, the conception/aesthetic between *Ideas of Movement at Bolton Landing* and *Solo* did not change in any significant way. While later composers of interactive systems increased the complexity of interactions, Chadabe conceptions demonstrate common characteristics of interactive systems:

1. Melodic constructions (horizontal relationships) are not difficult to codify, and can easily be "handed off" to the system;
2. harmonic constructions (vertical relationships) can be easily controlled by aligning voices to a harmonic grid, producing acceptable results;
3. complex relationships between voices (polyphony), as well as larger formal structures of variation and repetition, are left to the composer/performer in realtime.

These limitations are discussed in more detail in Eigenfeldt (2007).

GESMI (Generative Electronica Statistical Modeling Instrument) is an attempt to blend autonomous generative systems with the musical criteria of interactive systems. Informed by methods of AI in generating horizontal relationships (i.e. Markov chains), we apply these methods in order to generate vertical relationships, as well as high-level horizontal relationships (i.e. form) so as to create entire compositions, yet without the human intervention of interactive systems.

The Generative Electronica Research Project (GERP) is an attempt by our research group – a combination of scientists involved in artificial intelligence, cognitive science, machine-learning, as well as creative artists – to generate stylistically valid EDM using human-informed machine-learning. We have employed experts to hand-transcribe 100 tracks in four genres: Breaks, House, Dubstep, and Drum and Bass. Aspects of transcription include musical details (drum patterns, percussion parts, bass lines, melodic parts), timbral descriptions (i.e. "low synth kick, mid acoustic snare, tight noise closed hihat"), signal processing (i.e. the use of delay, reverb, compression and its alteration over time), and descriptions of overall musical form. This information is then compiled in a database, and analysed to produce data for generative purposes. More detailed information on the corpus is provided in (Eigenfeldt and Pasquier 2011).

Applying generative procedures to electronic dance music is not novel; in fact, it seems to be one of the most frequent projects undertaken by nascent generative musician/programmers. EDM's repetitive nature, explicit forms, and clearly delimited style suggest a parameterized approach.

Our goal is both scientific and artistic: can we produce complete musical pieces that are modeled on a corpus, and indistinguishable from that corpus' style? While minimizing human/artistic intervention, can we extract formal procedures from the corpus and use this data to generate all compositional aspects of the music so that a perspicacious listener of the genre will find it acceptable? We have already undertaken empirical validation studies of other styles of generative music (Eigenfeldt et al 2012), and now turn to EDM.

It is, however, the artistic purpose that dominates our motivation around GESMI. As the authors are also composers, we are not merely interested in creating test examples that validate methods. Instead, the goals remain artistic: can we generate EDM tracks and produce a full-evening event that is artistically satisfying, yet entertaining for the participants? We feel that we have been successful, even at the current stage of research, as output from the system has been selected for inclusion in an EDM concert[2] as well as a generative art festival[3].

Related Work

Our research employs several avenues that combine the work of various other researchers. We use Markov models to generate horizontal continuations, albeit with contextual constraints placed upon the queries. These constraints are learned from the corpus, which thus involve machine-learning. Lastly, we use a specific corpus, expert-transcribed EDM in order to generate style-specific music.

Markov models offer a simple and efficient method of deriving correct short sequences based upon a specific corpus (Pachet et al. 2011), since they are essentially quoting portions of the corpus itself. Furthermore, since the models are unaware of any rules themselves, they can be quickly adapted to essentially "change styles" by switching the corpus. However, as Ames points out (Ames 1989), while simple Markov models can reproduce the surface features

[2] http://www.metacreation.net/mumewe2013/
[3] http://xcoax.org/

of a corpus, they are poor at handling higher-level musical structures. Pachet points out several limitations of Markov-based generation, and notes how composers have used heuristic measures to overcome them (Pachet et al. 2011). Pachet's research aims to allow constraints upon selection, while maintaining the statistical distribution of the initial Markov model. We are less interested in maintaining this distribution, as we attempt to explore more unusual continuations for the sake of variety and surprise.

Using machine-learning for style modeling has been researched previously (Dubnov et al. 2003), however, their goals were more general in that composition was only one of many possible suggested outcomes from their initial work. Their examples utilized various monophonic corpora, ranging from "early Renaissance and baroque music to hard-bop jazz", and their experiments were limited to interpolating between styles rather than creating new, artistically satisfying music. Nick Collins has used music information retrieval (MIR) for style comparison and influence tracking (Collins 2010).

The concept of style extraction for reasons other than artistic creation has been researched more recently by Tom Collins (Collins 2011), who tentatively suggested that, given the state of current research, it may be possible to successfully generate compositions within a style, given an existing database.

Although the use of AI within the creation of EDM has been, so far, mainly limited to drum pattern generation (for example, Kaliakatsos-Papakostas et al. 2013), the use of machine-learning within the field has been explored: see (Diakopoulos 2009) for a good overview. Nick Collins has extensively explored various methods of modeling EDM styles, including 1980s synth-pop, UK Garage, and Jungle (Collins 2001, 2008).

Our research is unique in that we are attempting to generate full EDM compositions using completely autonomous methods informed by AI methods.

Description

We have approached the generation of EDM as a producer of the genres would: from both a top-down (i.e. form and structure) and bottom-up (i.e. drum patterns) at the same time. While a detailed description of our formal generation is not possible here (see Eigenfeldt and Pasquier 2013 for a detailed description of our evolutionary methods for form generation), we can mention that an overall form is evolved based upon the corpus, which determines the number of individual patterns required in all sixteen instrumental parts, as well as their specific relationships in time. It is therefore known how many different patterns are required for each part, and which parts occur simultaneously – and thus require vertical dependencies – and which parts occur consecutively, and thus require horizontal dependencies.

The order of generation is as follows:

1. Form – the score, determining which instruments are active for specific phrases, and their pattern numbers;
2. Drum Patterns – also called *beats*[4] (kick, snare, closed hihat, open hihat);
3. Auxiliary percussion – (ghost kick/snare, cymbals, tambourine, claps, shakers, percussive noises, etc.) generation is based upon the concurrent drum patterns;
4. Bassline(s) – onsets are based upon the concurrent drum pattern, pitches are derived from associated data;
5. Synth and other melodic parts – onsets are based upon bassline, pitches are derived from associated data. All pitch data is then corrected according to an analysis of the implied harmony of the bassline (not discussed here);
6. Drum breaks – when instruments stop (usually immediately prior to a phrase change, and a pattern variation (i.e. drum fill) occurs;
7. One hits – individual notes and/or sounds that offer colour and foreground change that are not part of an instrument's pattern (not discussed here).

Drum Pattern Generation

Three different methods are used to generate drum patterns, including:

1. Zero-order Markov generation of individual subparts (kick, snare, closed hihat, and open hihat);
2. first-order Markov generation of individual subparts;
3. first-order Markov generation of combined subparts.

In the first case, probabilities for onsets on a given beat subdivision (i.e. sixteen subdivisions per four beat measure) are calculated for each subpart based upon the selected corpus (see Figure 2). As with all data derived from the corpus, the specific context is retained. Thus, if a new drum pattern is required, and it first appears in the main verse (section C), only data derived from that section is used in the generation.

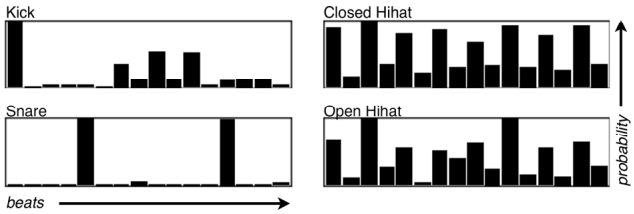


Figure 2. Onset probabilities for individual subparts, one measure (sixteenth-note subdivisions), main verse (C section), "Breaks" corpus.

In the second case, data is stored as subdivisions of the quarter note, as simple on/off flags (i.e. 1 0 1 0) for each subpart, and separate subparts are calculated independ-

[4] The term "beat" has two distinct meanings. In traditional music, beat refers to the basic unit of time – the pulse of the music – and thus the number of subdivisions in a measure; within EDM, beat also refers to the combined rhythmic patterns created by the individual subparts of the drums (kick drum, snare drum, hi-hat), as well as any percussion patterns.

ently. Continuations[5] are considered across eight measure phrases, rather than limited to specific patterns: for example, the contents of an eight measure pattern are considered as thirty-two individual continuations, while the contents of a one measure pattern that repeats eight times are considered as four individual continuations with eight instances, because they are heard eight separate times. As such, the inherent repetition contained within the music is captured in the Markov table.

In the third case, data is stored as in the second method just described; however, each subpart is considered 1 bit in a 4-bit nibble for each subdivision that encodes the four subparts together:

bit 1 = open hihat;
bit 2 = closed hihat;
bit 3 = snare;
bit 4 = kick.

This method ensures that polyphonic relationships between parts – vertical relationships – are encoded, as well as time-based relationships – horizontal relationships (see Figure 3).

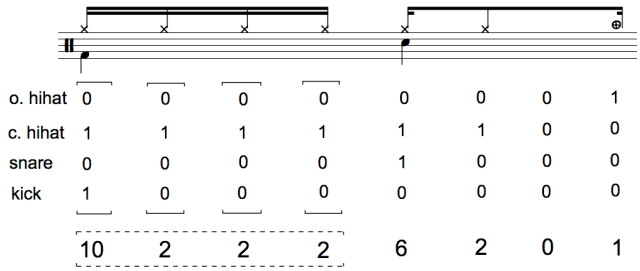


Figure 3. Representing the 4 drum subparts (of two beats), as a 4-bit nibble (each column of the four upper rows), translated to decimal (lower row), for each sixteenth-note subdivision. These values are stored as 4-item vectors representing a single beat.

It should be noted that EDM rarely, if ever, ventures outside of sixteenth-note subdivisions, and this representation is appropriate for our entire corpus.

The four vectors are stored, and later accessed, contextually: separate Markov tables are kept for each of the four beats of a measure, and for separate sections. Thus, all vectors that occur on the second beat are considered queries to continuations for the onsets that occur on the third beat; similarly, these same vectors are continuations for onsets that occur on the first beat. The continuations are stored over eight measure phrases, so the first beat of the second measure is a continuation for the fourth beat of the first measure. We have not found it necessary to move beyond first-order Markov generation, since our data involves four-items representing four onsets.

We found that the third method produced the most accurate re-creations of drum patterns found in the corpus, yet the first method produced the most surprising, while maintaining usability. Rather than selecting only a single method for drum pattern generation, it was decided that the three separate methods provided distinct "flavors", allowing users several degrees of separation from the original corpus. Therefore, all three methods were used in the generation of a large (>2000) database of potential patterns, from which actual patterns are contextually selected. See (Eigenfeldt and Pasquier 2013) for a complete description of our use of populations and the selection of patterns from these populations.

Auxiliary Percussion Generation

Auxiliary percussion consists of non-pitched rhythmic material not contained within the drum pattern. Within our corpus, we have extracted two separate auxiliary percussion parts, each with up to four subparts. The relationship between these parts to the drum pattern is intrinsic to the rhythmic drive of the music; however, there is no clear or consistent musical relationship between these parts, and thus no heuristic method available for their generation.

We have chosen to generate these parts through first-order Markov chains, using the same contextual beat-specific encoding just described; as such, logical horizontal relationships found in the corpus are maintained. Using the same 4-bit representation for each auxiliary percussion part as described in method 3 for drum pattern generation, vertical consistency is also imparted; however, the original relationship to the drum pattern is lost. Therefore, we constrain the available continuations.

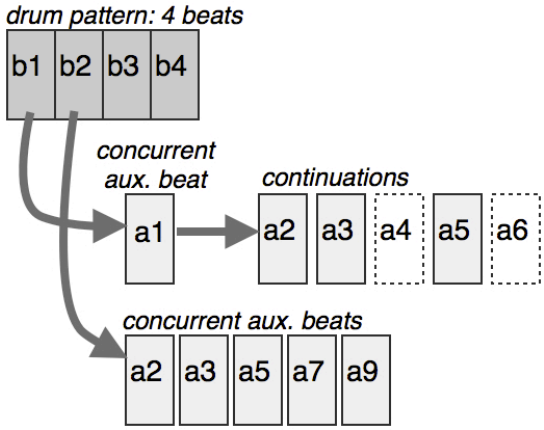


Figure 4. Maintaining contextual vertical and horizontal relationships between auxiliary percussion beats (*a*) and drum beats (*b*).

As the drum patterns are generated prior to the auxiliary percussion, the individual beats from these drum patterns serve as the query to a cross-referenced transition table made up of auxiliary percussion pattern beats (see Figure 4). Given a one measure drum pattern consisting of four beats *b1 b2 b3 b4*, all auxiliary percussion beats that occur simultaneously with *b1* in the corpus are considered as available concurrent beats for the auxiliary percussion pattern's initial beat. One of these, *a1*, is selected as the first beat, using a weighted probability selection. The available

[5] The generative music community uses the term "continuations" to refer to what is usually called transitions (weighted edges in the graph).

continuations for *a1* are *a2-a6*. Because the next auxiliary percussion beat must occur at the same time as the drum pattern's *b2*, the auxiliary percussion beats that occur concurrently with *b2* are retrieved: *a2, a3, a5, a7, a9*. Of these, only *a2, a3,* and *a5* intersect both sets; as such, the available continuations for *a1* are constrained, and the next auxiliary percussion beat is selected from *a2, a3,* and *a5*.

Of note is the fact that any selection from the constrained set will be horizontally correct due to the transition table, as well as being vertically consistent in its relationship to the drum pattern due to the constraints; however, since the selection is made randomly from the probabilistic distribution of continuations, the final generated auxiliary percussion pattern will not necessarily be a pattern found in the corpus.

Lastly, we have not experienced insufficient continuations since we are working with individual beats, rather than entire measures: while there are only a limited number of four-element combinations that can serve as queries, a high number of 1-beat continuations exist.

Bassline Generation

Human analysis determined there were up to two different basslines in the analysed tracks, not including bass drones, which are considered a synthesizer part. Bassline generation is a two-step process: determining onsets (which include held notes longer than the smallest quantized value of a sixteenth note); then overlaying pitches onto these onsets.

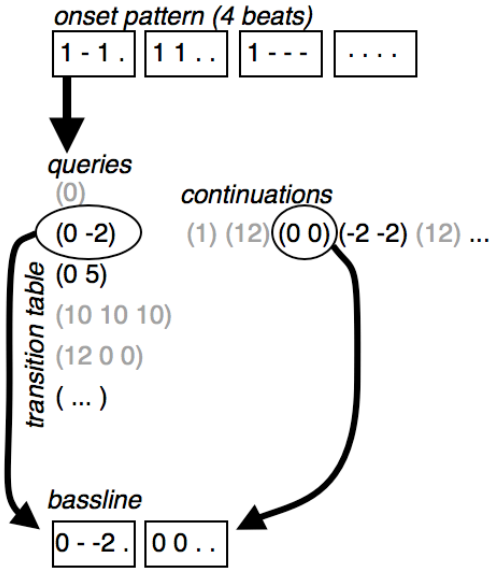


Figure 5. Overlaying pitch-classes onto onsets, with continuations constrained by the number of pitches required in the beat.

Bassline onset generation uses the same method as that of auxiliary percussion – contextually dependent Markov sequences, using the existing drum patterns as references. One Markov transition table encoded from the corpus' basslines contains rhythmic information: onsets (1), rests (.), and held notes (-). The second transition table contains only pitch data: pitch-classes relative to the track's key (-24 to +24). Like the auxiliary percussion transition tables, both the queries and the continuations are limited to a single beat.

Once a bassline onset pattern is generated, it is broken down beat by beat, with the number of onsets occurring within a given beat serving as the first constraint on pitch selection (see Figure 5). Our analysis derived 68 possible 1-beat pitch combinations within the "Breaks" corpus. In Figure 5, an initial beat contains 2 onsets (`1 – 1 .`) Within the transition table, 38 queries contain two values (not grayed out in Figure 5's vertical column): one of these is selected as the pitches for the first beat using a weighted probability selection (circled). As the next beat contains 2 onsets (`1 1 . .`), the first beat's pitches (`0 -2`) serve as the query to the transition table, and the returned continuations are constrained by matching the number of pitches required (not grayed out in Figure 5's horizontal row). One of these is selected for the second beat (circled) using additional constraints described in the next section. This process continues, with pitch-classes being substituted for onset flags (bottom).

Additional Bassline Constraints

Additional constraints are placed upon the bassline generation, based upon user set "targets". These include constraints the following:

- Density: favouring fewer or greater onsets per beat;
- straightness: favouring onsets on the beat versus syncopated;
- dryness: favouring held notes versus rests;
- jaggedness: favouring greater or lesser differentiation between consecutive pitch-classes.

Each available continuation is rated in comparison to the user-set targets using a Euclidean distance function, and an exponential random selection is made from the top 20% of these ranked continuations.

This notion of targets appears throughout the system. While such a method does allow some control over the generation, the main benefit will be demonstrated in the next stage of our research: successive generations of entire compositions – generating hour long sets of tracks, for example – can be guaranteed to be divergent by ensuring targets for parameters are different between runs.

Contextual Drum-fills

Fills, also known as drum-fills, drum-breaks, or simply breaks, occur at the end of eight measure phrases as variations of the overall repetitive pattern, and serve to highlight the end of the phrase, and the upcoming section change. Found in most popular music, they are often restricted to the drums, but can involve other instruments (such as auxiliary percussion), as well as a break, or silence, from the other parts.

Fills are an intuitive aspect of composition in pattern-based music, and can be conceptually reduced to a rhythmic variation. As such, they are not difficult to code algo-

rithmically: for example, following seven repetitions of a one measure drum pattern, a random shuffle of the pattern will produce a perfectly acceptable fill for the eighth measure (see Figure 6).

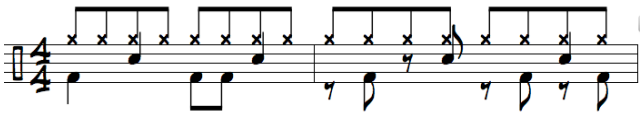


Figure 6. Left: drum pattern for kick, snare, and hihat; right: pattern variation by shuffling onsets can serve as a fill.

Rather than utilizing such creative "shortcuts", our fill generation is based entirely upon the corpus. First, the *location* of the fill is statistically generated based upon the location of fills within phrases in the corpus, and the generated phrase structure. Secondly, the *type* of fill is statistically generated based upon the analysis: for example, the described pattern variation using a simple onset shuffle has a 0.48 probability of occurring within the Breaks corpus – easily the most common fill type. Lastly, the actual variation is based upon the specific *context*.

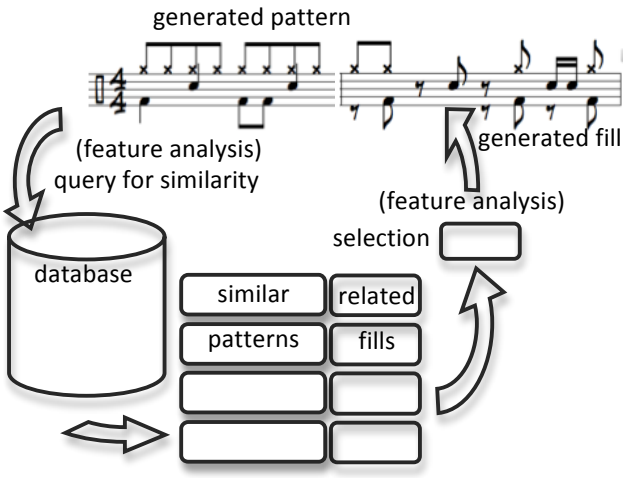


Figure 7. Fill generation, based upon contextual similarity

Fills always replace an existing pattern; however, the actual pattern to be replaced within the generated drum part may not be present in the corpus, and thus no direct link would be evident from a fill corpus. As such, the original pattern is analysed for various features, including *density* (the number of onsets) and *syncopation* (the percentile of onsets that are not on strong beats). These values are then used to search the corpus for patterns with similar features. One pattern is selected from those that most closely match the query. The relationship between the database's pattern and its fill is then analysed for *consistency* (how many onsets remain constant), *density change* (how many onsets are added or removed), and *syncopation change* (the percentile change in the number of onsets that are not on strong beats). This data is then used to generate a variation on the initial pattern (see Figure 7).

The resulting fill will display a relationship to its original pattern in a contextually similar relationship to the corpus.

Conclusions and Future Work

The musical success of EDM lies in the interrelationship of its parts, rather than the complexity of any individual part. In order to successfully generate a complete musical work that is representative of the model, rather than generating only components of the model (i.e. a single drum pattern), we have taken into account both horizontal relationships between elements in our use of a Markov model, as well as vertical relationships in our use of constraint-based algorithms. Three different methods to model these horizontal and vertical dependencies at generation time have been proposed in regards to drum pattern generation (through the use of a combined representation of kick, snare, open and closed hihat, as well as context-dependent Markov selection), auxiliary percussion generation (through the use of constrained Markov transitions) and bassline generation (through the use of both onset- and pitch-constrained Markov transitions. Each of these decisions contributes to what we believe to be a more successful generation of a complete work that is stylistically representative and consistent.

Future work includes validation to investigate our research objectively. We have submitted our work to EDM festivals and events that specialize in algorithmic dance music, and our generated tracks have been selected for presentation at two festivals so far. We also plan to produce our own dance event, in which generated EDM will be presented alongside the original corpus, and use various methods of polling the audience to determine the success of the music.

Lastly, we plan to continue research in areas not discussed in this paper, specifically autonomous timbral selection and signal processing, both of which are integral to the success of EDM.

This research was created in MaxMSP and Max4Live running in Ableton Live. Example generations can be heard at soundcloud.com/loadbang.

Acknowledgements

This research was funded by a grant from the Canada Council for the Arts, and the Natural Sciences and Engineering Research Council of Canada.

References

Ames, C. 1989. *The Markov Process as a Compositional Model: A Survey and Tutorial.* Leonardo 22(2).

Chadabe, J. 1980. *Solo: A Specific Example of Realtime Performance.* Computer Music - Report on an International Project. Canadian Commission for UNESCO.

Chadabe, J. 1984. *Interactive Composing.* Computer Music Journal 8:1.

Chuang, J. 1995. Mozart's Musikalisches Würfelspiel, http://sunsite.univie.ac.at/Mozart/dice/, retrieved September 10, 2012.

Collins, N. 2001. *Algorithmic composition methods for breakbeat science*. Proceedings of Music Without Walls, De Montfort University, Leicester, 21-23.

Collins, N. 2008. *Infno: Generating synth pop and electronic dance music on demand.* Proceedings of the International Computer Music Conference, Belfast.

Collins, N. 2010. *Computational analysis of musical influence: A musicological case study using MIR tools.* Proceedings of the International Symposium on Music Information Retrieval, Utrecht.

Collins, T. 2011. *Improved methods for pattern discovery in music, with applications in automated stylistic composition.* PhD thesis, Faculty of Mathematics, Computing and Technology, The Open University.

Cope, D. 1992. *Computer Modeling of Musical Intelligence in EMI.* Computer Music Journal, 16:2, 69–83.

Diakopoulos, D., Vallis, O., Hochenbaum, J., Murphy, J., and Kapur, A. 2009. 21st Century Electronica: MIR Techniques for Classification and Performance. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Kobe, 465–469.

Dubnov, S., Assayag, G., Lartillot, O. and Bejerano, G. 2003. *Using machine-learning methods for musical style modeling*. Computer, 36:10.

Eigenfeldt, A. 2007. *Computer Improvisation or Real-time Composition: A Composer's Search for Intelligent Tools.* Electroacoustic Music Studies Conference 2007, http://www.ems-network.org/spip.php?article264 Accessed 3 February 2013.

Eigenfeldt, A. 2012 *Embracing the Bias of the Machine: Exploring Non-Human Fitness Functions*. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Palo Alto.

Eigenfeldt, A. 2013. *Is Machine Learning the Next Step in Generative Music?* Leonardo Electronic Almanac, Special Issue on Generative Art, forthcoming.

Eigenfeldt, A., and Pasquier, P. 2011. *Towards a Generative Electronica: Human-Informed Machine Transcription and Analysis in MaxMSP*. Proceedings of the Sound and Music Computing Conference, Padua.

Eigenfeldt, A., and Pasquier, P. 2012. *Populations of Populations - Composing with Multiple Evolutionary Algorithms*. P. Machado, J. Romero, and A. Carballal (Eds.). In: EvoMUSART 2012, LNCS 7247, 72–83. Springer, Heidelberg.

Eigenfeldt, A., Pasquier, P., and Burnett, A. 2012. *Evaluating Musical Metacreation.* International Conference of Computational Creativity, Dublin, 140–144.

Eigenfeldt, A., and Pasquier, P. 2013. *Evolving Structures in Electronic Dance Music,* GECCO 2013, Amsterdam.

Hiller, L., and Isaacson, L. 1979. *Experimental Music; Composition with an Electronic Computer.* Greenwood Publishing Group Inc. Westport, CT, USA.

Kaliakatsos-Papakostas, M., Floros, A., and Vrahatis, M.N. 2013. EvoDrummer: Deriving rhythmic patterns through interactive genetic algorithms. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design. Lecture Notes in Computer Science Volume 7834, 2013, pp 25–36.

Lerdahl, F., Jackendoff, R. 1983. *A generative theory of tonal music.* The MIT Press.

Lewis, G. 1999. *Interacting with latter-day musical automata.* Contemporary Music Review, 18:3.

Pachet, F., Roy, P., and Barbieri, G. 2011. *Finite-length Markov processes with constraints.* Proceedings of the Twenty-Second international joint conference on Artificial Intelligence Volume One. AAAI Press.

Papadopoulos, G., and Wiggins, G. 1999. *AI methods for algorithmic composition: A survey, a critical view and future prospects.* In: AISB Symposium on Musical Creativity, 110–117, Edinburgh, UK.

Randel, D. 2003. *The Harvard Dictionary of Music.* Belknap Press.

Rowe, R. 1993. *Interactive Music Systems*. Cambridge, Mass., MIT Press.

Waschka, R. 2007. *Composing with Genetic Algorithms: GenDash.* Evolutionary Computer Music, Springer, London, 117–136.

Harmonising Melodies: Why Do We Add the Bass Line First?

Raymond Whorley and **Christophe Rhodes**
Department of Computing
Goldsmiths, University of London
New Cross, London, SE14 6NW, UK
{r.whorley, c.rhodes}@gold.ac.uk

Geraint Wiggins and **Marcus Pearce**
School of Electronic Engineering and Computer Science
Queen Mary, University of London
Mile End Road, London, E1 4NS, UK
{geraint.wiggins, marcus.pearce}@eecs.qmul.ac.uk

Abstract

We are taking an information theoretic approach to the question of the best way to harmonise melodies. Is it best to add the bass first, as has been traditionally the case? We describe software which uses statistical machine learning techniques to learn how to harmonise from a corpus of existing music. The software is able to perform the harmonisation task in various different ways. A performance comparison using the information theoretic measure *cross-entropy* is able to show that, indeed, the bass first approach appears to be best. We then use this overall strategy to investigate the performance of specialist models for the prediction of different musical attributes (such as pitch and note length) compared with single models which predict all attributes. We find that the use of specialist models affords a definite performance advantage. Final comparisons with a simpler model show that each has its pros and cons. Some harmonisations are presented which have been generated by some of the better performing models.

Introduction

In our ongoing research, we are developing computational models of four-part harmony such that alto, tenor and bass parts are added to a given soprano part in a stylistically suitable way. In this paper we compare different strategies for carrying out this creative task. In textbooks on four-part harmony, students are often encouraged to harmonise a melody in stages. In particular, it is usual for the bass line to be added first, with harmonic symbols such as Vb (dominant, first inversion) written underneath. The harmony is then completed by filling in the inner (alto and tenor) parts. This paper sets out to show what information theory has to say about the best way to approach harmonisation. Is adding the bass line first optimal, or is there a better approach?

In order to investigate questions such as this, we have written software based on *multiple viewpoint systems* (Conklin and Witten 1995) which enables the computer to learn for itself how to harmonise by building a statistical model from a corpus of existing music. The multiple viewpoint framework allows different attributes of music to be modelled. The predictions of these individual models are then combined to give an overall prediction. The multiple viewpoint systems are selected automatically, on the basis of minimising the information theoretic measure *cross-entropy*. We have developed and implemented three increasingly complex versions of the framework, which allow models to be constructed in different ways. The first two versions are particularly pertinent to the aims of this paper, since they facilitate precisely the comparisons we wish to make without the time complexity drawbacks of the more complex version 3. The latter is therefore not utilised in this part of our research.

The fact that the resulting models are statistical (and indeed self-learned from a corpus) means that harmonies are generated in a non-deterministic way. The harmonies are more or less probable, rather than right or wrong, with an astronomical number of ways for a melody to be harmonised from the probability distributions. Of course, there is little point in producing something novel if it is also deemed to be bad. Our aim is to hone the models in such a way that the subjective quality and style of the generated harmony is consistently similar to that of the corpus, whilst retaining almost infinite variety. In this way, the computational models can be thought of as creative in much the same way as a human composer (or at the very least that they imitate such creativity). Finding a good overall strategy for carrying out the harmonisation task is an important part of this improvement process.

Multiple Viewpoint Systems

There follows a brief description of some essential elements of multiple viewpoint systems. In order to keep things simple we look at things from the point of view of melodic modelling (except for the subsection entitled Cross-entropy and Evaluation).

Types of Viewpoint

Basic viewpoints are the fundamental musical attributes that are predicted, such as `Pitch` and `Duration`. The *domain* (or alphabet) of `Pitch` is the set of MIDI values of notes seen in the melodies comprising the corpus. A semibreve (or whole note) is divided into 96 `Duration` units; therefore the domain of `Duration` is the set of integer values representing note lengths seen in the corpus.

Derived viewpoints such as `Interval` (sequential pitch interval) and `DurRatio` (sequential duration ratio) are derived from, and can therefore predict, basic types (in this case `Pitch` and `Duration` respectively). A B4

(MIDI value 71) following a G4 (MIDI value 67) has an `Interval` value of 4. Descending intervals have negative values. Similarly, a minim (half note) following a crotchet (quarter note) has a `DurRatio` value of 2.

Threaded viewpoints are defined only at certain positions in a sequence, determined by Boolean test viewpoints such as `Tactus`; for example, `Pitch` $\ominus$ `Tactus` has a defined `Pitch` value only on `Tactus` beats (*i.e.*, the main beats in a bar).

A *linked* viewpoint is the conjunction of two or more simple (or *primitive*) viewpoints; for example, `DurRatio` $\otimes$ `Interval` is able to predict both `Duration` and `Pitch`. If any of the constituent viewpoints are undefined, then the linked viewpoint is also undefined. These are just a few of the viewpoints we have implemented. See Conklin and Witten (1995) for more information about viewpoints.

N-gram Models

So far, *N-gram* models, which are Markov models employing subsequences of N symbols, have been the modelling method of choice when using multiple viewpoint systems. The probability of the N^{th} symbol, the *prediction*, depends only upon the previous $N-1$ symbols, the *context*. The number of symbols in the context is the *order* of the model. Only defined viewpoint values are used in N-grams; sequence positions with an undefined viewpoint value are skipped. See Manning and Schütze (1999) for more details.

Modelling Viewpoints

What we call a *viewpoint model* is a weighted combination of various orders of N-gram model of a particular viewpoint. The combination is achieved by *Prediction by Partial Match* (Cleary and Witten 1984). PPM makes use of a sequence of models, which we call a *back-off sequence*, for context matching and the construction of complete prediction probability distributions. The back-off sequence begins with the highest order model, proceeds to the second-highest order, and so on. An *escape method* (in this research, method C) determines prediction probabilities, which are generally high for predictions appearing in high-order models, and *vice versa*. If necessary, a probability distribution is completed by backing off to a uniform distribution.

Combining Viewpoint Models

A multiple viewpoint system comprises more than one viewpoint; indeed, usually many more. The prediction probability distributions of the individual viewpoint models must be combined. The first step is to convert these distributions into distributions over the domain of whichever basic type is being predicted at the time. A weighted arithmetic or geometric (Pearce, Conklin, and Wiggins 2004) combination technique is then employed to create a single distribution. A run-time parameter called a *bias* affects the weighting. See Conklin (1990) for more information.

Long-term and Short-term Models

Conklin (1990) introduced the idea of using a combination of a *long-term model* (LTM), which is a general model of a style derived from a corpus, and a *short-term model* (STM), which is constructed as a piece of music is being predicted or generated. The latter aims to capture musical structure peculiar to that piece. Currently, the same multiple viewpoint system is used for each. The LTM and STM distributions are combined in the same way as the viewpoint distributions, for which purpose there is a separate bias (L-S bias).

Cross-entropy and Evaluation

Cross-entropy is used to objectively compare the prediction performance of different models. If we define $P_m(S_i|C_{i,m})$ as the probability of the i^{th} musical symbol given its context for a particular model m, and assume that there are a total of n sequential symbols, then cross-entropy is given by $-(1/n)\sum_{i=1}^{n}\log_2 P_m(S_i|C_{i,m})$. Jurafsky and Martin (2000) note that because the cross-entropy of a sequence of symbols (according to some model) is always higher than its true entropy, the most accurate model (*i.e.*, the one closest to the true entropy) must be the one with the lowest cross-entropy. In addition, because it is a "per symbol" measure, it is possible to similarly compare generated harmonisations of any length. Harmonisations with a low cross-entropy are likely to be simpler and more predictable to a listener, while those with a high cross-entropy are likely to be more complex, more surprising and in the extreme possibly unpleasant. See Manning and Schütze (1999) for more details on cross-entropy.

Model Construction

Cross-entropy is also used to guide the automatic construction of multiple viewpoint systems. Viewpoints are added (and sometimes removed) from a system stage by stage. Each candidate system is used to calculate the average cross-entropy of a ten-fold cross-validation of the corpus. The system producing the lowest cross-entropy goes on to the next stage of the selection process. For example, starting with the basic system {`Duration`, `Pitch`}, of all the viewpoints tried let us assume that `ScaleDegree` lowers the cross-entropy most on its addition. Our system now becomes {`Duration`, `Pitch`, `ScaleDegree`}. `Duration` cannot be removed at this stage, as a `Duration`-predicting viewpoint must be present. Assuming that on removing `Pitch` the cross-entropy rises, `Pitch` is also retained. Let us now assume that after a second round of addition we have the system {`Duration`, `Pitch`, `ScaleDegree`, `Interval`}. Trying all possible deletions, we may now find that the cross-entropy decreases on the removal of `Pitch`, giving us the system {`Duration`, `ScaleDegree`, `Interval`}. The process continues until no addition can be found to lower the cross-entropy by a predetermined minimum amount. When selection is complete, the biases are optimised.

Development of Multiple Viewpoints

The modelling of melody is relatively straightforward, in that a melody comprises a single sequence of non-overlapping notes. Such a sequence is ideal for creating N-grams. Harmony is much more complex, however. Not

only does it consist (for our purposes) of four interrelated parts, but it usually contains overlapping notes. In other words, music is usually not homophonic; indeed, very few of the major key hymn tune harmonisations (Vaughan Williams 1933) in our corpora are completely homophonic. Some preprocessing of the music is necessary, therefore, to make it amenable to modelling by means of N-grams. We use *full expansion* on our corpora (corpus 'A' and corpus 'B' each contain fifty harmonisations), which splits notes where necessary to achieve a sequence of block chords (*i.e.*, without any overlapping notes). This technique has been used before in relation to viewpoint modelling (Conklin 2002). To model harmony correctly, however, we must know which notes have been split. Basic viewpoint `Cont` is therefore introduced to distinguish between notes which are freshly sounded and those which are a continuation of the preceding one. Currently, the basic viewpoints (or attributes) are predicted at each point in the sequence in the following order: `Duration`, `Cont` and then `Pitch`.

Version 1

The starting point for the definition of the strictest possible application of viewpoints is the formation of vertical viewpoint elements (Conklin 2002). An example of such an element is $\langle 69, 64, 61, 57 \rangle$, where all of the values are from the domain of the same viewpoint (*i.e.*, `Pitch`, as MIDI values), and all of the parts (SATB) are represented. This method reduces the entire set of parallel sequences to a single sequence, thus allowing an unchanged application of the multiple viewpoint framework, including its use of PPM. Only those elements containing the given soprano note are allowed in the prediction probability distribution, however. This is the base-level model, to be developed with the aim of substantially improving performance.

Version 2

In this version, it is hypothesised that predicting all unknown symbols in a vertical viewpoint element at the same time is neither necessary nor desirable. It is anticipated that by dividing the overall harmonisation task into a number of subtasks (Allan and Williams 2005; Hild, Feulner, and Menzel 1992), each modelled by its own multiple viewpoint system, an increase in performance can be achieved. Here, a subtask is the prediction or generation of at least one part; for example, given a soprano line, the first subtask might be to predict the entire bass line. This version allows us to experiment with different arrangements of subtasks. As in version 1, vertical viewpoint elements are restricted to using the same viewpoint for each part. The difference is that not all of the parts are now necessarily represented in a vertical viewpoint element.

Comparison of Subtask Combinations

In this section we carry out the prediction of bass given soprano, alto/tenor given soprano/bass, tenor given soprano, alto/bass given soprano/tenor, alto given soprano, and tenor/bass given soprano/alto (*i.e.*, prediction in two stages), in order to ascertain the best performing combination for subsequent comparisons. Prediction in three stages is not considered here because of time limitations.

Earlier studies in the realm of melodic modelling revealed that the model which performed best was an LTM updated after every prediction in conjunction with an STM (a *BOTH+* model) using weighted geometric distribution combination. Time constraints dictate the assumption that such a model is likely to perform similarly well with respect to the modelling of harmony. In addition, only corpus 'A', a bias of 2 and an L-S bias of 14 are used for viewpoint selection (as for the best melodic BOTH+ runs using corpus 'A'). As usual, the biases are optimised after completion of selection. Here, we predict `Duration`, `Cont` and `Pitch` together (*i.e.*, using a single multiple viewpoint system at each prediction stage). We also use the *seen* `Pitch` domain at this juncture (*i.e.*, the domain of `Pitch` vertical viewpoint elements seen in the corpus, as opposed to all possible such elements).

It is appropriate at this point to make some general observations about the bar charts presented in this paper. Comparisons are made for a range of $\hbar$ (maximum N-gram order) from 0 to 5. Each value of $\hbar$ may have a different automatically selected multiple viewpoint system. Please note that all bar charts have a cross-entropy range of 2.5 bits/prediction, often not starting at zero. All bars have standard errors associated with them, calculated from the cross-entropies obtained during ten-fold cross-validation (using final multiple viewpoint systems and optimised biases).

Figure 1 compares the prediction of alto given soprano, tenor given soprano, and bass given soprano. The first thing to notice is that the error bars overlap. This could be taken to mean that we cannot (or should not) draw conclusions in such cases; however, the degree of overlap and the consistency of the changes across the range of $\hbar$ is highly suggestive of the differences being real. A clinching quantitative argument is reserved until consideration of Figure 3. Prediction of the alto part has the lowest cross-entropy and prediction of the bass has the highest across the board. This is very likely to be due to the relative number of elements in the `Pitch` domains for the individual parts (*i.e.*, 18, 20 and 23 for alto, tenor and bass respectively). The lowest cross-entropies occur at an $\hbar$ of 1 except for the bass, which has its minimum at an $\hbar$ of 2 (this cross-entropy is only very slightly lower than that for an $\hbar$ of 1, however).

There is a completely different picture for the final stage of prediction. Figure 2 shows that, having predicted the alto part with a low cross-entropy, the prediction of tenor/bass has the highest. Similarly, the high cross-entropy for the prediction of the bass is complemented by an exceptionally low cross-entropy for the prediction of alto/tenor (notice that the error bars do not overlap with those of the other prediction combinations). Once again, this can be explained by the number of elements in the part domains: the sizes of the cross-product domains are 460, 414 and 360 for tenor/bass, alto/bass and alto/tenor respectively. Although we are not using cross-product domains, it is likely that the seen domains are in similar proportion. The lowest cross-entropies occur at an $\hbar$ of 1.

Combining the two stages of prediction, we see in Fig-

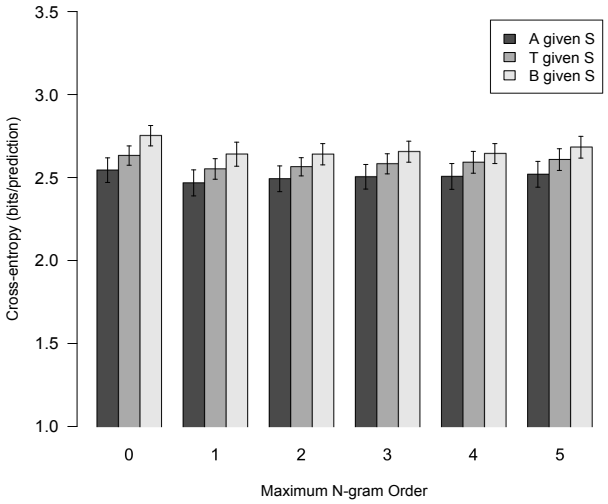


Figure 1: Bar chart showing how cross-entropy varies with $\hbar$ for the version 2 prediction of alto given soprano, tenor given soprano, and bass given soprano using the seen `Pitch` domain. `Duration`, `Cont` and `Pitch` are predicted using a single multiple viewpoint system at each prediction stage.

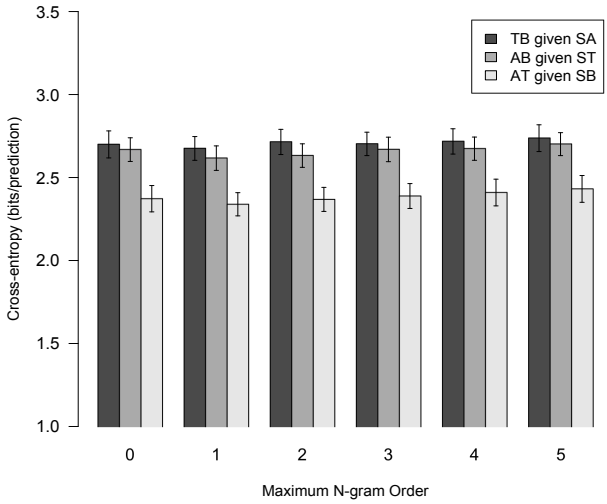


Figure 2: Bar chart showing how cross-entropy varies with $\hbar$ for the version 2 prediction of tenor/bass given soprano/alto, alto/bass given soprano/tenor and alto/tenor given soprano/bass using the seen `Pitch` domain. `Duration`, `Cont` and `Pitch` are predicted using a single multiple viewpoint system at each prediction stage.

ure 3 that predicting bass first and then alto/tenor has the lowest cross-entropy. Notice, however, that the error bars of this model overlap with those of the other models. This is a critical comparison, requiring a high degree of confidence in the conclusions we are drawing. Let us look at the $\hbar = 1$ and $\hbar = 2$ comparisons in more detail, as they are particularly pertinent. In both cases, all ten cross-entropies produced by ten-fold cross-validation are lower for B then AT than for A then TB; and nine out of ten are lower for B then AT than for T then AB. The single increase is 0.11 bits/chord for an $\hbar$ of 1 and 0.09 bits/chord for an $\hbar$ of 2 compared with a mean decrease of 0.22 bits/chord for the other nine values in each case. This demonstrates that we can have far greater confidence in the comparisons than the error bars might suggest. A likely reason for this is that there is a range of harmonic complexity across the pieces in the corpus which is reflected as a range of cross-entropies (ultimately due to compositional choices). This inherent cross-entropy variation seems to be greater than the true statistical variation applicable to these comparisons.

We can be confident, then, that predicting bass first and then alto/tenor is best, reflecting the usual human approach to harmonisation. The lowest cross-entropy is 4.98 bits/chord, occurring at an $\hbar$ of 1. Although having the same cross-entropy to two decimal places, the very best model combines the bass-predicting model using an $\hbar$ of 2 (optimised bias and L-S bias are 1.9 and 53.2 respectively) with the alto/tenor-predicting model using an $\hbar$ of 1 (optimised bias and L-S bias are 1.3 and 99.6 respectively).

Table 1 gives some idea of the complexity of the multiple viewpoint systems involved, listing as it does the first six viewpoints automatically selected for the prediction of bass given soprano ($\hbar = 2$) and alto/tenor given soprano/bass ($\hbar = 1$). Many of the primitive viewpoints involved have already been defined or are intuitively obvious. `LastInPhrase` and `FirstInPiece` are either true of false, and `Piece` has three values: first in piece, last in piece or otherwise. `Metre` is more complicated, being an attempt to define metrical equivalence within and between bars of various time signatures. Notice that only two of the viewpoints are common to both systems. In fact, of the twenty-four viewpoints in the B given S system and twelve in the AT given SB system, only five are common. This demonstrates the degree to which the systems have specialised in order to carry out these rather different tasks. The difference in the size of the systems suggests that the prediction of the bass part is more complicated than that of the inner parts, as reflected in the difference in cross-entropy.

The Effect of Model Order

Figure 1 indicates that, for example, there is only a small reduction in cross-entropy from $\hbar = 0$ to $\hbar = 1$. The degree of error bar overlap means that even this small reduction is questionable. Is it possible that there is no real difference in performance between a model using unconditional probabilities and one using the shortest of contexts? Let us, in the first place, examine the individual ten-fold cross-validation cross-entropy values. All ten of these values are lower for an $\hbar$ of 1, giving us confidence that there is indeed a small improvement. Having established that, however, it would be useful to explain why the improvement is perhaps smaller than we might have expected.

One important reason for the less than impressive improvement is that although the $\hbar = 0$ model is nominally unconditional, the viewpoints `Interval`, `DurRatio` and `Interval` $\ominus$ `Tactus` appear in the $\hbar = 0$ multiple view-

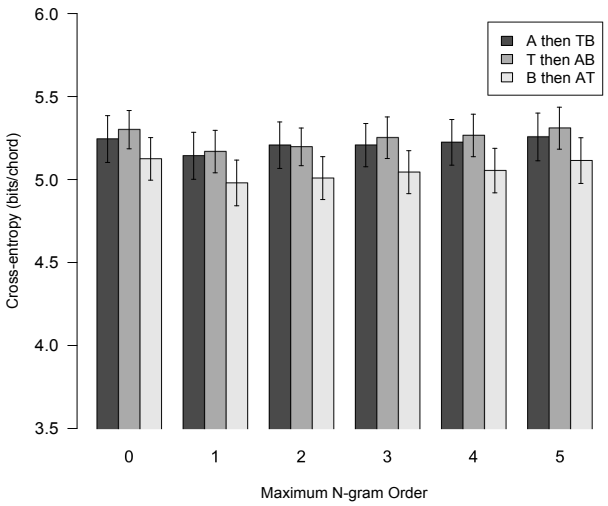


Figure 3: Bar chart showing how cross-entropy varies with $\hbar$ for the version 2 prediction of alto then tenor/bass, tenor then alto/bass and bass then alto/tenor given soprano using the seen `Pitch` domain. `Duration`, `Cont` and `Pitch` are predicted using a single multiple viewpoint system at each prediction stage.

point system (linked with other viewpoints). These three viewpoints make use of attributes of the preceding chord; therefore with respect to predicted attributes `Duration` and `Pitch`, this model is partially $\hbar = 1$. This hidden conditionality is certainly enough to substantially improve performance compared with a completely unconditional model.

Another reason is quite simply that the corpus has failed to provide sufficient conditional statistics; in other words, the corpus is too small. This is the fundamental reason for the performance dropping off above an $\hbar$ of 1 or 2. We would expect peak performance to shift to higher values of $\hbar$ as the quantity of statistics substantially increases. Supporting evidence for this is provided by our modelling of melody. Much better melodic statistics can be gathered from the same corpus because the `Pitch` domain is very much smaller than it is for harmony. A BOTH+ model shows a large fall in cross-entropy from $\hbar = 0$ to $\hbar = 1$ (with error bars not overlapping), while peak performance occurs at an $\hbar$ of 3.

Viewpoint	B	AT
`Pitch`	×	
`Interval` ⊗ `InScale`	×	
`Cont` ⊗ `TactusPositionInBar`	×	×
`Duration` ⊗ (`ScaleDegree` ⊖ `LastInPhrase`)	×	×
`Interval` ⊗ (`ScaleDegree` ⊖ `Tactus`)	×	
`ScaleDegree` ⊗ `Piece`	×	
`Cont` ⊗ `Interval`		×
`DurRatio` ⊗ `TactusPositionInBar`		×
`ScaleDegree` ⊗ `FirstInPiece`		×
`Cont` ⊗ `Metre`		×

Table 1: List of the first six viewpoints automatically selected for the prediction of bass given soprano (B, $\hbar = 2$) and alto/tenor given soprano/bass (AT, $\hbar = 1$).

Figure 2 reveals an even worse situation with respect to performance differences across the range of $\hbar$. For TB given SA, for example, it is not clear that there is a real improvement from $\hbar = 0$ to $\hbar = 1$. In this case, there is a reduction in five of the ten-fold cross-validation cross-entropy values, but an increase in the other five. This is almost certainly due to the fact that, having fixed the soprano and alto notes, the number of tenor/bass options are severely limited; so much so, that conditional probabilities can rarely be found. This situation should also improve with increasing corpus size.

Separate Prediction of Attributes

We now investigate the use of separately selected and optimised multiple viewpoint systems for the prediction of `Duration`, `Cont` and `Pitch`. Firstly, however, let us consider the utility of creating an *augmented* `Pitch` domain. Approximately 400 vertical `Pitch` elements appear in corpus 'B' which are not present in corpus 'A', and there are undoubtedly many more perfectly good chords which are absent from both corpora. Such chords are unavailable for use when the models generate harmony, and their absence must surely skew probability distributions when predicting existing data. One solution is to use a full Cartesian product; but this is known to result in excessively long run times. Our preferred solution is to transpose chords seen in the corpus up and down, a semitone at a time, until one of the parts goes out of the range seen in the data. Such elements not previously seen are added to the augmented `Pitch` domain. Derived viewpoints such as `ScaleDegree` are able to make use of the extra elements. We shall see shortly that this change increases cross-entropies dramatically; but since this is not a like-for-like comparison, it is not an indication of an inferior model.

Figure 4 shows that better models can be created by selecting separate multiple viewpoint systems to predict individual attributes, rather than a single system to predict all of them. The difference in cross-entropy is quite marked, although there is a substantial error bar overlap. An $\hbar$ of 1 is optimal in both cases. All ten cross-entropies produced by ten-fold cross-validation are lower for the separate system case, providing confidence that the improvement is real. The lowest cross-entropy for separate prediction at $\hbar = 1$ is 5.44 bits/chord, compared with 5.62 bits/chord for prediction together. The very best model for separate prediction, with a cross-entropy of 5.35 bits/chord, comprises the best performing systems of whatever the value of $\hbar$.

Comparison of Version 1 with Version 2

A comparison involving `Duration`, `Cont` and `Pitch` would show that version 2 has a substantially higher cross-entropy than version 1. This is due to the fact that whereas the duration of an entire chord is predicted only once in version 1, it is effectively predicted twice (or even three times)

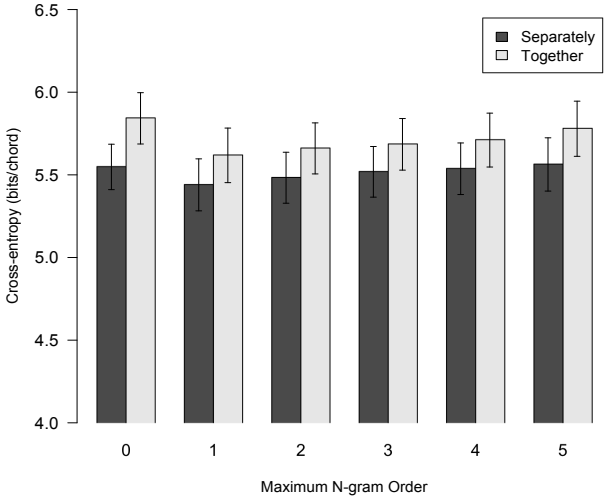


Figure 4: Bar chart showing how cross-entropy varies with $\hbar$ for the version 2 prediction of bass given soprano followed by alto/tenor given soprano/bass using the augmented `Pitch` domain. The prediction of `Duration`, `Cont` and `Pitch` separately (*i.e.*, using separately selected multiple viewpoint systems) and together (*i.e.*, using a single multiple viewpoint system) are compared.

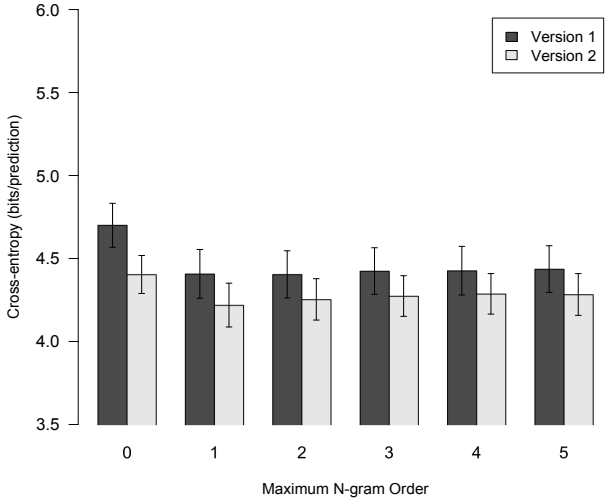


Figure 5: Bar chart showing how cross-entropy varies with $\hbar$ for the separate prediction of `Cont` and `Pitch` in the alto, tenor and bass given soprano using the augmented `Pitch` domain, comparing version 1 with version 2.

in version 2. Prediction of `Duration` is set up such that, for example, a minim may be generated in the bass given soprano generation stage, followed by a crotchet in the final generation stage, whereby the whole of the chord becomes a crotchet. This is different from the prediction and generation of `Cont` and `Pitch`, where elements generated in the first stage are not subject to change in the second. The way in which the prediction of `Duration` is treated, then, means that versions 1 and 2 are not directly comparable with respect to that attribute.

By ignoring `Duration` prediction, and combining only the directly comparable `Cont` and `Pitch` cross-entropies, we can make a judgement on the overall relative performance of these two versions. Figure 5 is strongly indicative of version 2 performing better than version 1. Again, there is an error bar overlap; but for an $\hbar$ of 1, nine out of ten cross-entropies produced by ten-fold cross-validation are lower for version 2; and for an $\hbar$ of 2, eight out of ten are lower for version 2. The single increase for an $\hbar$ of 1 is 0.07 bits/chord, compared with a mean decrease of 0.22 bits/chord for the other nine values. The mean of the two increased values for an $\hbar$ of 2 is 0.03 bits/chord, compared with a mean decrease of 0.20 bits/chord for the other eight values.

As one might expect from experience of harmonisation, predicting the bass first followed by the alto and tenor is better than predicting all of the lower parts at the same time. It would appear that the selection of specialist multiple viewpoint systems for the prediction of different parts is beneficial in rather the same way as specialist systems for the prediction of the various attributes. The optimal version 2 cross-entropy, using the best subtask models irrespective of the value of $\hbar$, is 0.19 bits/prediction lower than that of version 1.

Finally, the systems selected using corpus 'A' are used in conjunction with corpus 'A+B'. Compared with Figure 5, Figure 6 shows a much larger drop in cross-entropy for version 1 than for version 2: indeed, the bar chart shows the minimum cross-entropies to be exactly the same. Allowing for a true variation smaller than that suggested by the error bars, as before, we can certainly say that the minimum cross-entropies are approximately the same. The only saving grace for version 2 is that the error bars are slightly smaller. We can infer from this that version 1 creates more general models, better able to scale up to larger corpora which may deviate somewhat from the characteristics of the original corpus. Conversely, version 2 is capable of constructing models which are more specific to the corpus for which they are selected. This hypothesis can easily be tested by carrying out viewpoint selection in conjunction with corpus 'A+B' (although this would be a very time-consuming process).

Notice that there are larger reductions in cross-entropy from $\hbar = 0$ to $\hbar = 1$ in Figure 6 than in Figure 5. The only difference between the two sets of runs is the corpus used; therefore this performance change must be due to the increased quantity of statistics gathered from a larger corpus, as predicted earlier in the paper.

Generated Harmony

Generation is achieved simply by random sampling of overall prediction probability distributions. Each prediction probability has its place in the total probability mass; for example, attribute value X having a probability of 0.4 could be positioned in the range 0.5 to 0.9. A random number from 0 to 1 is generated, and if this number happens to fall between 0.5 and 0.9 then X is generated.

It was quickly very obvious, judging by the subjective quality of generated harmonisations, that a modification

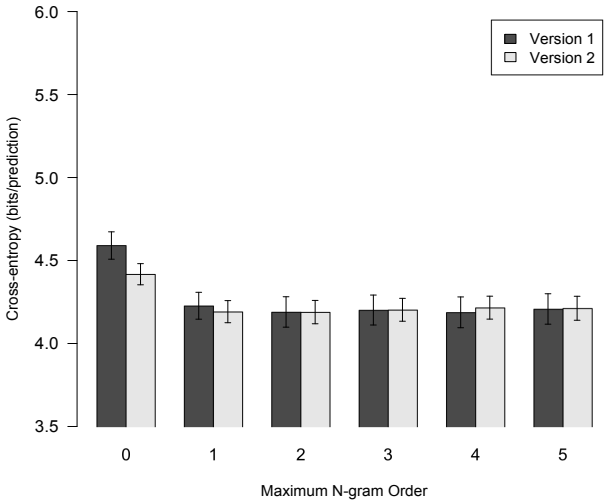


Figure 6: Bar chart showing how cross-entropy varies with $\hbar$ for the separate prediction of `Cont` and `Pitch` in the alto, tenor and bass given soprano using the augmented `Pitch` domain and corpus 'A+B' with systems selected using corpus 'A', comparing versions 1 and 2.

to the generation procedure would be required to produce something coherent and amenable to comparison. The problem was that random sampling sometimes generated a chord of very low probability, which was bad in itself because it was likely to be inappropriate in its context; but also bad because it then formed part of the next chord's context, which had probably rarely or never been seen in the corpus. This led to the generation of more low probability chords, resulting in harmonisations of much higher cross-entropy than those typically found in the corpus (quantitative evidence supporting the subjective assessment). The solution was to disallow the use of predictions below a chosen value, the *probability threshold*, defined as a fraction of the highest prediction probability in a given distribution. This definition ensures that there is always at least one usable prediction in the distribution, however high the fraction (*probability threshold parameter*). Bearing in mind that an expert musician faced with the task of harmonising a melody would consider only a limited number of the more likely options for each chord position, the removal of low probability predictions was considered to be a reasonable solution to the problem. Separate thresholds have been implemented for `Duration`, `Cont` and `Pitch`, and these thresholds may be different for different stages of generation. It is hoped that as the models improve, the thresholds can be reduced.

The probability thresholds of models used for generating harmony are optimised such that the cross-entropy of each subtask, averaged across twenty harmony generation runs using the ten melodies from test dataset 'A+B', approximately matches the corresponding prediction cross-entropy obtained by ten-fold cross-validation of corpus 'A+B'.

One of the more successful harmonisations of hymn tune *Das walt' Gott Vater* (Vaughan Williams 1933, hymn no. 36), automatically generated by the best version 1 model with optimised probability threshold parameters, is shown in Figure 7. It is far from perfect, with the second phrase being particularly uncharacteristic of the corpus. There are two parallel fifths in the second bar and another at the beginning of the fourth bar. The bass line is not very smooth, due to the many large ascending and descending leaps.

One of the more successful harmonisations of the same hymn tune, automatically generated by the best version 2 model with optimised probability threshold parameters, is shown in Figure 8. The first thing to notice is that the bass line is more characteristic of the corpus than that of the version 1 harmonisation. This could well be due to the fact that this version employs specialist systems for the prediction of bass given soprano. It is rather jumpy in the last phrase, however, and in the final bar there is a parallel unison with the tenor. The second chord of the second bar does not fit in with its neighbouring chords, and there should be a root position tonic chord on the third beat of the fourth bar. On the positive side, there is a fine example of a passing note at the beginning of the fifth bar; and the harmony at the end of the third phrase, with the chromatic tenor movement, is rather splendid.

Conclusion

The first set of version 2 viewpoint selection runs, for attribute prediction together using the seen `Pitch` domain, compare different combinations of two-stage prediction. By far the best performance is obtained by predicting the bass part first followed by the inner parts together, reflecting the usual human approach to harmonisation. It is interesting to note that this heuristic, almost universally followed during harmonisation, therefore has an information theoretic explanation for its success.

Having demonstrated the extent to which multiple viewpoint systems have specialised in order to carry out these two rather different prediction tasks, we use an even greater number of specialist systems in a second set of runs. These show that better models can be created by selecting separate multiple viewpoint systems to predict individual musical attributes, rather than a single system to predict them all.

In comparing version 1 with version 2, only `Cont` and `Pitch` are taken into consideration, since the prediction of `Duration` is not directly comparable. On this basis, version 2 is better than version 1 when using corpus 'A', which again tallies with human experience of harmonisation; but when corpus 'A+B' is used, their performance is identical. We can infer from this that version 1 creates more general models, better able to scale up to larger corpora which may deviate somewhat from the characteristics of the original corpus. Conversely, version 2 is capable of constructing models which are more specific to the corpus for which they are selected.

Acknowledgements

We wish to thank the three anonymous reviewers for their constructive and insightful comments, which greatly improved this paper.

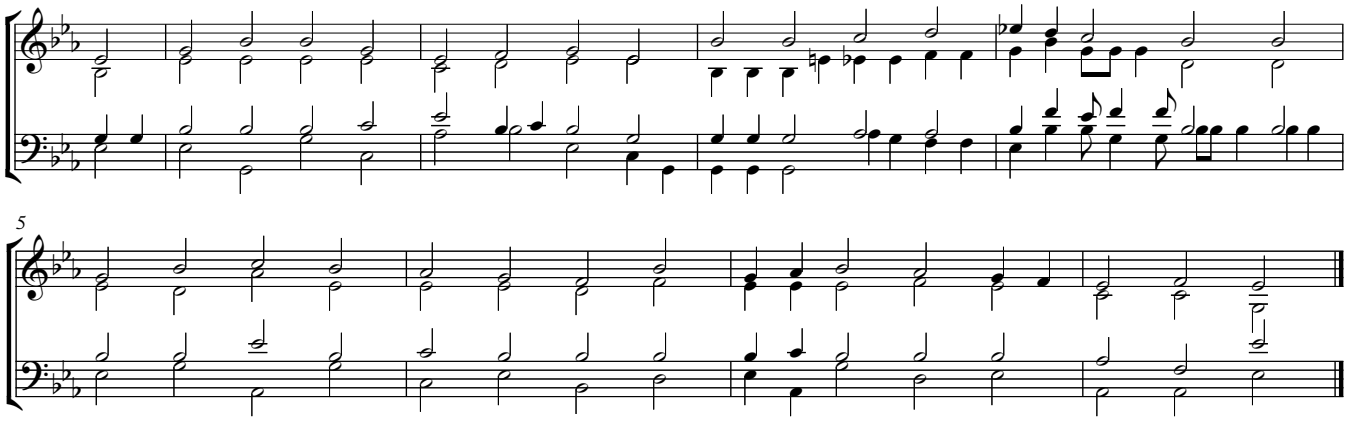


Figure 7: Relatively successful harmonisation of hymn tune *Das walt' Gott Vater* (Vaughan Williams 1933, hymn no. 36) automatically generated by the best version 1 model with optimised probability threshold parameters, using corpus 'A+B'.

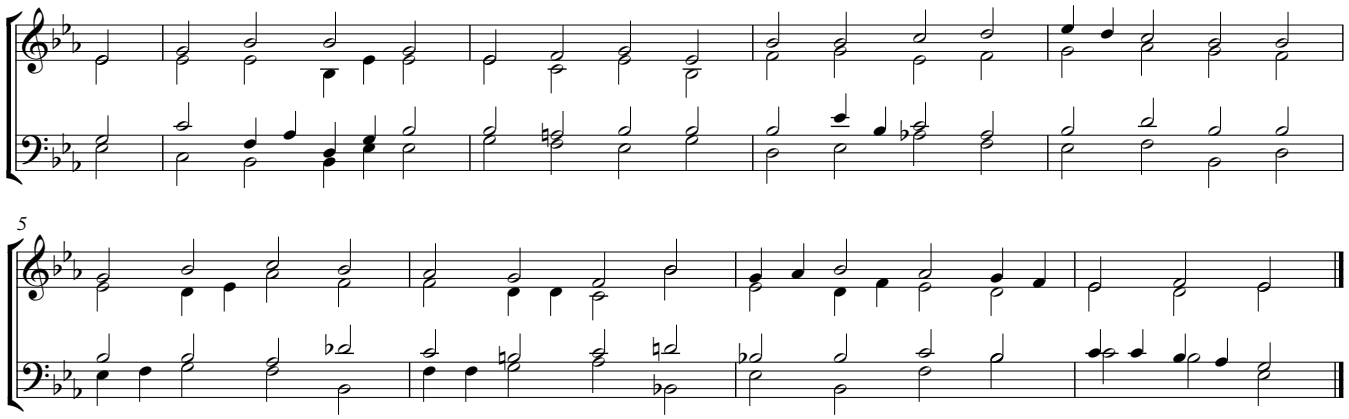


Figure 8: Relatively successful harmonisation of hymn tune *Das walt' Gott Vater* (Vaughan Williams 1933, hymn no. 36) automatically generated by the best version 2 model with optimised probability threshold parameters, using corpus 'A+B'.

References

Allan, M., and Williams, C. K. I. 2005. Harmonising chorales by probabilistic inference. In L. K. Saul; Y. Weiss; and L. Bottou., eds., *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Cleary, J. G., and Witten, I. H. 1984. Data compression using adaptive coding and partial string matching. *IEEE Trans Communications* COM-32(4):396–402.

Conklin, D., and Witten, I. H. 1995. Multiple viewpoint systems for music prediction. *Journal of New Music Research* 24(1):51–73.

Conklin, D. 1990. Prediction and entropy of music. Master's thesis, Department of Computer Science, University of Calgary, Canada.

Conklin, D. 2002. Representation and discovery of vertical patterns in music. In C. Anagnostopoulou; M. Ferrand; and A. Smaill., eds., *Music and Artificial Intelligence: Proc. ICMAI 2002, LNAI 2445*, 32–42. Springer-Verlag.

Hild, H.; Feulner, J.; and Menzel, W. 1992. Harmonet: A neural net for harmonizing chorales in the style of J. S. Bach. In R. P. Lippmann; J. E. Moody; and D. S. Touretzky., eds., *Advances in Neural Information Processing Systems*, volume 4, 267–274. Morgan Kaufmann.

Jurafsky, D., and Martin, J. H. 2000. *Speech and Language Processing*. New Jersey: Prentice-Hall.

Manning, C. D., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Pearce, M. T.; Conklin, D.; and Wiggins, G. A. 2004. Methods for combining statistical models of music. In *Proceedings of the Second International Symposium on Computer Music Modelling and Retrieval*.

Vaughan Williams, R., ed. 1933. *The English Hymnal*. Oxford University Press.

Automatical Composition of Lyrical Songs

Jukka M. Toivanen and **Hannu Toivonen** and **Alessandro Valitutti**
HIIT and Department of Computer Science
University of Helsinki
Finland

Abstract

We address the challenging task of automatically composing lyrical songs with matching musical and lyrical features, and we present the first prototype, M.U. Sicus-Apparatus, to accomplish the task. The focus of this paper is especially on generation of art songs (lieds). The proposed approach writes lyrics first and then composes music to match the lyrics. The crux is that the music composition subprocess has access to the internals of the lyrics writing subprocess, so the music can be composed to match the intentions and choices of lyrics writing, rather than just the surface of the lyrics. We present some example songs composed by M.U. Sicus, and we outline first steps towards a general system combining both music composition and writing of lyrics.

Introduction

Creation of songs, combinations of music and lyrics, is a challenging task for computational creativity. Obviously, song writing requires creative skills in two different areas: composition of music and writing of lyrics. However, these two skills are not sufficient: independent creation of an excellent piece of music and a great text does not necessarily result in a good song. The combination of lyrics and music could sound poor (e.g., because the music and lyrics express conflicting features) or be downright impossible to perform (e.g., due to a gross mismatch between pronunciation of lyrics and rhythm of the melody).

A crucial challenge in computational song writing is to produce a coherent, matching pair of music and lyrics. Given that components exist for both individual creative tasks, it is tempting to consider one of the two following sequential approaches to song writing:

- First write the lyrics (e.g. a poem). Then compose music to match the generated lyrics.

Or:

- First compose the music. Then write lyrics to match the melody.

Obviously, each individual component of the process should produce results that are viable to be used in songs. In addition, to make music and lyrics match, the second step should be able to use the result from the first step as its guidance. Consider, for instance, the specific case where lyrics are written first. They need to be analyzed so that matching music can be composed.

Several issues arise here. The first challenge is to make such a modular approach work on a surface level. For instance, pronunciation, syllable lengths, lengths of pauses, and other phonetic features related to the rhythm can in many cases be analyzed by existing tools. The composition process should then be able to work under constraints set by these phonetic features, to produce notes and rhythmic patterns matching the phonetics. Identification of relevant types of features, their recognition in the output of the first step of the process, and eventually generation of matching features in the second step of the process are not trivial tasks.

The major creative bottleneck of the simple process outlined above is making music and lyrics match each other at a deeper level, so that they jointly express the messages, emotions, feelings, or whatever the intent of the creator is. The pure sequential approach must rely on analysis of the lyrics to infer the intended meaning of the author. Affective text analysis may indicate emotions, and clever linguistic analysis may reveal words with more emphasis. However, text analysis techniques face the great challenge of natural language understanding. They try to work backwards from the words to the meaning the author had in mind. In the case of composing music first and then writing corresponding lyrics, the task is equally challenging.

Fortunately, in an integrated computational song writing system, the second step can have access to some information about the creative process of the first step, to obtain an internal understanding of its intentions and choices. Figuratively speaking, instead of analyzing the lyrics to guess what was in the mind of the lyricist, the composer looks directly inside the head of the lyricist. We call this approach *informed sequential* song writing (Figure 1). In this model, information for the music composition process comes directly from the lyrics writing process, as well as from text analysis and user-given input.

In this paper we study and propose an instance of the informed sequential song writing approach. The presented system, M.U. Sicus-Apparatus, writes lyrics first and then composes matching music. Since lyrics generation is in this approach independent of music composition, our emphasis will be on the latter. Empirical evaluation of the obtained results is left for future work.

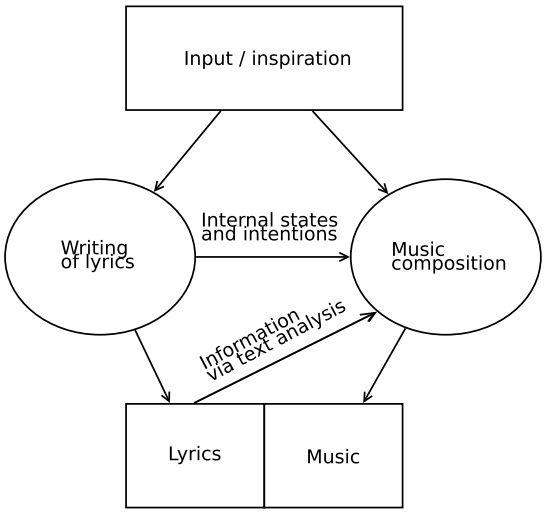


Figure 1: Schema of the informed sequential song generation

Art Songs

Songs can be divided in rough categories like art, folk, and pop songs. This paper concentrates on the genre of so called art songs which are often referred to as lieds in the German tradition or mélodies in the French tradition. Art songs are a particularly interesting category of compositions with strong interaction of musical and lyrical features. Finest examples of this class include the songs composed by F. Schubert. Art songs are composed for performance, usually with piano accompaniment, although the accompaniment may be written for an orchestra or a string quartet as well. [1]

Art songs are always notated and the accompaniment, which is considered to be an important part of the composition, is carefully written to suit the overall structure of the song. The lyrics are often written by a poet or lyricist and the music separately by a composer. The lyrics of songs are typically of a poetic, rhyming nature, though they may be free prose, as well. Quite often art songs are through-composed which means that each section of the lyrics goes with fresh music. In contrast, folk songs and some art songs are strophic which means that all the poem's verses are sung to the same melody, sometimes possibly with little variations. In this paper, we concentrate on through-composed art songs with vocal melody, lyrics, and piano accompaniment.

Related Work on Music and Poetry Generation

Generation of music and poetry on their own right have been studied separately in the field of computational creativity and there have been a few attempts to study the interaction of textual and musical features (Mihalcea and Strapparava 2012). Some attempts have also been made to compose musical accompaniments for text (Monteith et al. 2011; Monteith, Martinez, and Ventura 2012). Interestingly however, generation of lyrical songs has received little attention in the past. Because of the lack of earlier work on combining music and lyrics in a single generative system, we next briefly review work done in the area of music and poetry/lyrics generation separately.

Song Generation

Composing music algorithmically is an old and much studied field. Several different approaches and method combinations have been used to accomplish this task (Roads 1996). One of the most well-known examples, usually known as Mozart's Musikalisches Würfelspiel, dates back to the year 1792, long before modern computers. Many musical procedures such as voice-leading in Western counterpoint can be reduced to algorithmic determinacy. Additionally algorithms originally invented in other fields than music such as L-systems, fractals, constraint based methods, Hidden Markov Models, and conversion of arbitrary data like electro-magnetic fields into music, have been used as the basis for music composition. A review of the approaches used in algorithmic music composition is outside the scope of this paper. For example, Roads (1996) presents a good overview of different methodologies.

Monteith et al. (2012) have proposed a model of generating melodic accompaniments for given lyrics. This approach concentrates on the extraction of linguistic stress patterns and composition of a melody with matching note lengths and fulfilment of certain aesthetic metrics for musical and linguistic match. Differently from this approach, our system composes all aspects of a song including the lyrics, harmony, and melody, and thus it is not limited to the musicalization of existing lyrics. It also employs an informed-sequential architecture and thus the integration of lyrics writing and music composition subprocesses is tighter.

Poetry or Lyrics Generation

A number of approaches and techniques exist for automatic generation of poetry (Manurung, Ritchie, and Thompson 2000; Gervás 2001; Manurung 2003; Toivanen et al. 2012). Some systems have also been proposed to be used for generating song lyrics (Ramakrishnan, Kuppan, and Devi 2009) and not only pure poetry. Again, a review of the approaches used to produce poetry or lyrics automatically is outside the scope of this paper.

Informed Sequential Song Generation

The lyrics part of the song contains the denotational content of the song and partly some connotational aspects like word choices and associations. In the current implementation, the lyrics are written about a user-specified theme (e.g. life) (Toivanen et al. 2012). The music composition module, on the other hand, conveys only connotational information: in the current implementation mood and intensity of the song. The mood is a user-specified input parameter, currently sad or happy, respectively corresponding to positive or negative

[1] Sometimes songs with other instruments besides piano are referred to as vocal chamber music and songs for voice and orchestra are called orchestral songs.

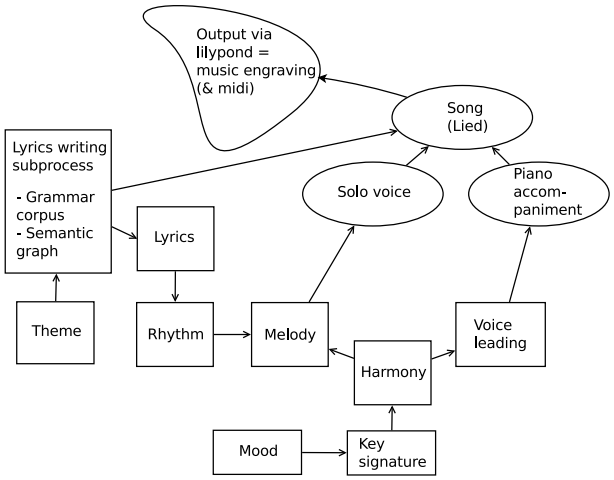


Figure 2: Detailed structure of M.U. Sicus-Apparatus

emotional valence. Intensity corresponds to the emotional arousal to be expressed in the song. It comes from the lyrics writing process and illustrates how internal information of creative processes can be passed between the subprocesses. It is also used as a way to direct the attention to the words expressing the input theme.

We employ the informed sequential song generation scheme with the overall flow of Figure 2. First, the user provides a theme (e.g., snow) and mood (e.g., happy) of the song. M.U. Sicus-Apparatus then generates lyrics for the song that tell about the given theme. The rhythm of the melody is composed by a stochastic process that takes into account the number of syllables, syllable lengths, and punctuation of the lyrics. The harmony of the song is generated either in a randomly selected major (for happy songs) or minor (for sad songs) according to the user's input. Next we discuss each of these phases and the overall structure of M.U. Sicus-Apparatus in more detail.

Lyrics Generation

The lyrics for a new song consist of a verse of automatically generated poetry. Typically a theme for the song is given by the user, and the method then aims to provide a new and grammatically well structured poem with content related to the theme. For lyrics generation, we use the method of Toivanen et al. (2012). We give a short overview of the methodology here.

The lyrics generation method is designed to avoid explicit specifications for grammar and semantics, in order to reduce human effort in modeling natural language generation. Instead of explicit rule systems, the method uses existing corpora and statistical methods. One of the reasons behind this approach is also to keep language-dependency of the methods small. The system automatically learns word associations to model semantic relations. An explicit grammar is avoided by using example instances of actual language use and replacing the words in these instances by words related to a given theme in suitable morphological forms.

As the lyrics writing module is writing lyrics for the song it subsitutes varying proportions of words in a randomly selected piece of text by new words (Toivanen et al. 2012). This proportion can vary between 0% and 100% for every individual line of lyrics although we required the overall substitution rate to be over 50% in the experiments for this paper. The arousal level of the song in a particular place is determined by this substitution rate as discussed in the Dynamics section.

Music Generation

As an overview, M.U. Sicus-Apparatus works as follows. The system first generates a rhythm for the melody, based on the phonetics of the lyrics already written. A harmonical structure is then generated, followed by generation of a melody matching the underlying harmony. A piano accompaniment is generated directly from the harmony with additional rules for voice leading and different accompaniment styles. Finally the resulting song is transformed to a music sheet and a midi file. We next discuss each of the phases in some more detail.

Affective Content Affective connotation has a central role in the overall process. It is provided by the combination of two elements. The first one is the emotional valence, expressing the input mood via harmony and melody. The second element is intensity, expressing emergent information of the lyrics writing process (i.e. word replacement rates, see below).

Rhythm of the Melody The rhythm generation procedure takes into account the number of syllables in the text, lengths of the syllables, and punctuation. Words in the lyrics are broken into syllables and the procedure assigns for every word a rhythmic element with equally many notes as there are syllables in the word. These rhythmic elements are randomly chosen from a set of rhythmic patterns usually found in art songs so that in addition to the number of syllables also the syllable lengths constrain the set of possible candidates. Longer syllables get usually longer time values and shorter syllables get usually shorter time values. The punctuation mark is often stressed with a rest in the melody rhythm.

Harmony The harmony is composed according to the user-specified mood. If the valence polarity of the mood is positive the key signature is constrained to major and then randomly selected from the set of possible major keys. In the opposite case the key is selected from the set of minor keys.

The system database contains different sets of harmonic patterns regularly found in diatonic western classical music for major and minor keys. The construction of harmony is based on a second-order Markov-chain selections of these harmonic patterns and expression of these as chord sequences in a given key. A typical harmonic pattern is, for instance, the chord sequence *I*, *IV*, *V*. When dealing with minor keys, harmonic minor scale is used. The harmony generation procedure also assigns time values for each of the chords in a probabilistic manner so that the length of the generated harmonical structure matches the length of the

melody rhythm generated earlier. Usually each chord is assigned a time value of either half note or a whole note. After generating the sequence of chords, the method moves on to determine pitches of the melody notes.

Melody The melody note pitches are generated on the basis of the underlying harmony and pitch of the previous note by a random walk. Firstly, the underlying chord defines a probability distribution for pitches which can be used. For example, if the underlying chord is C major as well as the key signature, the notes *c*, *e*, and *g* are quite probable candidates, *a*, *f* and *d* are less probable and *h* is even less probable. Secondly, the pitch of the previous note affects the pitch of the next note in a way that small intervals between these two notes are more probable than large intervals. Finally, the note pitch is generated according to a combined probability distribution that is a product of the probability distribution determined by the underlying chord and the probability distribution determined by the previous melody note.

Accompaniment and Voice Leading The harmonical structure provides the basic building blocks of the accompaniment but the chord sequence can be realised in many styles. Currently, we have implemented several different styles like Alberti bass and other chord patterns.

In order to have smooth transitions between chords in the accompaniment, we apply a simple model of voice leading. For a given chord sequence our current implementation chooses chord inversions that lead to minimal total movement i.e. smallest sum of intervals, of simultaneous voices.

Dynamics The arousal level of the song is expressed as dynamic marks in the music. Higher arousal is associated with higher loudness (e.g. forte) and lower arousal is associated with more peaceful songs (e.g. piano). For every line of lyrics this proportion of substituted words (S) in a line of poetry is expressed in the music either as piano (*p*, $S < 25\%$), mezzo-piano (*mp*, $25\% < S < 50\%$), mezzo-forte (*mf*, $50\% < S < 75\%$), or forte (*f*, $75\% < S < 100\%$).

Output The system outputs both sheet music to be performed by musicians and a midi file to be played through a synthesizer. The music engraving is produced with the LilyPond music score language (Nienhuys and Nieuwenhuizen 2003).

Examples

Figures 3 and 4 contain two example songs generated by M.U. Sicus-Apparatus[2]. The song in Figure 3 is a sad one about life, and the one in Figure 4 is a happy song about flower buds. The words that have been emphasised by the lyrics writing process are marked in bold in lyrics.

The proposed methodology seems to provide relatively good combinations of text and music. As explained above, the transmission of information on song dynamics comes directly from the lyrics writing process. This is interesting because that particular information would be impossible to extract directly from the lyrics itself. For instance, in the

[2]These and other songs are available also in midi form at http://www.cs.helsinki.fi/discovery/mu-sicus

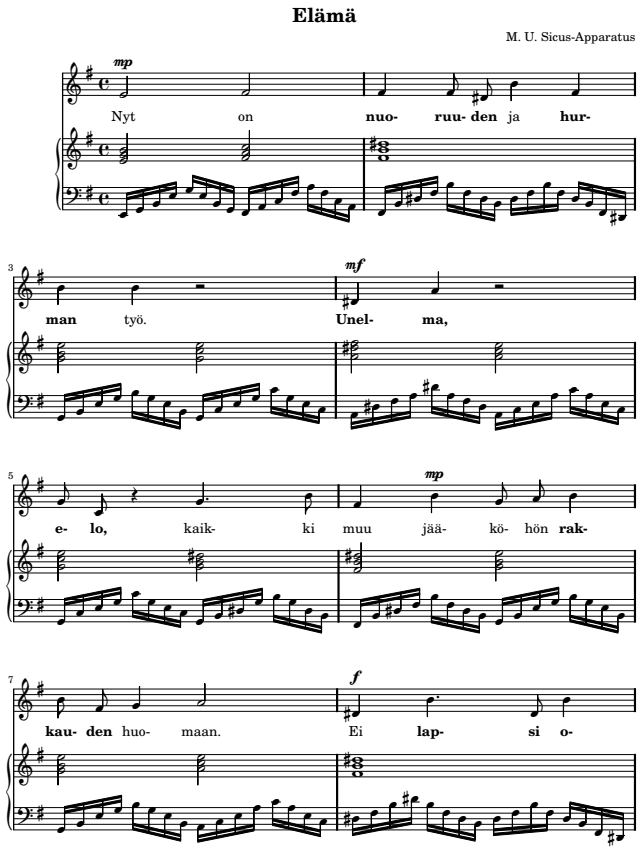


Figure 3: Excerpt of a sad song composed with the theme "life" (in Finnish "elämä").

song of Figure 4, first two phrases have a very high arousal (forte) due to the high emphasis on the overall song theme whereas after that the arousal calms down.

Taking the syllable lengths and punctuation into account in the rhythm seems to lead to quite singable melodies (our subjective view which needs to be evaluated objectively in later work). However, taking the syllable stress into account as well could lead to further improvements.

The melody, harmony, and rhythm seem to constitute quite a coherent whole. The major weakness in the music is a lack of clear phrase structures, which has also been a problem in many music generation systems before. The lyrics writing method has been evaluated earlier with good results by Toivanen et al. (2012).

Conclusions and Future Work

We have proposed the task of generating lyrical songs as a research topic of computational creativity. This topic has received only little attention in the past although both music composition and poetry/lyrics generation have been studied on their own.

As a first step towards a generative music-lyrics model we have implemented a system, M.U. Sicus-Apparatus, that

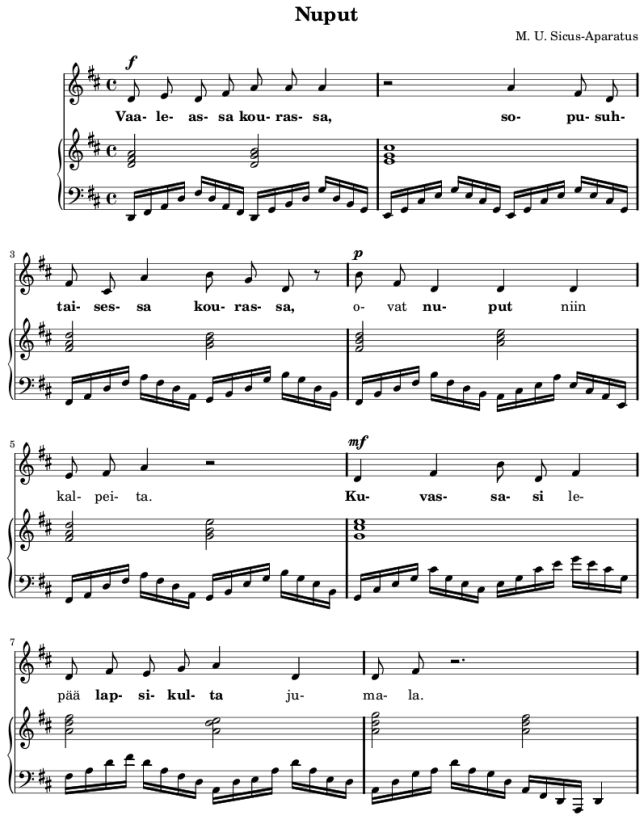


Figure 4: Example of a happy song composed with the theme "flower buds" (in Finnish "nuput").

generates simple lyrical art songs. The current system composes happy and sad songs about a given theme by writing first lyrics of the song and composing then music with a melody rhythm that matches the phonetic structure of the lyrics. The system works in an informed-sequential manner. This means that writing of the lyrics and composition of the music are not performed separately but the lyrics writing module can convey part of its internal data structures directly to the music composition system.

An automatical generation procedure of lyrical and musical content also offers interesting possibilities for musicalization of data (Tulilaulu et al. 2012). For example, converting news stories automatically into songs could be an interesting application of the presented methodology.

In the future, we would like to carry out an empirical evaluation of the methods and results using recorded performances of a collection of songs. As a further step, we would like to study how emotional states can be transferred in songs, partly by conveying the affective state in the lyrics and partly by modifying the tempo, loudness, modality, melody movements, and rhythm of the music. A wide body of research exists on correlation of musical features with perceived affective states. For example, varying the note rate in the accompaniment and using dissonance and consonance as well as different instrumental techniques like staccato to convey intensity, could be used to improve the system.

The ultimate goal is to break the inherently sequential structure of the architecture, and to develop a song generation system with a much tighter integration or interaction between the lyrics writing and music composition processes.

Acknowledgements

This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland under grant 118653.

References

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. *Journal of Knowledge-Based Systems* 14(3–4):181–188.

Manurung, H. M.; Ritchie, G.; and Thompson, H. 2000. Towards a computational model of poetry generation. In *Proceedings of AISB Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, 79–86.

Manurung, H. M. 2003. *An Evolutionary Algorithm Approach to Poetry Generation*. Ph.D. Dissertation, University of Edinburgh, Edinburgh, United Kingdom.

Mihalcea, R., and Strapparava, C. 2012. Lyrics, music, and emotions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2012)*, 590–599.

Monteith, K.; Francisco, V.; Martinez, T.; Gervás, P.; and Ventura, D. 2011. Automatic generation of emotionally-targeted soundtracks. In *Proceedings of the Second International Conference on Computational Creativity*, 60–62.

Monteith, K.; Martinez, T.; and Ventura, D. 2012. Automatic generation of melodic accompaniments for lyrics. In *Proceedings of the International Conference on Computational Creativity*, 87–94.

Nienhuys, H. W., and Nieuwenhuizen, J. 2003. Lilypond, a system for automated music engraving. In *Colloquium on Musical Informatics (XIV CIM 2003)*.

Ramakrishnan, A.; Kuppan, S.; and Devi, S. L. 2009. Automatic generation of tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity (CALC'09)*, 40–46.

Roads, C. 1996. *The Computer Music Tutorial*. The MIT Press.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *International Conference on Computational Creativity*, 175–179.

Tulilaulu, A.; Paalasmaa, J.; Waris, M.; and Toivonen, H. 2012. Sleep musicalization: Automatic music composition from sleep measurements. In *Eleventh International Symposium on Intelligent Data Analysis (IDA)*, volume 7619 of *LNCS*, 392–403.

Implications from Music Generation for Music Appreciation

Amy K. Hoover, Paul A. Szerlip, and Kenneth O. Stanley
Department of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2362 USA
{ahoover@eecs.ucf.edu,paul.szerlip@gmail.com,kstanley@eecs.ucf.edu}

Abstract

This position paper argues that fundamental principles that are exploited to achieve effective music generation can also shed light on the elusive question of why humans appreciate music, and which music is easiest to appreciate. In particular, we highlight the key principle behind an existing approach to assisted accompaniment generation called *functional scaffolding for musical composition* (FSMC). In this approach, accompaniment is generated as a *function* of the preexisting parts. The success of this idea at generating plausible accompaniment according to studies with human participants suggests that perceiving a *functional relationship* among parts in a composition may be essential to the appreciation of music in general. This insight is intriguing because it can help to explain without any appeal to traditional music theory why humans with no knowledge or training in music can nevertheless find satisfaction in coherent musical structure.

Introduction

Among the most fundamental questions on the human experience of music is why we appreciate it so universally and what makes some pieces more appealing than others (Hanslick, 1891; Sacks, 2008; Frith, 2004; Gracyk, 1996). There are many possible approaches to addressing these questions, from studies of expectation fulfillment (Huron, 2006; Schmuckler, 1989; Pearce and Wiggins, 2012; Abdallah and Plumbley, 2009) to cultural factors (Balkwill and Thompson, 1999; Peddie, 2006). Our aim in this paper is to propose an alternative route to addressing the fundamental basis for music appreciation, by beginning with an approach to *music generation* and from its mechanics drawing implications for at least one key underlying ingredient in the appreciation of music. The motivation is that the process of designing an effective music generator implicitly forces the designer to confront the basis of music appreciation as well. After all, a music generator is little use if its products are not appealing.

Particularly revealing would be a simple principle that almost always can be applied. The simpler such a principle, the more plausible that it might explain some aspect of music appreciation. One approach to assisted music generation based on such a simple principle is called *functional scaffolding for musical composition* (FSMC) (Hoover and Stanley, 2009; Hoover, Szerlip, and Stanley, 2011b,a; Hoover et al., 2012). Our position is that the principle at the heart of this approach, initially conceived as a basis for generating accompaniment, offers a unique hint at the machinery behind human musical appreciation. In this way, it can contribute to explaining in part both when and why humans appreciate music.

Functional Scaffolding for Musical Composition (FSMC)

The FSMC approach is based on the insight that music is at heart a pattern of notes played over time with some regularity. As a result, one way to conceptualize music is as a *function of time*. Formally, for any musical voice, the pattern of pitches and the pattern of durations and rests can be expressed together as a vector function of time $f(t)$ that outputs both pitch and rhythm information. In practice, to generate a sequence of notes, f could be queried at every time t and the complete output sequence would constitute the pattern. The parts played by each instrument in an ensemble piece could also be output simultaneously by such a function.

This perspective is helpful for music generation when combined with the insight that all the instrumental sequences (i.e. each track) in a single piece must be somehow related to each other. For example, in a popular rock piece, the drum pattern, say $d(t)$, typically establishes the rhythm for the rest of the piece. Therefore, the bass pattern, $b(t)$, which helps structure the harmonic form, will by necessity depend in some way upon the drum pattern. This idea of relatedness between parts can be expressed more formally by saying that the bass pattern is a *function* of the drum pattern, which can be expressed by a function h that relates $b(t)$ to $d(t)$: $b(t) = h(d(t))$. Building on the drum and bass patterns, vocalists and other instrumental parts can then explore more complicated melodic patterns that are themselves also related to the established rhythmic and chord patterns. It follows then that not only can each of these instrumental parts be represented as a function of time, but that they are indeed each functions of *each other*.

Beyond just observations, these insights imply a practical opportunity for generating musical accompaniment. By casting instrumental parts as functions of each other, the problem of accompaniment is illuminated in a new light:

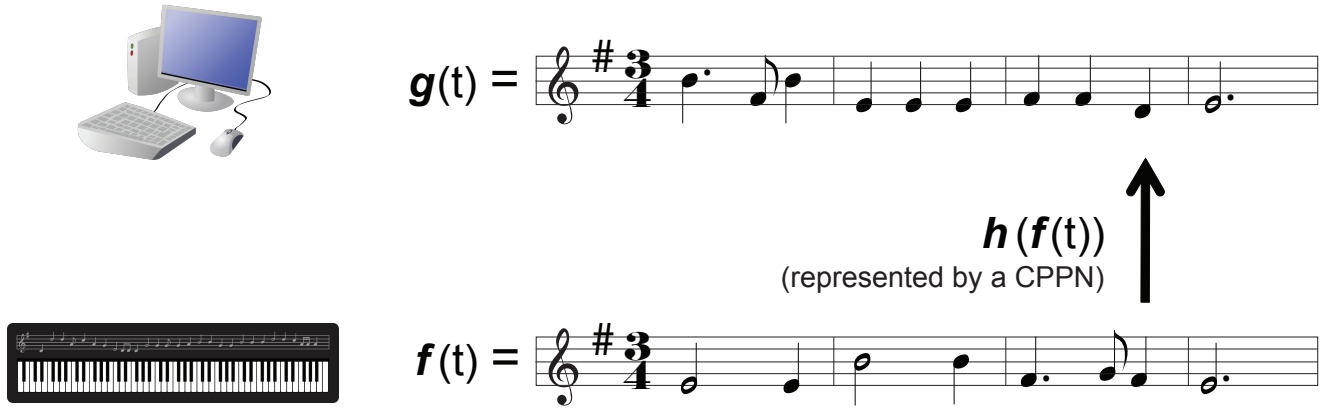


Figure 1: **Representing and Searching for Accompaniments with FSMC.** The function $f(t)$, which is depicted by a piano keyboard, represents the human composition called the *scaffold*, from which the computer-generated accompaniments are created. A possible such accompaniment, $g(t)$, is shown atop and depicted by the image of a computer. Each accompaniment is internally represented by a helper function, $h(f(t))$, which is represented by a special type of artificial neural network (ANN) called a compositional pattern producing network (CPPN). Like ANNs, CPPNs can theoretically approximate any continuous function. Thus these CPPNs represent h, which transforms the scaffold into an accompaniment.

Given an existing part $f(t)$, the problem of formulating an appealing accompaniment becomes the problem of searching for accompaniment $g(t)$ such that $g(t)$ complements $f(t)$. Yet while applying a search algorithm directly to finding such a function $g(t)$ would be difficult because the search space is vast, instead the search can be significantly constrained by searching for $h(f(t))$, as depicted in figure 1. The major benefit of this approach is that because h is a function of the part it will accompany, it cannot help but follow to some extent its contours. Therefore, the idea for generating accompaniment in FSMC is to search with the help of a human user for a *function* $h(f(t))$, where $f(t)$ is a preexisting part or *scaffold*. By searching for a transforming function instead of an explicit sequence of notes, the plausibility of output accompaniments is enhanced. In effect, $f(t)$ provides the *functional scaffolding* for the accompaniment.

The idea in FSMC that searching for $h(f(t))$ can yield plausible accompaniment to $f(t)$ can be exploited in practice by programming a search algorithm to explore possible variations of the function h. In fact, this approach has been tested extensively in practice through an implementation called MaestroGenesis (`http://maestrogenesis.org/`), whose results have been reported in a number of publications (Hoover, Szerlip, and Stanley, 2011b,a; Hoover et al., 2012). In MaestroGenesis, the function h is represented by a special kind of artificial neural network called a compositional pattern producing network (CPPN). A population of candidate CPPNs is *evolved* interactively by allowing a human user to direct the search algorithm by picking his or her favorite candidate accompaniments to produce the offspring for the next generation. Thus the *representation* of the transforming function is a CPPN (which is a kind of function approximator) and the *search algorithm* is interactive evolution (which is an evolutionary algorithm guided by a human; Takagi, 2001). A full technical description is given in Hoover, Szerlip, and Stanley (2011a), Hoover, Szerlip, and Stanley (2011b), and Hoover et al. (2012).

Interestingly, listener study results from FSMC-generated music showed that musical pieces with accompaniments that were generated purely through functional relationships were indistinguishable from fully human composed pieces (Hoover, Szerlip, and Stanley, 2011a). In fact, some fully human composed pieces were rated more mechanical-sounding than those that were only partially human composed. Similarly positive results were also reported in other studies (Hoover and Stanley, 2009; Hoover, Szerlip, and Stanley, 2011b,a; Hoover et al., 2012). Although more variation in the initial human composition (i.e. a polyphonic versus monophonic scaffold) provides more richness from which to work, as Hoover et al. (2012) show, plausible accompaniments can nevertheless be generated from as little as a single monophonic starting melody. Furthermore, often in MaestroGenesis even the first generation of candidate accompaniments, which are randomly-generated CPPNs, sounds plausible because the functional relationship ensures at least some relationship between the scaffold and its accompaniment (Hoover and Stanley, 2009).

From Music Generation to Music Appreciation

These results are of course relevant to progress in music generation, but they hint at a deeper implication. In particular, it is notable that MaestroGenesis (and FSMC behind it) has *almost no musical knowledge* programmed into it. In fact, the only real musical rule in the program is that CPPN outputs are forced to be interpreted as notes within the key of the scaffold track. Aside from that, MaestroGenesis has no knowledge of chords, rhythm, progression, melody, har-

mony, dissonance, style, genre, or anything else that a typical music generator might have (Simon, Morris, and Basu, 2008; Chuan, 2009; Ebcioglu, 1990). It thus relies almost entirely on the functional relationship between the scaffold and the accompaniment to achieve plausibility. In effect, the functional relationship causes the accompaniment to *inherit* the gross structure of the scaffold, thereby endowing it with many of the same aesthetic properties. Thus the key observation behind this position paper is that establishing such a functional relationship between different parts of a song seems to be *sufficient on its own* to achieve plausible musical structure.

This observation is intriguing because it implies a hypothesis about the nature of musical appreciation: If a functional relationship alone is sufficient to achieve musical plausibility in the experience of human listeners, then perhaps musical appreciation itself is at least in part the result of perceiving a functional relationship between different parts of a composition. That is, functional relationships, which are *mathematical properties* of patterns that do not require any specific musical knowledge to perceive, could explain why listeners without any musical training or expertise nevertheless experience and appreciate music and separate it firmly from cacophony. In effect the human is appreciating the functional relationship that binds different parts of a composition together.

If true, this hypothesis can explain to some extent when humans will or will not appreciate a composition. For example, the harder it is to perceive how one part is functionally related to another, the less pleasing that piece may be. Such functional relationships are potentially perceived not only between different instrumental parts or tracks, but also within a single instrumental part played over time. That is, if a functional relationship can be perceived between an earlier sequence of notes and a later one, then the entire sequence may succeed as musically plausible or even appealing.

At the same time, it may also explain why some compositions are more difficult to enjoy. For example, some research in computer music explores the sonification of non-auditory data (Cope, 2005; Park et al., 2010; Vickers, 2005). Typically, the user inputs semi-random data to a computer model (e.g. cellular automata, swarms, etc.) that outputs music. While the output is a function of the input, because the initial seed does not stem from inherently musical events, the outputs are often difficult for non-creators to immediately understand. However, as these systems develop, composers begin to build musical frames for anticipation and expectation. While there can certainly be beauty in such pieces, the audience needs some familiarity, like the composer, with the style to begin to perceive the important relationships.

Though perhaps not explicitly, composers have long intuited the importance of incorporating functional relationships into compositions. Not only are musical lines regularly translated, inverted, and reflected, but some logarithmic and modular transformations (and set theory concepts) predate the mathematical formalisms themselves (Risset, 2002; Harkleroad, 2006). The implicit nature of the composers' insight is that people appreciate these functional transformations within a "relatable" musical context.

In fact, much of compositional musical theory was developed to produce consistent aesthetic results (Payne, 1995; Christensen, 2002). By following certain heuristics and established patterns, composers ensure that pieces fit within particular styles and genres. Such heuristics generally encompass narrow sets of phenomena, e.g. waltzes, counterpoint, jigs, etc. The hypothesis that functional relationships provide a general principle for musical appreciation provides a unifying perspective for all such disparate stylistic conventions: At some level, all of them ultimately establish some kind of functional relationship among the parts of a composition.

Furthermore, this perspective suggests that as long as they are perceptible (i.e. not so complex as to sound cacophonous), relatively simple functions likely exist that generate relationships among parts that are aesthetically appealing yet not related to any genre, rule, or heuristic currently taught or even yet conceived. For example, music generated with FSMC exhibits a range of complexity, suggesting little restriction on the type of function necessary to create plausible accompaniments. Some of the most appealing accompaniments are generated from very simple relationships (Hoover, Szerlip, and Stanley, 2011b,a), while sometimes more complex relationships between melody and percussion are also appreciated by listeners (Hoover and Stanley, 2009). To some extent this theory thereby suggests without any other musical theory when breaking the rules might be appealing and when it might not; as long as a functional relationship among parts can still be perceived, the average listener will not necessarily react negatively to breaking established conventions.

Sometimes it can also take a while to habituate to styles or genres that do not follow conventional rules. For example, atonal pieces can be difficult to enjoy for the uninitiated. Interestingly, the functional hypothesis provides a potential insight into why such unconventional styles can become appealing with experience. The explanation is that initially the functional relationships among different parts in such an unconventional context are difficult to perceive because the relationships are both complex and unfamiliar. Therefore, the brain initially struggles to identify the functional relationship. However, over time, repeated exposure familiarizes the listener with the kinds of transformations that are typical in the new context, such that eventually the brain can pick out functional relationships that once were too complex to perceive. At that point, the music becomes possible again to appreciate. In fact, as noted by Huron (2006), the patterns associated with a particular style are designed to elicit emotions by playing on the listeners' expectations. Those expectations can be viewed as mediated by the kinds of functional relationships with which the listener is familiar.

Music theory provides many heuristics for composing plausible types of music like fugues or walking bass lines. But as any musician knows, simply following such rules without the elusive element of inspiration results in plausible yet dry-sounding pieces. A good musician must know the standard rules for composition and also when to break them, but the problem of when to break the rules in music theory is less understood than the standard rules for composition. The

insight that a rule is well-broken if it still preserves a perceptible functional relationship provides a possible direction for studying this issue further.

Conclusions

This type of general theory follows directly from taking a minimalist approach to music generation. Approaches that rely on acquiring or enumerating all the complexities of music of certain types or composers of certain types, such as through statistical inference (Rhodes, Lewis, and Müllensiefen, 2009; Kitani and Koike, 2010) or grammatical rules (Holtzman, 1981; McCormack, 1996), cannot probe the possibility of deeper underlying principles than the rules that are apparent at the surface. In contrast, FSMC and MaestroGenesis took the minimalist approach to music generation by predicating everything only on functional relationships. While a potential criticism of such an approach is that it is too simplistic to capture all the subtlety of sophisticated musical composition, its benefit in a scientific context is that it *isolates* a single phenomenon so that the full implication of that phenomenon can be tested. The result is a simple hypothesis that reduces musical theory to a mathematical principle, i.e. perceiving functional relationships, that can plausibly be appreciated even by listeners without musical training. It also becomes a tool for music generation, as in MaestroGenesis, that does not require enumerating complex rules. While functional relationships need not constitute the entire explanation for all musical appreciation, they are an appealing ingredient because of their simplicity and possibility for future study – they suggest that within the mind of a composer perhaps at some level such a function is realized as the overall pattern of a musical piece is first conceived.

In a broader context, explaining the appreciation of music through perceiving functional relationships also connects musical appreciation to non-musical aesthetics. After all, across the spectrum of art, architecture, and even human beauty, symmetry, repetition, and variation on a theme are paramount. It is notable that all such regularities ultimately reduce to one instance of a pattern being functionally related to another. Given that we appreciate such relationships in so many spheres of our experience, that music too would draw from such an affinity follows elegantly.

Acknowledgments

This work was supported in part by the National Science Foundation under grant no. IIS-1002507 and also by a NSF Graduate Research Fellowship. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

References

Abdallah, S., and Plumbley, M. 2009. Information dynamics: Patterns of expectation and surprise in the perception of music. *Connection Science* 21(2).

Balkwill, L.-L., and Thompson, W. F. 1999. A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music Perception* 43–64.

Christensen, T. 2002. The cambridge history of western music theory. Cambridge University Press.

Chuan, C.-H. 2009. Supporting compositional creativity using automatic style-specific accompaniment. In *Proc. of the CHI Computational Creativity Support Workshop*.

Cope, D. 2005. *Computer Models of Musical Creativity*.

Ebcioglu, K. 1990. An expert system for harmonizing chorales in the style of j.s.bach. *Journal of Logic Programming* 145–185.

Frith, S. 2004. What is bad music? In Washburne, C., and Derno, M., eds., *Bad Music: The Music We Love to Hate*. Routledge. 15–36.

Gracyk, T. 1996. *Rhythm and Noise: An Aesthetics of Rock*. IB Tauris.

Hanslick, E. 1891. *On the Musically Beautiful: A Contribution Towards the Revision of the Aesthetics of Music*. Novello Ewer and Company.

Harkleroad, L. 2006. *The Math Behind the Music*. Outlooks. Cambridge University Press.

Holtzman, S. R. 1981. Using generative grammars for music composition. *Computer Music Journal* 5(1):51–64.

Hoover, A. K., and Stanley, K. O. 2009. Exploiting functional relationships in musical composition. *Connection Science Special Issue on Music, Brain, & Cognition* 21(2):227–251.

Hoover, A. K.; Szerlip, P. A.; Norton, M. E.; Brindle, T. A.; Merritt, Z.; and Stanley, K. O. 2012. Generating a complete multipart musical composition from a single monophonic melody with functional scaffolding. In *To appear in: Proceedings of the Third International Conference on Computational Creativity (ICCC-2012, Dublin, Ireland)*.

Hoover, A. K.; Szerlip, P. A.; and Stanley, K. O. 2011a. Generating musical accompaniment through functional scaffolding. In *Proceedings of the Eighth Sound and Music Computing Conference (SMC 2011)*.

Hoover, A. K.; Szerlip, P. A.; and Stanley, K. O. 2011b. Interactively evolving harmonies through functional scaffolding. In *Proceedings of the Genectic and Evolutionary Computation Conference (GECCO-2011)*. New York, NY: The Association for Computing Machinery.

Huron, D. 2006. *Sweet Anticipation: Music and the Psychology of Expectation*.

Kitani, K. M., and Koike, H. 2010. Improvgenerator: Online grammatical induction for on-the-fly improvisation accompaniment. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*.

McCormack, J. 1996. Grammar based music composition. *Complex Systems* 96:321–336.

Park, S.; Kim, S.; Lee, S.; and Yeo, W. S. 2010. Composition with path: Musical sonification of geo-referenced data with online map interface. In *Proceedings of the International Computer Music Conference*.

Payne, S. K. D. 1995. *Tonal Harmony: With an Introduction to Twentieth*. McGraw-Hill.

Pearce, M. T., and Wiggins, G. A. 2012. Auditory expectation: The information dynamics of music perception and cognition. *Topics in Cognitive Science* 4(4):625–652.

Peddie, I. 2006. *The Resisting Muse: Popular Music and Social Protest*. Ashgate Publishing Company.

Rhodes, C.; Lewis, D.; and Müllensiefen, D. 2009. Bayesian model selection for harmonic labelling. *Mathematics and Computation in Music* 107–116.

Risset, J.-C. 2002. Computing musical sound. In Assayag, G.; Feichtinger, H. G.; and Rodrigues, J. F., eds., *Mathematic and Music: A Diderot Mathematical Forum*. Springer-Verlag. chapter 13.

Sacks, O. 2008. *Musicophilia: Tales of Music and the Brain*. Vintage Canada.

Schmuckler, M. A. 1989. Expectation in music: Investigation of melodic and harmonic processes. *Music Perception: An Interdisciplinary Journal* 7(2):109–149.

Simon, I.; Morris, D.; and Basu, S. 2008. Mysong: Automatic accompaniment generation for vocal melodies. In *Proc. of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, 725–734. ACM.

Takagi, H. 2001. Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation. *Proceedings of the IEEE* 89(9):1275–1296.

Vickers, P. 2005. Ars informatica–ars electronica: Improving sonification aesthetics. In *Understanding and Designing for Aesthetic Experience Workshop at HCI 2005 The 19th British HCI Group Annual Conference*.

Autonomously Communicating Conceptual Knowledge Through Visual Art

Derrall Heath, David Norton, Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
dheath@byu.edu, dnorton@byu.edu, ventura@cs.byu.edu

Abstract

In visual art, the communication of meaning or intent is an important part of eliciting an aesthetic experience in the viewer. Building on previous work, we present three additions to DARCI that enhances its ability to communicate concepts through the images it creates. The first addition is a model of semantic memory based on word associations for providing meaning to concepts. The second addition composes universal icons into a single image and renders the image to match an associated adjective. The third addition is a similarity metric that maintains recognizability while allowing for the introduction of artistic elements. We use an online survey to show that the system is successful at creating images that communicate concepts to human viewers.

Introduction

DARCI (Digital ARtist Communicating Intention) is a system for generating original images that convey meaning. The system is part of ongoing research in the subfield of computational creativity, and is inspired by other artistic image generating systems such as AARON (McCorduck 1991) and The Painting Fool (Colton 2011). Central to the design philosophy of DARCI is the notion that the communication of meaning in art is a necessary part of eliciting an aesthetic experience in the viewer (Csíkzentmihályi and Robinson 1990). DARCI is unique from other computationally creative systems in that DARCI creates images that explicitly express a given concept.

DARCI is composed of two major subsystems, an *image analysis* component, and an *image generation* component. The image analysis component learns how to annotate images with adjectives by training a series of neural networks with labeled images. The specific inputs to these neural networks, called *appreciation networks*, are global features extracted from each image, including information about the general occurrence of color, lighting, and texture in the images (Norton, Heath, and Ventura 2010). The image generation component uses a genetic algorithm, governed partly by the analysis component, to render a *source image* to visually convey an adjective (Norton, Heath, and Ventura 2011). While often effective, excessive filtering and extreme parameters can leave the source image unrecognizable.

In this paper we introduce new capabilities to DARCI—primarily, the ability to produce original source images rather than relying upon pre-existing, human-provided images. DARCI composes these original source images as a collage of iconic concepts in order to express a range of concepts beyond adjectives, similar to a recently introduced system for The Painting Fool that creates collages from the text of web documents (Krzeczkowska et al. 2010). However, in contrast to that system, ours creates collages from conceptual icons discovered with a semantic memory model. The resulting source images are then rendered according to an adjective discovered with this same semantic memory model. In order to preserve the content of the collages after rendering them, we introduce a variation on DARCI's traditional image rendering technique. Figure 1 outlines the two major components and their interaction, including the new elements presented in this paper. By polling online volunteers, we show that with these additions, DARCI is capable of creating images that convey selected concepts while maintaining the aesthetics achieved with filters.

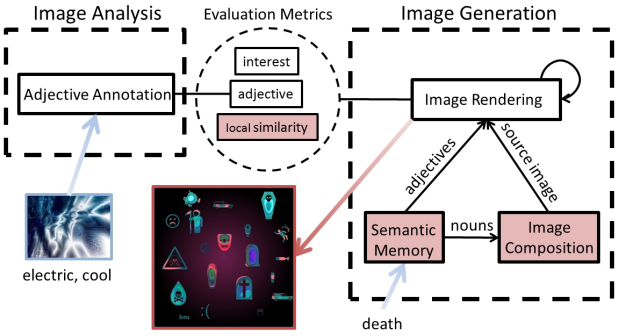


Figure 1: A diagram outlining the two major components of DARCI. *Image analysis* learns how to annotate new images with adjectives using a series of *appreciation networks* trained with labeled images. *Image generation* uses a *semantic memory* model to identify nouns and adjectives associated with a given concept. The nouns are composed into a source image that is rendered to reflect the adjectives, using a genetic algorithm that is governed by a set of evaluation metrics. The final product is an image that reflects the given concept. Additions from this paper are highlighted.

Methodology

Here we introduce the improvements to DARCI that enhance the system's capability to communicate intended meaning in an aesthetic fashion: a semantic memory model for broadening the range of concepts the system can communicate, an image composer for composing concrete representations of concepts into source images to be rendered, and a new metric for governing the evolution of the rendering process. We also describe an online survey that we use to evaluate the success of these additions.

Semantic Memory Model

In cognitive psychology, the term *semantic memory* refers to the memory of meaning and other concept-based knowledge that allows people to consciously recall general information about the world. It is often argued that creativity requires intention (and we are certainly in this camp). In this context, we mean creativity in communicating a concept, and at least one part of this can be accommodated by an internal knowledge of the concept (i.e, a semantic memory).

The question of what gives words (or concepts) meaning has been debated for years; however, it is commonly agreed that a word, at least in part, is given meaning by how the word is used in conjunction with other words (i.e., its context) (Erk 2010). Many computational models of semantic memory consist of building associations between words (Sun 2008; De Deyne and Storms 2008), and these word associations essentially form a large graph that is typically referred to as a *semantic network*. Associated words provide a level of meaning to a concept (word) and can be used to help convey its meaning.

Word associations are commonly acquired in one of two ways: from people and automatically by inferring them from a corpus. Here we describe a computational model of semantic memory that combines human free association norms with a simple corpus-based approach. The idea is to use the human word associations to capture general knowledge and then to fill in the gaps using the corpus method.

Lemmatization and Stop Words In gathering word associations, we use the standard practice of removing stop words and lemmatizing. The latter process is accomplished using WordNet's (Fellbaum 1998) database of word forms; it should be noted, however, that lemmatization with WordNet has its limits. For example, we cannot lemmatize a word across different parts of speech. As a result, words like 'redeem' and 'redeeming' will remain separate concepts because 'redeeming' could be the gerund form of the verb 'redeem' or it could be an adjective (i.e., the act of 'a redeeming quality').

Free Association Norms One of the most common means of gathering word associations from people is through *Free Association Norms* (FANs), which is done by asking hundreds of human volunteers to provide the first word that comes to mind when given a cue word. This technique is able to capture many different types of word associations including word co-ordination (pepper, salt), collocation (trash, can), super-ordination (insect, butterfly), synonymy (starving, hungry), and antonymy (good, bad). The association strength between two words is simply a count of the number of volunteers that said the second word given the first word. FANs are considered to be one of the best methods for understanding how people, in general, associate words in their own minds (Nelson, McEvoy, and Schreiber 1998). In our model we use two preexisting databases of FANs: The Edinburgh Associative Thesaurus (Kiss et al. 1973) and the University of Florida's Word Association Norms (Nelson, McEvoy, and Schreiber 1998).

Note that in this model we consider word associations to be undirected. In other words, if word A is associated with word B, then word B is associated with word A. Hence, when we encounter data in which word A is a cue for word B and word B is also a cue for word A, we combine them into a single association pair by adding their respective association strengths. Between these two databases, there are a total of 19,327 unique words and 288,069 unique associations. We refer to these associations as *human data*.

Corpus Inferred Associations Discovering word associations from a corpus is typically accomplished using a family of techniques called *Vector Space Models* (Turney and Pantel 2010), which uses a matrix for keeping track of word counts either co-occurring with other words (term $\times$ term matrix) or within each document (term $\times$ document matrix).

One of the most popular vector space models is *Latent Semantic Analysis* (LSA) (Deerwester et al. 1990), based on the idea that similar words will appear in similar documents (or contexts). LSA builds a term $\times$ document matrix from a corpus and then performs Singular Value Decomposition (SVD), which essentially reduces the large sparse matrix to a low-rank approximation of that matrix along with a set of vectors, each representing a word (as well as a set of vectors for each document). These vectors also represent points in semantic space, and the closer words are to each other in this space, the closer they are in meaning (and the stronger the association between words).

Another popular method is the *Hyperspace Analog to Language* (HAL) model (Lund and Burgess 1996). This model is based on the same idea as LSA, except the notion of context is reduced more locally to a word co-occurrence window of ± 10 words instead of an entire document. Thus, the HAL model builds a term $\times$ term matrix of word co-occurrence counts from a corpus. HAL then uses the co-occurrence counts directly as vectors representing each word in semantic space. The size of the term $\times$ term matrix is invariant to the size of the corpus and has been argued to be more congruent to human cognition than the term $\times$ document matrix used in LSA (Wandmacher, Ovchinnikova, and Alexandrov 2008; Burgess 1998).

The corpus component of our model is constructed similarly to HAL but with some important differences. We restrict the model to the same number of unique words as the human-generated free associations, building a 19,327 $\times$ 19,327 (term $\times$ term) co-occurrence matrix M using a co-occurrence window of ± 50 words. To account for the fact that common words will have generally higher co-occurrence counts, we scale these counts by weighting each element of the matrix by the inverse of the total frequency

of both words at each element. This is done by considering each element $M_{i,j}$, then adding the total number of occurrences of each word (i and j), subtracting out the value at $M_{i,j}$ (to avoid counting it twice), then dividing $M_{i,j}$ by this computed number, as follows:

$$M_{i,j} \leftarrow \frac{M_{i,j}}{(\sum_i M_{i,j} + \sum_j M_{i,j} - M_{i,j})} \quad (1)$$

The result could be a very small number, and therefore we then also normalize the values between 0 and 1.

For our corpus we use Wikipedia, as it is large, easily accessible, and covers a wide range of human knowledge (Denoyer and Gallinari 2006). Once the co-occurrence matrix is built from the entire text of Wikipedia, we use the weighted/normalized co-occurrence values themselves as association strengths between words. This approach works, since we only care about the strongest associations between words, and it allows us to reduce the number of irrelevant associations by ignoring any word pairs with a co-occurrence count less than some threshold. We chose a threshold of 100 (before weighting), which provides a good balance of producing a sufficient number of associations, while reducing the number of irrelevant associations. When looking up a particular word, we return the top n other words with the highest weighted/normalized co-occurrence values. This method, which we will call *corpus data* from now on, gives a total of 4,908,352 unique associations.

Combining Word Associations Since each source (human and corpus) provide different types of word associations, a combination of these methods into a single model has the potential to take advantage of the strengths of each method. The hypothesis is that the combined model will better communicate meaning to a person than either model individually because it presents a wider range of associations.

Our method merges the two separate databases into a single database before querying it for associations. This method assumes that the human data contains more valuable word associations than the corpus data because the human data is typically used as the gold standard in the literature. However, the corpus data does contain some valuable associations not present in the human data. The idea is to add the top n associations for each word from the corpus data to the human data but to weight the association strength low. This is beneficial for two reasons. First, if there are any associations that overlap, adding them again will strengthen the association in the combined database. Second, new associations not present in the human data will be added to the combined database and provide a greater variety of word associations. We keep the association strength low because we want the corpus data to reinforce, but not dominate, the human data.

To do this, we first copy all word associations from the human data to the combined database. Next, let W be the set of all 19,327 unique words, let $A_{i,n} \subseteq W$ be the set of the top n words associated with word $i \in W$ from the corpus data, let $score_{i,j}$ be the association strength between words i and j from the corpus data, let max_i be the maximum association score present in the human data for word i, and let θ be a weight parameter. Now for each $i \in W$ and for each $j \in A_{i,n}$, the new association score between words i and j is computed as follows:

$$score_{i,j} \leftarrow (max_i \cdot \theta) \cdot score_{i,j} \quad (2)$$

This equation scales $score_{i,j}$ (which is already normalized) to lie between 0 and a certain percentage (θ) of max_i. The n associated words from the corpus are then added to the combined database with the updated scores. If the word pair is already in the database, then the updated score is added to the score already present. For the results presented in this paper we use $n = 20$ and $\theta = 0.2$, which were determined based on preliminary experiments. After the merge, the combined database contains 443,609 associations.

Image Composer

The semantic memory model can be considered to represent the meaning of a word as a (weighted) collection of other words. DARCI effectively makes use of this collection as a decomposition of a (high-level) concept into simpler concepts that together represent the whole, the idea being that in many cases, if a (sub)concept is simple enough, it can be represented visually with a single icon (e.g., the concept 'rock' can be visually represented with a picture of a 'rock'). Given such collection of iconic concepts, DARCI composes their visual representations (icons) into a single image. The image is then rendered to match some adjective associated with the original (collective) concept.

To represent these "simple enough" concepts, DARCI makes use of a collection of icons provided by *The Noun Project*, whose goal is to build a repository of symbols/icons that can be used as a visual language (Thomas et al. 2013). The icons are intended to be simple visual representations of any noun and are published by various artists under the Creative Commons license. Currently, The Noun Project provides 6,334 icons (each 420 $\times$ 420 pixels) representing 2,535 unique nouns and is constantly growing.

When given a concept, DARCI first uses the semantic memory model to retrieve all words associated with the given concept, including itself. These word associations are filtered by returning only nouns for which DARCI has icons and adjectives for which DARCI has appreciation networks. The nouns are sorted by association strength and the top 15 are kept. For each noun, multiple icons are usually available and one or two of these icons are are chosen at random to create a set of icons for use in composing the image.

The icons in the set are scaled to between 25% and 100% of their original size according to their association strength rank. Let I be the set of icons, and let $r : I \rightarrow [0, |I| - 1]$ be the rank of icon $i \in I$, where the icon with rank 0 corresponds to the noun with the highest association strength. Finally, let ϕ_i be the scaling factor for icon i, which is computed as follows:

$$\phi_i \leftarrow 1 - \frac{0.75}{|I| - 1} r(i) \quad (3)$$

An initial blank white image of size 2000 $\times$ 2000 pixels is created and the set of scaled icons are drawn onto the blank

image at random locations, the only constraints being that no icons are allowed to overlap and no icons are allowed to extend beyond the border of the image. The result is a collage of icons that represents the original concept. DARCI then randomly selects an adjective from the set returned by the semantic memory model weighted by each adjective's association strength. DARCI uses its adjective rendering component, described in prior work, to render the collage image according to the selected adjective (Norton, Heath, and Ventura 2011; 2013; Heath, Norton, and Ventura 2013). The final image will both be artistic and in some way communicate the concept to the viewer. Figure 1 shows how this process is incorporated into the full system.

Similarity Metric

To render an image, DARCI uses a genetic algorithm to discover a combination of filters that will render a source image (in this case, the collage) to match a specified adjective. The fitness function for this process combines an *adjective metric* and an *interest metric*. The former measures how effectively a potential rendering, or *phenotype*, communicates the adjective, and the latter measures the "difference" between the phenotype and the source image. Both metrics use only global image features and so fail to capture important local image properties correlated with image content.

In this paper we introduce a third metric, *similarity*, that borrows from the growing research on *bag-of-visual-word* models (Csurka et al. 2004; Sivic et al. 2005) to analyze local features, rather than global ones. Typically, these interest points are those points in an image that are the most surprising, or said another way, the least predictable. After an interest point is identified, it is described with a vector of features obtained by analyzing the region surrounding the point. *Visual words* are quantized local image features. A dictionary of visual words is defined for a domain by extracting local interest points from a large number of representative images and then clustering them (typically with k-means) by their features into n clusters, where n is the desired dictionary size. With this dictionary, visual words can be extracted from any image by determining which clusters the image's local interest points belong. A bag-of-visual-words for the image can then be created by organizing the visual word counts for the image into a fixed vector. This model is analogous to the bag-of-words construct for text documents in natural language processing.

For the new *similarity metric*, we first create a bag-of-visual-words for the source image and each phenotype, and then calculate the Euclidean distance between these two vectors. This metric has the effect of measuring the number of interest points that coincide between the two images.

We use the standard SURF (Speeded-Up Robust Features) detector and descriptor to extract interest points and their features from images (Bay et al. 2008). SURF quickly identifies interest points using an approximation of the difference of Gaussians function, which will often identify corners and distinct edges within images. To describe each interest point, SURF first assigns an orientation to the interest point based on surrounding gradients. Then, relative to this orientation, SURF creates a 64 element feature vector by summing both the values and magnitudes of Haar wavelet responses in the horizontal and vertical directions for each square of a four by four grid centered on the point.

We build our visual word dictionary by extracting these SURF features from the database of universal icons mentioned previously. The 6334 icons result in more than two hundred thousand interest points which are then clustered into a dictionary of 1000 visual words using Elkan k-means (Elkan 2003). Once the Euclidean distance, d, between the source image's and the phenotype's bags-of-visual-words is calculated, the metric, S, is calculated to provide a value between 0 and 1 as follows:

$$S = MAX(\frac{d}{100}, 1)$$

where the constant 100 was chosen empirically.

Online Survey

Since our ultimate goal is a system that can create images that both communicate intention and are aesthetically interesting, we have developed a survey to test our most recent attempts at conveying concepts while rendering images that are perceived as creative.

The survey asks users to evaluate images generated for ten concepts across three rendering techniques. The ten concepts were chosen to cover a variety of abstract and concrete topics. The abstract concepts are 'adventure', 'love', 'music', 'religion', and 'war'. The concrete concepts are 'bear', 'cheese', 'computer', 'fire', and 'garden'.

We refer to the three rendering techniques as *unrendered*, *traditional*, and *advanced*. For *unrendered*, no rendering is applied—these are the plain collages. For the other two techniques, the images are rendered using one of two fitness functions to govern the genetic algorithm. For *traditional*, the fitness function is the average of the adjective and interest metrics. For *advanced* rendering, the new similarity metric is added. Here the adjective metric is weighted by 0.5, while the interest and similarity metrics are each weighted by 0.25. For each rendering technique and image, DARCI returned the 40 highest ranking images discovered over a period of 90 generations. We then selected from the pools of 40 for each concept and technique, the image that we felt best conveyed the intended concept while appearing aesthetically interesting. An example image that we selected from each rendering technique can be seen in Figure 2.

To query the users about each image, we followed the survey template that we developed previously to study the perceived creativity of images rendered with different adjectives (Norton, Heath, and Ventura 2013). In this study, we presented users with six five-point Likert items (Likert 1932) per image; volunteers were asked how strongly they agreed or disagreed (on a five point scale) with each statement as it pertained to one of DARCI's images. The six statements we used were (abbreviation of item in parentheses):

I like the image. (*like*)
I think the image is novel. (*novel*)
I would use the image as a desktop wallpaper. (*wallpaper*)
Prior to this survey, I have never seen an image like this one. (*never seen*)
I think the image would be difficult to create. (*difficult*)
I think the image is creative. (*creative*)

(a) unrendered (b) traditional (c) advanced

Figure 2: Example images[1] for the three rendering techniques representing the concept ‘garden’.

(a) unrendered (b) traditional (c) advanced

Figure 3: Example dummy images[2] for the concept ‘water’ that appeared in the survey for the indicated rendering techniques.

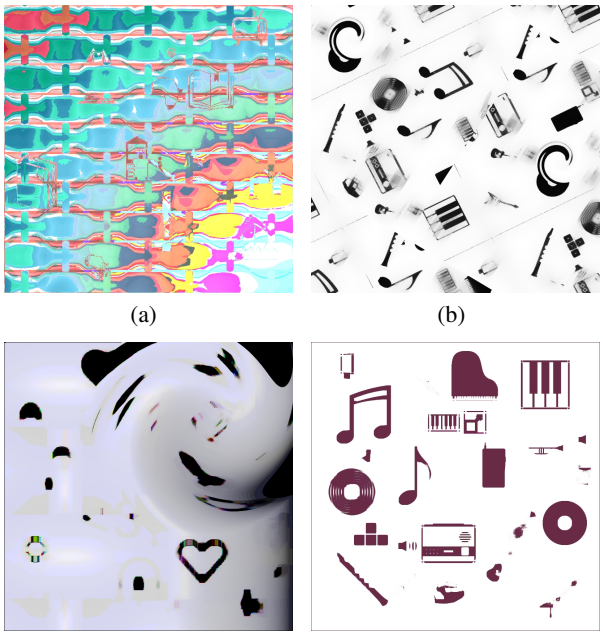
In previous work, we showed that the first five statements correlated strongly with the sixth, “I think the image is creative” (Norton, Heath, and Ventura 2013), justifying this test as an accurate evaluation of an image’s subjective creativity. In this paper, we use the same six Likert items and add a seventh to determine how effective the images are at conveying their intended concept:

I think the image represents the concept of “____.” (*concept*)

To avoid fatigue, volunteers were only presented with images from one of the three rendering techniques mentioned previously. The technique was chosen randomly and then the images were presented to the user in a random order. To help gauge the results, three dummy images were introduced into the survey for each technique. These dummy images were created for arbitrary concepts and then assigned different arbitrary concepts for the survey so that the image contents would not match their label. Unfiltered dummy collages were added to the unrendered set of images, while traditionally rendered versions were added to the traditional and advanced sets of images. The three concepts used to generate the dummy images were: ‘alien’, ‘fruit’, and ‘ice’. The three concepts that were used to describe these images in the survey were respectively: ‘restaurant’, ‘water’, and ‘freedom’. To avoid confusion, from here on we will always refer to these dummy images by their description word. The dummy images for the concept of ‘water’ are shown in Figure 3. In total, each volunteer was presented with 13 images.

[1] The original icons used for the images in Figure 2 were designed by Adam Zubin, Birdie Brain, Evan Caughey, Rachel Fisher, Prerak Patel, Randall Barriga, dsathiyaraj, Jeremy Bristol, Andrew Fortnum, Markus Koltringer, Bryn MacKenzie, Hernan Schlosman, Maurizio Pedrazzoli, Mike Endale, George Agpoon, and Jacob Eckert of The Noun Project.

[2] The original icons used for the images in Figure 3 were designed by Alessandro Suraci, Anna Weiss, Riziki P.M.G. Nielsen, Stefano Bertoni, Paulo Volkova, James Pellizzi, Christian Michael Witternigg, Dan Christopher, Jayme Davis, Mathies Janssen, Pavel Nikandrov, and Luis Prado of The Noun Project.



(a) (b) (c) (d)

Figure 4: The images[3] that were rated the highest on average for each statement. Image (a) is the advanced rendering of ‘adventure’ and was rated highest for *like*, *novel*, *difficult*, and *creative*. Image (b) is the traditional rendering of ‘music’ and was rated highest for *wallpaper*. Image (c) is the advanced rendering of ‘love’ and was rated highest for *never seen*. Image (d) is the advanced rendering of ‘music’ and was rated highest for *concept*.

Results

A total of 119 anonymous individuals participated in the online survey. Volunteers could quit the survey at anytime, thus not evaluating all 13 images. Each person evaluated an average of 9 images and each image was evaluated by an average of 27 people. The highest and lowest rated images for each question can be seen in Figures 4 and 5 respectively.

The three dummy images for each rendering technique are used as a baseline for the concept statement. The results of the dummy images versus the valid images are show in Figure 6. The average concept rating for the valid images is significantly better than the dummy images, which shows that the intended meaning is successfully conveyed to human viewers more reliably than an arbitrary image. These results confirm that the intelligent use of iconic concepts is beneficial for the visual communication of meaning. Further, it is suggestive that the ratings for the other statements are generally lower for the dummy images than for the valid

[3] The original icons used for the images in Figure 4 were designed by Oxana Devochkina, Kenneth Von Alt, Paul te Kortschot, Marvin Kutscha, James Fenton, Camilo Villegas, Gustavo Perez Rangel, and Anuar Zhumaev of The Noun Project.

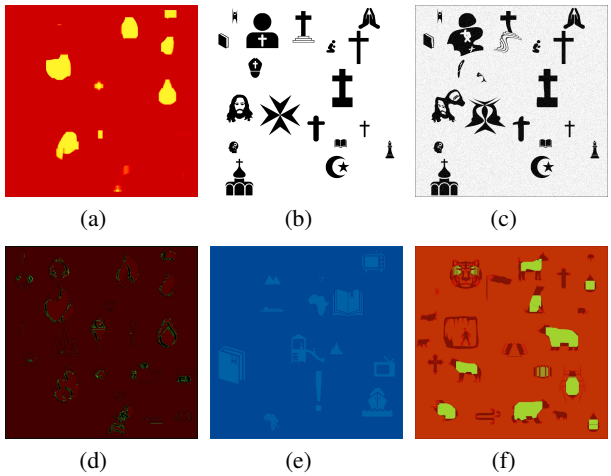


Figure 5: The images[4] that were rated the lowest on average for each statement. Image (a) is the advanced rendering of 'fire' and was rated lowest for *difficult* and *creative*. Images (b) and (c) are the unrendered and advanced version of 'religion' and were rated lowest for *neverseen* and *wallpaper* respectively. Images (d), (e), and (f) are the traditional renderings of 'fire', 'adventure', and 'bear', respectively, and were rated lowest for *like*, *novel*, and *concept* respectively.

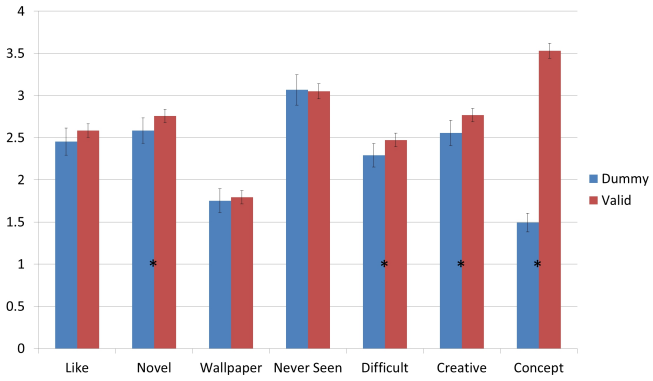


Figure 6: The average rating from the online survey for all seven statements comparing the dummy images with the valid images. The valid images were more successful at conveying the intended concept than the dummy images by a significant margin. Results marked with an asterix (*) indicate statistical significance using the two tailed independent t-test. The lines at the top of each bar show the 95% confidence interval for each value. The sample sizes for dummy and valid images are 251 and 818 respectively.

images. Since the the dummy images were created for a different concept than the one which they purport to convey in the survey, this may be taken as evidence that successful conceptual or intentional communication is an important factor for the attribution of creativity.

The results of the three rendering techniques (unrendered, traditional, and advanced) for all seven statements are shown in Figure 7. The unrendered images are generally the most successful at communicating the intended concepts. This is likely because the objects/icons in the unrendered images are left undisturbed and are therefore more clear and discernible, requiring the least perceptual effort by the viewer. The rendered images (traditional and advanced) often distort the icons in ways that make them less cohesive and less discernible and can thus obfuscate the intended meaning. The trade-off, of course, is that the unrendered images are generally considered less likable, less novel, and less creative than the rendered images. The advanced images are generally considered more novel and creative than the traditional images, but the traditional images are liked slightly more. The advanced images also convey the intended meaning more reliably than the traditional images, which indicates that the similarity metric is finding a better balance between adding artistic elements and maintaining icon recognizability.

The difference between the traditional and advanced rendering was minimized by the fact that we selected the image (out of DARCI's top 40) from each group that best conveyed the concept while also being aesthetically interesting. Out of all the traditional images, 39% had at least one recognizable icon, while 74% of the advanced images had at least one recognizable icon. This difference demonstrates that the new similarity metric helps to preserve the icons and provides a greater selection of good images from which to choose, which is consistent with the results of the survey. For comparison, Figure 8 shows some example images (both traditional and advanced) that were not chosen for the survey.

The results comparing the abstract concepts with the concrete concepts are shown in Figure 9. For all seven statements, the abstract concepts are, on average, rated higher than the concrete concepts. One possible reason for this is that concrete concepts are not easily decomposed into a collection of iconic concepts because, being concrete, they are more likely to be iconic themselves. For concrete concepts, the nouns returned by the semantic memory model are usually other related concrete concepts, and it becomes difficult to tell which object is the concept in question. For example, the concept 'bear' returns nouns like 'cave', 'tiger', 'forest', and 'wolf', which are all related, but don't provide much indication that the intended concept is 'bear'. A person might be inclined to generalize to a concept such as 'wildlife'. Another possible reason why abstract concepts result in better survey results than do concrete concepts is because abstract concepts allow a wider range of interpretation and are generally more interesting. For example, the concept 'cheese' would generally be considered straightforward to most people, while the concept 'love' could have variable meanings to different people in different circumstances. Hence, the

[4] The original icons used for the images in Figure 5 were designed by Melissa Little, Dan Codyre, Carson Wittenberg, Kenneth Von Alt, Nicole Kathryn Griffing, Jenifer Cabrera, Renee Ramsey-Passmore, Ben Rex Furneaux, Factorio.us collective, Anuar Zhumaev, Luis Prado, Ahmed Hamzawy, Michael Rowe, Matthias Schmidt, Jule Steffen, Monika Ciapala, Bru Rakoto, Patrick Trouv, Adam Heller, Marco Acri, Mehmet Yavuz, Allison Dominguez, Dan Christopher, Nicholas Burroughs, Rodny Lobos, and Norman Ying of The Noun Project.

[5] The original icons used for the images in Figure 8 are the same as those used in Figures 4 and 5 with attribution to the same designers.

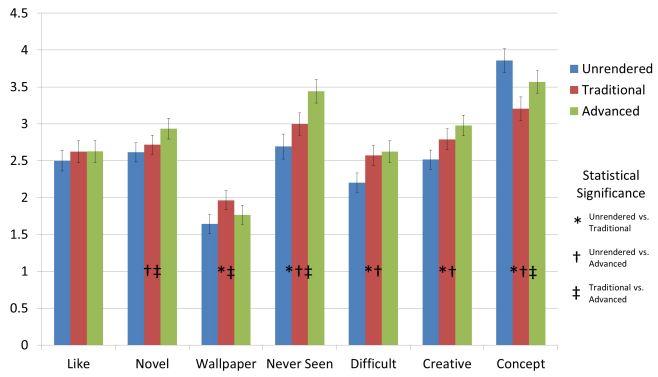


Figure 7: The average rating from the online survey for all seven statements comparing the three rendering techniques. The unrendered technique is most successful at representing the concept, while the advanced technique is generally considered more novel and creative. Statistical significance was calculated using the two tailed independent t-test. The lines at the top of each bar show the 95% confidence interval for each value. The sample sizes for the unrendered, traditional, and advanced techniques are 256, 285, and 277 respectively.

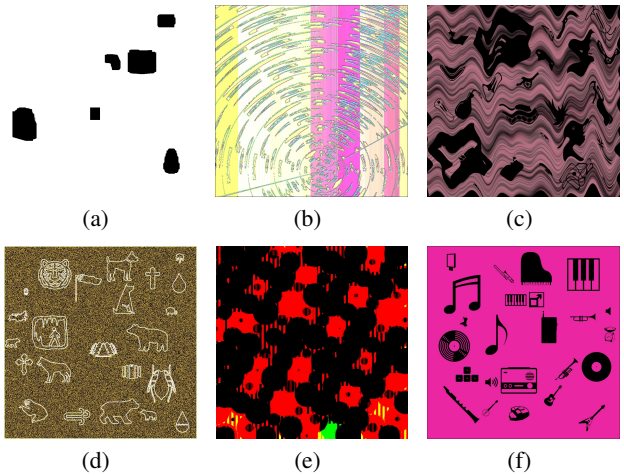


Figure 8: Sample images[5] that were not chosen for the online survey. Images (a), (b), and (c) are traditional renderings of 'adventure', 'love', and 'war' respectively. Images (d), (e), and (f) are advanced renderings of 'bear', 'fire', and 'music' respectively.

images generated for abstract concepts are generally considered more likable, more novel, and more creative than the concrete images.

Conclusions and Future Work

We have presented three additions to the computer system, DARCI, that enhance the system's ability to communicate specified concepts through the images it creates. The first addition is a model of semantic memory that provides conceptual knowledge necessary for determining how to compose and render an image by allowing the system to make decisions and reason (in a limited manner) about common world knowledge. The second addition uses the word associations from a semantic memory model to retrieve conceptual icons and composes them into a single image, which is then rendered in the manner of an associated adjective. The third addition is a new similarity metric used during the adjective rendering phase that preserves the discernibility of the icons while allowing for the introduction of artistic elements.

We used an online survey to evaluate the system and show that DARCI is significantly better at expressing the meaning of concepts through the images it creates than an arbitrary image. We show that the new similarity metric allows DARCI to find a better balance between adding interesting artistic qualities and keeping the icons/objects recognizable. We show that using word associations and universal icons in an intelligent way is beneficial for conveying meaning to human viewers. Finally, we show that there is some degree of correlation between how well an image communicates the intended concept and how well liked, how novel, and how creative the image is considered to be. To further illustrate DARCI's potential, Figure 10 shows additional images encountered during various experiments with DARCI that we thought were particularly interesting.

In future research we plan to do a direct comparison of the images created by DARCI with images created by human artists and to further investigate how semantic memory contributes to the creative process. We plan to improve the semantic memory model by going beyond word-to-word associations and building associations between words and other objects (such as images). This will require expanding DARCI's image analysis capability to include some level of image noun annotation. The similarity metric presented in this paper is a step in that direction. An improved semantic memory model could also help enable DARCI to discover its own topics (i.e., find its own inspiration) and to compose icons together in more meaningful ways, by intentional choice of absolute and relative icon placement, for example.

References

Bay, H.; Ess, A.; Tuytelaars, T.; and Gool, L. V. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110:346–359.

Burgess, C. 1998. From simple associations to the building blocks of language: Modeling meaning in memory with the HAL model. *Behavior Research Methods, Instruments, & Computers* 30:188–198.

Colton, S. 2011. The Painting Fool: Stories from building an automated painter. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Springer-Verlag.

Csíkzentmihályi, M., and Robinson, R. E. 1990. *The Art of Seeing*. The J. Paul Getty Trust Office of Publications.

Csurka, G.; Dance, C. R.; Fan, L.; Willamowski, J.; and Bray, C. 2004. Visual categorization with bags of keypoints.

[6]The original icons used for the images in Figure 10 were designed by Alfredo Astort, Simon Child, Samuel Eidam, and Jonathan Keating of The Noun Project.

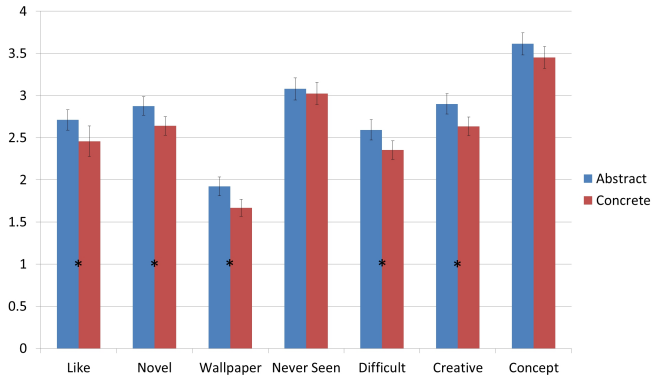
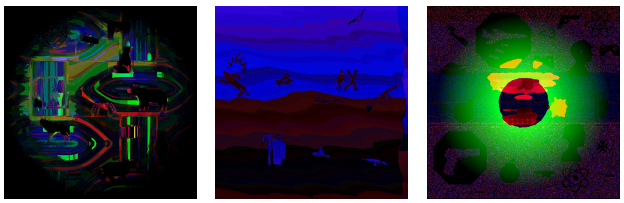


Figure 9: The average rating from the online survey for all seven statements comparing the abstract concepts with the concrete concepts. The abstract concepts generally received higher ratings for all seven statements. Results marked with an asterix (*) indicate statistical significance using the two tailed independent t-test. The lines at the top of each bar show the 95% confidence interval for each value. The sample sizes for abstract and concrete concepts are 410 and 408 respectively.



(a) bear (b) murder (c) war

Figure 10: Notable images[6] rendered by DARCI during various experiments and trials.

In *Proceedings of the Workshop on Statistical Learning in Computer Vision*, 1–22.

De Deyne, S., and Storms, G. 2008. Word associations: Norms for 1,424 Dutch words in a continuous task. *Behavior Research Methods* 40(1):198–205.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Denoyer, L., and Gallinari, P. 2006. The Wikipedia XML corpus. In *INEX Workshop Pre-Proceedings*, 367–372.

Elkan, C. 2003. Using the triangle inequality to accelerate k-means. In *Proceedings of the Twentieth International Conference on Machine Learning*, 147–153.

Erk, K. 2010. What is word meaning, really?: (and how can distributional models help us describe it?). In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, 17–26. Stroudsburg, PA, USA: Association for Computational Linguistics.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Heath, D.; Norton, D.; and Ventura, D. 2013. Conveying semantics through visual metaphor. *ACM Transactions of Intelligent Systems and Technology, to appear*.

Kiss, G. R.; Armstrong, C.; Milroy, R.; and Piper, J. 1973. An associative thesaurus of English and its computer analysis. In Aitkin, A. J.; Bailey, R. W.; and Hamilton-Smith, N., eds., *The Computer and Literary Studies*. Edinburgh, UK: University Press.

Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation — with intent. In *Proceedings of the 1st International Conference on Computational Creativity*, 36–40.

Likert, R. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 22(140):1–55.

Lund, K., and Burgess, C. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers* 28:203–208.

McCorduck, P. 1991. *AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. W. H. Freeman & Co.

Nelson, D. L.; McEvoy, C. L.; and Schreiber, T. A. 1998. The University of South Florida word association, rhyme, and word fragment norms. http://www.usf.edu/FreeAssociation/.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *Proceedings of the 1st International Conference on Computational Creativity*, 26–35.

Norton, D.; Heath, D.; and Ventura, D. 2011. Autonomously creating quality images. In *Proceedings of the 2nd International Conference on Computational Creativity*, 10–15.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior, to appear*.

Sivic, J.; Russell, B. C.; Efros, A. A.; Zisserman, A.; and Freeman, W. T. 2005. Discovering objects and their location in images. *International Journal of Computer Vision* 1:370–377.

Sun, R. 2008. *The Cambridge Handbook of Computational Psychology*. New York, NY, USA: Cambridge University Press, 1st edition.

Thomas, S.; Boatman, E.; Polyakov, S.; Mumenthaler, J.; and Wolff, C. 2013. The noun project. `http://thenounproject.com`.

Turney, P. D., and Pantel, P. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37:141–188.

Wandmacher, T.; Ovchinnikova, E.; and Alexandrov, T. 2008. Does latent semantic analysis reflect human associations? In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, 63–70.

A Computer Model for the Generation of Visual Compositions

Rafael Pérez y Pérez[1], María González de Cossío[1], Iván Guerrero[2]
División de Ciencias de la Comunicación y Diseño[1]
Universidad Autónoma Metropolitana, Cuajimalpa, México D. F.
Posgrado en Ciencia e Ingeniería de la Computación[2]
Universidad Nacional Autónoma de México
{rperez/mgonzalezc}@correo.cua.uam.mx; cguerreror@uxmcc2.iimas.unam.mx

Abstract

This paper describes a computer model for visual compositions. It formalises a series of concepts that allows a computer agent to progress a visual work. We implemented a prototype to test the model; it employs letters from the alphabet to create its compositions. The knowledge base was built from examples provided by designers. From these examples the system obtained the necessary information to produce novel compositions. We asked a panel of experts to evaluate the material produced by our system. The results suggest that we are in the right track although much more work needs to be done.

Introduction

This text reports a computer model for visual compositions. The following lines describe the motivation behind it. One of the most important topics that a student in design needs to master is that related to visual composition. By composition we refer to the way in which elements in a graphic work are organised on the canvas. The design process of a composition implies the selection, planning and conscious organisation of visual elements that aim to communicate (Myers 1989; Deepak 2010). Compositions can be very complex with several elements interacting in diverse ways.

Unfortunately, an important number of design texts include what we called "unclear" explanations about composition and its characteristics; in many cases, they are based on personal appreciations rather than on more objective criteria. To illustrate our point, here are descriptions of the concept of visual balance found in some design texts: "Psychologically we cannot stand a state of imbalance for very long. As time passes, we become increasingly fearful, uncomfortable, and disoriented" (Myers 1989: 85); "The formal quality in symmetry imparts an immediate feeling of permanence, strength, and stability. Such qualities are important in public buildings to suggest the dignity and power of a government" (Lauer and Pentak 2012: 92); "exacting, noncasual and quiet, but can also be boring" (Brainard 1991:96). Similar definitions can be found in Germani-Fabris (1973); Faimon and Weigand (2004); Fullmer (2012); and so on. As one can see there is a need for clearer explanations that can guide designers, teachers and students on these topics.

We believe that computer models of creativity are very useful tools that can contribute to formalize this type of concepts and, hopefully, to make them more accessible and clearer to students and the general public. Therefore, the purpose of this project is to develop a computer model of visual composition and implement a prototype. Particularly, we are interested in representing the genesis of the visual composition process; c.f. with other computer models that represent more elaborated pieces of visual works like ERI-Designer (Pérez y Pérez et al. 2010), The Painting Fool (Colton 2012), DARSY (Norton et al. 2011). Related works also include shape grammars (Stiny 1972) and relational production systems (Vere 1977, 1978). Other interesting approaches are those based in evolutionary mechanism (e.g. Goldberg 1991; Bentley 1999). However, we are interested in understanding each step in the composition process rather than look for optimization processes.

This paper is organised as follows: section 2 describes some characteristics that we consider essential in visual composition; section 3 describes the core aspects of our model; section 4 describes the core characteristics of our prototype and how we used it to test our model; section 5 discusses the results we obtained.

Characteristics of a Composition

Composition is a very complex process that usually involves several features and multiple relations between them. It is out of the scope of this project to attempt to represent the whole elements involved in a composition.

A composition is integrated by design elements and by design principles. The design elements are dots, lines, colours, textures, shapes and planes that are placed on a canvas. The design principles are the way these elements relate to

each other and to the canvas. The principles that we employ in this project are rhythm, balance and symmetry.

Rhythm is the regular repetition of elements. For regular repetition we mean that the distance between adjacent elements is constant. Groups of repeated elements make patterns. The frequency of a pattern describes how many times the same element is repeated within a given area in a canvas. Thus, the frequency depends on the size and distance between elements. A composition might include two or more patterns with the same or different frequencies.

Balance is related to the distribution of visual elements on the canvas. If there is an equal distribution on both sides of the canvas, there is a formal balance. If the elements are not placed with equal distribution, there is an informal balance. Myers describes informal balance as

> "Off-centre balance. It is best understood as the principle of the seesaw. Any large, 'heavy' figure must be placed closer to the fulcrum in order to balance a smaller, 'lighter' figure located on the opposite side. The fulcrum is the point of support for this balancing act. It is a physical principle transposed into a pictorial field. The fulcrum is never seen, but its presence must be strongly felt" (1989: 90).

Symmetry, (from the Greek συμμετρεῖν symmetreín), "with measure", means equal distribution of elements on both sides of the canvas. The canvas is divided into many equal areas as needed. The basic divisions separate the canvas in four areas using a vertical axis and a horizontal axis. Diagonal divisions can also be included. Symmetry can be explained as follows: "Given plane A, a figure is symmetrical in relation to it, when it reflects in A, and goes back to its initial position" (Agostini 1987:97). In other words "symmetry of a (planar) picture [is] a motion of the plane that leaves that picture unchanged" (Field 1995:41). In this project we work with three types of symmetry:

1. *Reflectional symmetry or mirror symmetry*. It refers to the reflection of an element from a central axis or mirror line. If one half of a figure is the mirror image of the other, we say that the figure has reflectional or mirror symmetry, and the line marking the division is called the line of reflection, the mirror line, or the line of symmetry (Kinsey and Moore 2002:129).
2. *Rotational symmetry*. The elements rotate around a central axis. It can be in any angle or frequency, whilst the elements share the same centre. For example, in nature, a sunflower shows each element rotating around a centre.
3. *Bilateral symmetry or translational symmetry*. Refers to equivalent elements that are placed in different locations but with the same direction. "The element moves along a line to a position parallel to the original" (Kinsey and Moore 2002:148).

Description of the Model

For this work we assume that all compositions are generated on a white canvas with a fixed size. Compositions are comprised by the following elements: blank, simple elements and compound elements, also referred to as groups. Blank is the space of the canvas that is not occupied by any element. A simple-element is the basic graphic unit employed to create a visual composition. A compound-element is a group formed by simple-elements (as it will be explained later, all adjacent elements within a group must have the same distance). A compound-element might also include other compound-elements. Once a simple-element is part of a group, it cannot participate in another group as a simple-element.

All elements have associated a set of attributes:

1. Blank has an area.
2. Simple-elements have a position (determined by the centre of the element), an orientation, an area and an inclination.
3. Compound-elements have a position, an area, a shape, a rhythm and a size. The position is calculated as the geometric centre of the element. Compound-elements can have four possible shapes: horizontal, vertical, diagonal and any other. The rhythm is defined as the constant repetition of elements. The size is defined by the number of elements (simple or compound) that comprise the group.

There are three basic primitive-actions that can be performed on simple and compound elements: insert in the canvas, eliminate from the canvas and modify its attributes.

Relations. All elements in a canvas have relations with the other elements. Our model represents three types of relations: distance, balance and symmetry.

Distance. We include four possible distances between elements:

- Lying-on: one element is on top of other element.
- Touch: the edge of one element is touching the edge of other element.
- Close: none of the previous classifications apply and the distance between the centre of element 1 and element 2 is equal or minor to a distance known as Distance of Closeness (DC). It represents that an element is close to another element. The appropriate value of DC depends on cultural aspects and might change between different societies (see Hall 1999).
- Remote: the distance between the centres of element 1and element 2 is major to DC.

Balance. We employ two different axes to calculate balance: horizontal and vertical. They all cross the centre of the canvas. The balance between two elements is obtained as follows. The area of each element is calculated and then multi-

plied by its distance to the centre. If the results are alike the elements are balanced. Unbalanced relations are not explicitly represented.

Symmetry. We work with three types of symmetry: reflectional (Rf), translational (Tr) and rotational (Rt). We employ two different axes to calculate it: horizontal (H) and vertical (V). So, two different elements in a canvas might have one of five different symmetric relations between them: horizontal-reflectional (H-Rf), vertical-reflectional (V-Rf), horizontal-translational (H-Tt), vertical-translational (V-Tt) and rotational (Rt). Asymmetrical relations are not explicitly represented.

Creation of Groups. Inspired by Gestalt studies in perception (Wertheimer 2012) in this work, groups are created based on the distance between its elements. The minimum distance (MD) is the smallest distance between two elements (e.g. if the distance between element 1 and element 2 is 1 cm, the distance between element 2 and element 3 is 3 cm, and the distance between element 1 and element 3 is 4 cm, MD is equal to 1 cm). Its value ranges from zero (when the centre of element 1 is lying on top of the centre of element 2) to DC.

$$0 \leq MD \leq DC$$

That is, inspired by Gestalt studies that indicate that the eye perceives elements that are close as a unit, a group cannot include elements with a remote distance.

The process of grouping works as follows. All simple-elements that are separated from other simple-elements by the same distance are grouped together, as long as such a distance is minor to the remote distance. If as a result of this process at least one group is created, the same process is performed again. The process is repeated until it is not possible to create more groups. Notice that this way of grouping produces that all groups have associate a rhythm, i.e. all groups include the constant repetition of (at least one) elements. We refer to the groups created during this process as Groups of Layer 1. Figure 1 layer 0 shows simple elements on a canvas before the system groups them; Figure 1 layer 1 shows the groups that emerge after performing this process: group 1 (the blue one), group 2 (the purple one) and group 3 (the yellow one); d1 represents the distance between elements in group 1; d2 represents the distance between elements in group 2; d3 represents the distance between elements in group 3. The following lines describe the algorithm:

First iteration, Layer 1

1. Considering only simple-elements find the MD value.
2. If there are not at least two simple-elements whose MD is equal or minor to DC then finish.
2. All simple-elements that are separated from other simple-elements by a distance MD form a new group.
3. Go to step 1.

Now, employing a similar mechanism, we can try to create new groups using the Groups of Layer 1 as inputs (see Figure 1 Layer 2). We refer to the groups created during this second process as Groups of Layer 2. Groups at layer 2 are comprised by simple-elements and/or compound-elements. The algorithm works as follows:

If at least one group was created during Layer 1 then perform Layer 2.

Second iteration, Layer 2

1. Considering simple and compound elements, that have not formed a group in this layer yet, find the value of the MD.
2. If there are not at least two elements whose MD is equal or minor to DC then finish.
2. All elements that are separated from other elements by a distance MD form a new group.
3. Go to step 1.

Notice how the blue group and the purple group merge; the reason is that the distance between purple group and the blue group (d21) is smaller than the distance between the blue group and the yellow group (d13), or the distance between the purple group and the yellow group (d23). Because there is no other group to merge, the yellow group has to wait until the next cycle (next layer) to be integrated (see Figure

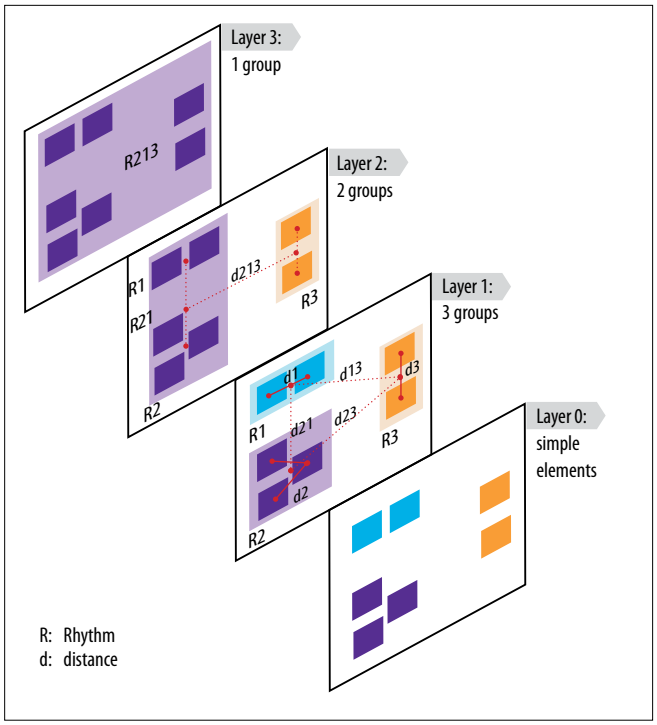


Figure 1. A composition represented by 3 layers.

1 layer 3). This process is repeated until no more layers can be created. All groups created during the first iteration are known as Groups at Layer 1; all groups created during the second iteration are known as Groups at Layer 2; all groups created during the nth iteration are known as Groups at Layer n. A composition that generates n layers is referred to as nth Layers Composition.

Calculating rhythms. The process to calculate rhythms within a composition works as follows. Each group at layer 1 has its own rhythm (see Figure 1 layer 1). So, the blue group has a rhythm 1 (R1), the purple group has a rhythm 2 (R2) and the yellow group has a rhythm 3 (R3). When the system blends the blue and purple groups, the new group includes three different rhythms (see Figure 1 Layer 2): R1, R2 and a new rhythm R21. Rhythm R21 is the result of the distance between the centre of the blue group and the centre of the purple group. We can picture groups as accumulating the rhythms of its members. So, in Figure 1 Layer 2 we can observe four rhythms: R1, R2, R21 (inside the purple group) and R3 in the yellow group. A group that includes only one rhythm is classified monotonous; a group that includes two or more rhythms is classified as varied. So, the purple blue has a varied rhythm while the yellow group has a monotonous rhythm.

Analysis of the composition. Our model represents a composition in terms of all existing relations between its elements. This representation is known as Context.

Because each layer within a composition includes different elements, and possible different relations between them, the number of contexts associated to one composition depends on its number of layers. Thus, a 3 layers composition has associated three contexts: context-layer 1, context-layer 2 and context-layer 3.

$$\text{Context of the composition} = \text{Context-layer 1} + \text{Context-layer 2} + \text{Context-layer 3}$$

Besides relationships, a context-layer also includes information about the attributes of each element, and what we refer to as the attributes of the layer: Density of the layer, Balance of the layer, Symmetry of the layer and Rhythm of the layer. The Density of the Layer (DeL) is the relation between the blank's area and all elements' area:

$$\text{Density of the Layer} = \frac{\text{All Elements' area}}{\text{Blanks' area}}$$

The Balance of the layer and Symmetry of the layer indicate if the layer as a whole is symmetrical and is balanced. The Rhythm of the layer indicates the type of rhythm that the layer has as a whole. Like in the case of the groups it can have the following values: Monotonous or Varied (see Figure 2).

Components of a context-layer
Relation between elements
Attributes of the elements
Attributes of the layer

Figure 2. Components of a context layer.

Composition process

> We can describe a composition as a process that consists on sequentially applying a set of actions, which generate several partial or incomplete works… until the right composition arises or the process is abandoned (Pérez y Pérez et al. 2010)

Thus, if we have a blank canvas and perform an action on it, we will produce an initial partial composition; if we modify that partial composition by performing another action, then we will produce a more elaborated partial composition; we can keep on repeating this process until, with some luck, we will end producing a whole composition. Thus, by performing actions we progress the composition (see Figure 3).

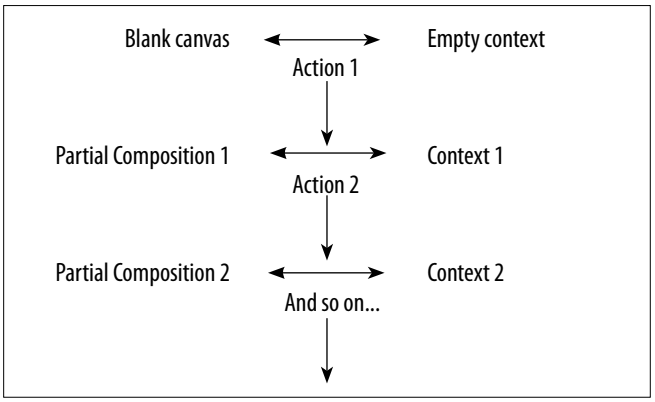


Figure 3. A composition process.

The model allows calculating for each partial composition all its contextual-layers. This information is crucial for generating novel compositions.

Producing new works

Our model includes two main processes: the generation of knowledge structures and the generation of compositions.

Generation of knowledge structures

The model requires a set of examples that are provided by human experts; we refer to them as the previous designs. So, each previous design is comprised by one or more partial compositions; each of these partial compositions is more elaborated than the previous one. At the end we have the final composition.

As explained earlier, we can picture a composition process as a progression of contexts mediated by actions until the last context is generated. In the same way, if we have the sequence of actions that leads towards a composition (and that is the type of information we can get from the set of examples), we can analyse and register how the composition process occurred. The goal is to create knowledge structures that group together a context and an action to be performed. In other words, the knowledge base is comprised by contexts (representing partial compositions) and actions to transform them in order to progress the composition.

Because the previous designs do not represent explicitly their associated actions, it is necessary to obtain them. The following lines explain how this process is done. We compare two contexts and register the differences between them. Such differences become the next action to perform. For example, if Context 1 represents an asymmetrical composition and Context 2 represents a horizontal symmetrical one, we can associate the action "make the current composition horizontally symmetrical" to Context 1 as the next action to continue the work in progress.

Once this relation has been established, it is recorded in the knowledge base as a new knowledge structure. We do the same with all the contexts in all the layers of a given partial composition. The actions that can be associated to a context are: make (reflectional, rotational or translational) symmetrical the current composition; balance (horizontally or vertically) the current composition; insert, delete or modify a simple or compound element; make (reflectional, rotational or translational) asymmetrical the current composition; unbalance (horizontally or vertically) the current composition; end the process of composition. The following lines describe the algorithm to process the previous designs.

1. Obtain the number of all the partial compositions of a given example (NumberPC)
2. Calculate all the contexts for each partial composition
3. For n:= 1 to (NumberPC – 1)
    - 3.1 Compare the differences between Context n and Context n+1
    - 3.2 Find the action that transform Context n into Context n+1
    - 3.3 Create a new knowledge structure associating Context n and the new Action
    - 3.4 Record in the knowledge base this new knowledge structure.
4. The context of the last partial composition gets the action "end of the process of composition".

We repeat the same process for each example in the set of previous designs. All the knowledge structures obtained in this way are recorded in the knowledge base. The bigger the set of previous designs the richer our knowledge base is.

Generation of compositions: The composition process follows the E-R model described in (Pérez y Pérez and Sharples 2001). The following lines describe how it works.

The E-R model has two main processes: engagement and reflection. During engagement the system generates material; during reflection such material is evaluated and, if necessary, modified. The composition is a constant cycle between engagement and reflection. The model requires an initial state, i.e. an initial partial composition to start; then, the process is triggered. The following lines describe how we defined engagement and reflection:

Engagement:

1. The system calculates all the Contexts that can be obtained from the current partial composition.
2. All these contexts are employed as cues to probe memory.
3. The system retrieves from memory all the knowledge structures that are equal or similar to the current contexts. If none structure is retrieved, an impasse is declared and the system switches to reflection.
4. The system selects one of them at random and performs its associated action. As a consequence the current partial composition is updated.
5. And the cycles repeats again (step 1).

Reflection:

1. If there is an impasse the system attempts to break it and then returns to the generation phase.
2. The system checks that the current composition satisfies the requirements of coherence (e.g. the system verifies that all the elements are within the area of the canvas; that elements are not accidentally on top of each other; and so on).
3. The system verifies the novelty of the composition in progress. A composition is novel if it is not similar to any of the compositions in the set of previous designs.

The system starts in engagement; after three actions it switches to reflection and then goes back to engagement. If during engagement an impasse is declared, the system switches to reflection to try to break it and then switches back to engagement. The cycle ends when an unbreakable impasse is triggered or when the action "end of the process of composition" is performed.

Example of a composition: For space reasons, it is impossible to describe in detail how the system creates a whole new design. Instead, in Figure 4 we show some partial compositions generated by our program and their associated contexts. To create the partial-composition in Figure 4A, the system starts

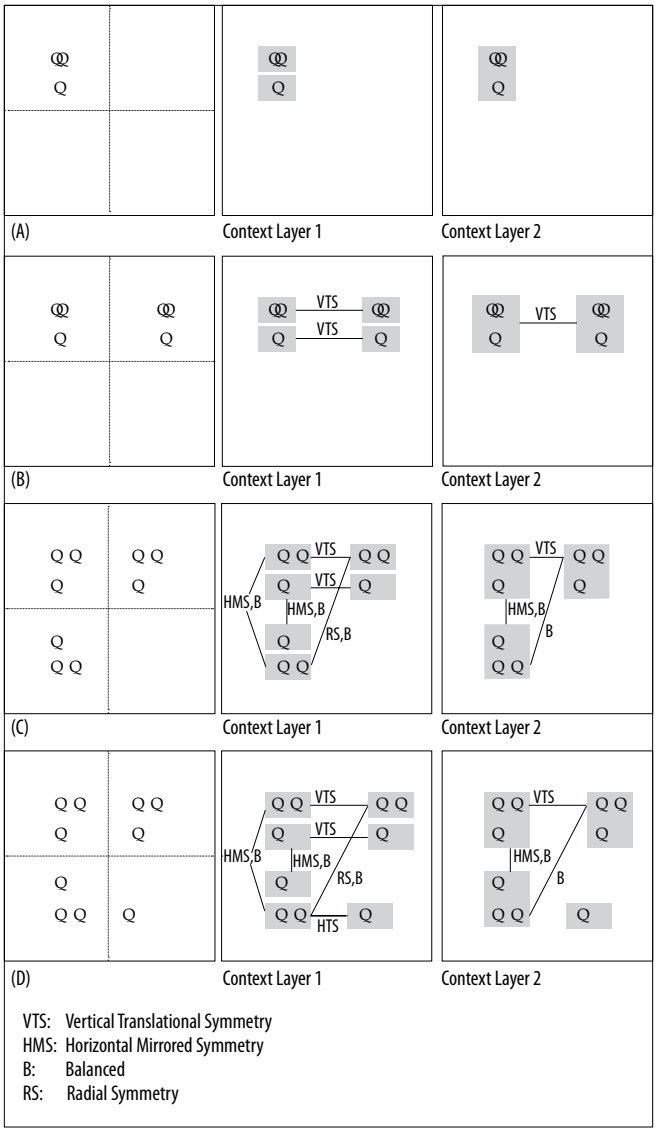


Figure 4. Partial compositions and their contexts.

with a blank canvas and then inserts three elements at random (the three elements on the top-left). This partial composition has two layers: the context of each layer is depicted on the right side of Figure 4A. For the sake of clarity the figure does not include the attributes of the elements; then, during engagement, it takes the current contexts as cues to probe memory and retrieves some actions to progress the work. Between the retrieved actions one is selected at random. So, it inserts three new elements that produce a vertical translational symmetry (see Figure 4B). The context in each layer clearly shows the relation between all elements in the canvas. In this case, in Layer 1 we have two Vertical Translational Symmetry (VTS) and in Layer 2 we have one VTS symmetry.

The system switches to reflection and realises that some elements are on top of others. Employing some heuristics to analyse the composition, the program decides that is better to separate them. The system switches back to engagement, takes the current contexts as cues to probe memory and retrieves actions to be performed. In this occasion, the system inserts in the third quadrant a new group with a horizontal mirrored symmetry (see Figure 4C). The right side of the figure shows the context at each layer. The process is repeated again generating the partial composition in Figure 4D and its corresponding contexts.

Tests and Results

We implemented a prototype to test our model. Because of the technical complexity of implementing the whole model we decided to include some constraints. In our prototype all simple-elements have the same size, colour and shape: in this work, simple elements are letters of the alphabet. Because of the technical difficulty of implementing relationships, in this prototype we only use symmetry and balance.

Like the model, the prototype has two main parts: creation of knowledge structures and generation of new compositions. The prototype has an interface that allows the user to create her own compositions. She can insert, delete or modify letters in the canvas. By clicking one button she can also build new symmetrical or balanced elements, or generate random groups. The program automatically indicates all the existing groups in all layers; it also shows all the relationships that currently exist between the elements in the canvas. In the same way, the attributes of all elements are displayed as well as their rhythms. So, the user only has to create her composition on the canvas (the program includes a partial-composition button that allows the user to indicate when a partial composition is ready). In this way, the system automatically creates the file of previous designs. Once the knowledge base is ready, the user can trigger the E-R cycle to generate novel compositions.

We provided our prototype with five previous designs; Figures 5 and 6 show two works generated by our program.

In order to obtain an external feedback we decided to ask a panel of experts their opinion about our program's work. The

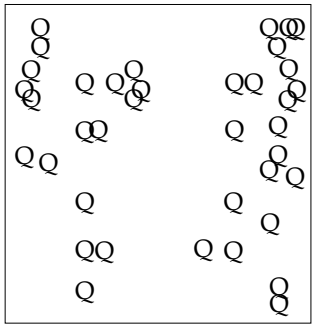


Figure 5. A composition created by our agent. It is Composition 2 in the questionnaire.

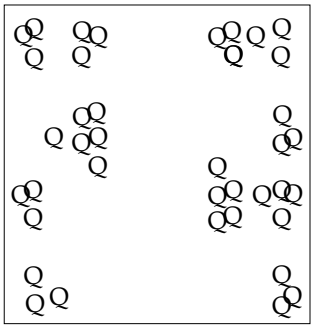


Figure 6. A second composition created by our agent. It is Composition 3 in the questionnaire.

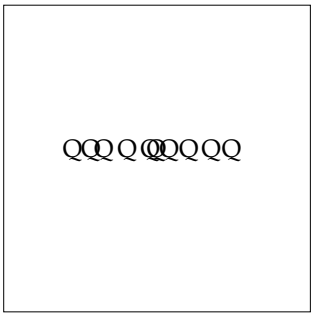


Figure 7: Human generated composition. Corresponds to composition 1 in the questionnaire.

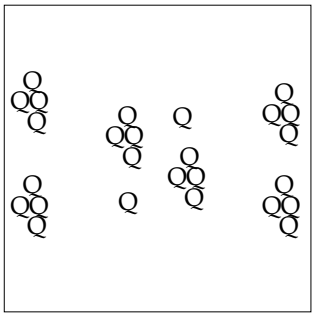


Figure 8: Human generated composition. Corresponds to composition 4 in the questionnaire.

panel consisted of twelve designers: four men and eight women. All of them had studied a bachelor's degree in design and half of them got a postgraduate degree. We developed a questionnaire that included four compositions: two were created by our system (compositions 2 and 3, Figures 5 and 6) and two were created by a designer (composition 1 and 4, Figures 7 and 8). The human compositions had to follow similar constraints to those of our program's compositions: they had to be in black and white, the designer can only employ one letter to develop her work, and so on. The participants were not told that some works had been done by a computer program. Subjects were asked to assess in a range from 1 (lowest) to 5 (highest) four characteristics for each composition: a) whether they liked the composition, b) whether they considered that the composition had symmetry, c) whether the composition had balance and, d) what kind of rhythm the composition had. They were also invited to comment freely on each composition regarding balance and symmetry. In the last part of the questionnaire, participants were asked to rank the compositions from the best to the worst. Figure 9 shows the results of the questionnaire.

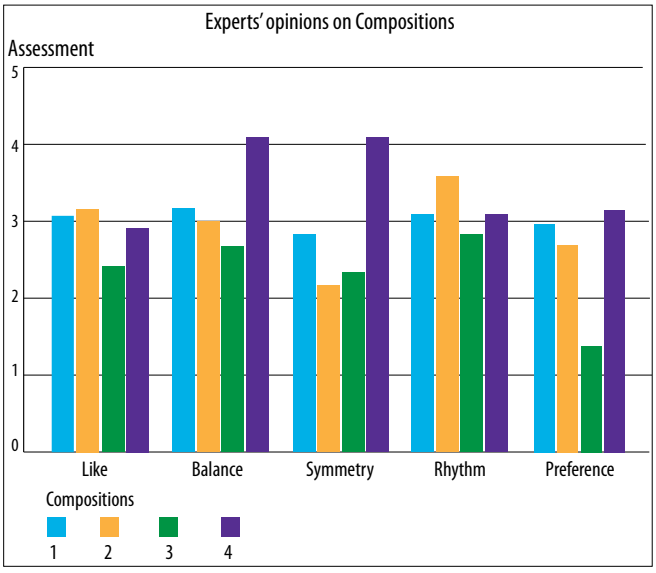


Figure 9: Results of the questionnaire.

Experts liked composition 1 and 2. This was an interesting result because it suggested that our model was capable of generating designs with an acceptable quality. It was also clear that most experts disliked composition 3 (Figure 6); although it is fair to say that their evaluation was only one point lower than the highest evaluation.

Compositions 1 and 4 (made by the human designer) had a better evaluation regarding balance and symmetry than compositions 2 and 3 (made by our program). We could have forced our program to generate symmetrical or balanced designs, but that was exactly what we wanted to avoid. Our system had the capacity of detecting such characteristics and nevertheless attempted something different. Expert's assessment on symmetry was neither clear nor unanimous. We were surprised to find this out, since symmetry does not depend on subjective judgment. Something similar occurred with balance and to some extent with rhythm. These results seemed to suggest that experts had different ways of evaluating these characteristics. Experts considered that the rhythm in Composition 2 was the best.

Overall subjects preferred composition 4; compositions 1 and 2 got similar results, with a slightly preference for composition 1; composition 3 got the lowest rank.

Discussion and Conclusions

This project describes a computer model for visual composition. The model establishes:

- A clear criteria to define simple-elements and groups.
- A set of attributes for simple-elements, groups and layers.
- Relationships between elements and a mechanism to identify such relationships.
- A method to analyse a visual composition based on layers, relationships and attributes.
- A mechanism based on the E-R model to produce novel compositions.

As far as we know, there is no other similar model. Although we are aware that many important features of compositions are not considered yet, we claim that our model allows a computer agent to produce novel visual designs.

We tested our model implementing a computer agent. The system was capable of producing compositions. None of them are alike to any of the previous designs, although some of its characteristics resemble the set of examples.

A panel of experts evaluated two compositions generated by our system and two compositions generated by a human designer. We decided to ask a small group of experts, who we believe share core concepts about design, to evaluate our prototype's compositions rather than to ask lots of people with different backgrounds. The results suggest two interesting points:

1. In most cases, the opinions of the experts were not unani-

mous. That is, some experts found more interesting some of the characteristics of the computer-generated composition than those produced by humans.

2. Experts seem to have different ways of perceiving and evaluating compositions.

Point 1 suggests that our model is capable of generating interesting compositions. That is, it seems that we are moving in the right direction.

Point 2 seems to confirm the necessity of clearer mechanisms to evaluate a composition. Of course, we are not suggesting that personal taste and intuition should be eliminated from design. We are only recommending the use of clearer definitions and mechanisms for evaluations. We are convinced that they will be very useful, especially in teaching and learning graphic composition.

One of the reviewers of this paper suggested comparing our work with shape grammars (Vere 1977, 1978). Our proposal is far of being a grammar; it does not include features like terminal shape elements and non-terminal shape elements. In the same way, we do not work with shapes but with relations between the elements that comprise the composition. Those relations drive the generation of new compositions. We believe that our approach is much more flexible than the grammars approach. A second reviewer suggested comparing our work with relational productions (Stiny 1972). It is true that our work also employs the "before and after" situations described by Stiny. However, we are not interested in modelling inductive (or any other type of) learning; our purpose is to record the actions that the user performs in order to progress a composition. Later, the system employs this information to develop its own composition. None of these two approaches include characteristics like a flexible generation process intertwined to an evaluation process, analysis by layers of the relations between the elements that comprise a composition, and other characteristics that our approach does. Thus, although some of the features that our model employs remind us of previous works, we claim that our approach introduces interesting novel features.

We hope this work encourage other researches to work on visual composition generation.

References

Agostini, F. 1987. *Juegos con la imagen*. Madrid: Editorial Pirámide.

Bentley, P. 1999. *An Introduction to Evolutionary Design by Computers*. Morgan Kaufmann Publishers.

Brainard, S. 1991. *A Design Manual*. New Jersey: Prentice Hall.

Colton, S. 2012. The Painting Fool: Stories from Building an Automated Painter, In *Computers and Creativity*, edited by J. McCormack and M. d'Inverno, Springer-Verlag.

Deepak J. M. 2010. *Principles of design through photography*. New Delhi: Wisdom Tree / Ahmedabad: National Institute of Design.

Faimon, P. and Weigand, J. 2004. *The nature of design*. USA: How Design Books.

Field, M. and Golubitsky, M. 1995. *Symmetry in Chaos: a Search for Pattern in Mathematics, Art and Nature*. Oxford: Oxford University Press.

Fullmer, D.L. 2012. *Design Basics*. USA: Fairchild Books.

Germani-Fabris 1973. *Fundamentos del proyecto gráfico*. España: Ediciones Don Bosco.

Goldberg, D. E. 1991. Genetic Algorithms as a Computational Theory of Conceptual Design. In *Proc. of Applications of Artificial Intelligence in Engineering* 6, pp. 3-16.

Hall T. E. 1999. *La Dimensión Oculta*. México D.F.: Siglo XXI. (Original title: The Hidden Dimension; Translated by: Félix Blanco).

Kinsey, L.C. and Moore, T.E. 2002. *Symmetry, Shape and Space. An introduction to Mathematics through Geommetry*. USA: Key College Publishing/Springer.

Lauer, D. A. and Pentak, S. 2012. *Design Basics*. USA: Wadsworth, 8th Edition.

Myers, J.F. 1989. *The Language of Visual Art. Perception as a basis for Design*. EUA: Holt, Reinehart and Winston, Inc.

Norton, D, Heath, D. and Ventura, D. 2011. Autonomously Creating Quality Images. In *Proceedings of the Second International Conference on Computational Creativity*, Mexico City, Mexico, pp. 10-15.

Pérez y Pérez, R., Aguilar, A. and Negrete, S. 2010. The ERI-Designer: A Computer Model for the Arrangement of Furniture. *Minds an Machines*, 20 (4): 483-487.

Pérez y Pérez, R. and Sharples, M. 2001 MEXICA: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13 (2):119-139.

Stiny, G. (1972). Shape grammars and the generative specification of painting and sculpture. *Information Processing* 71.

Vere, S. (1977). Relational production systems, *Artificial Intelligence*, Volume 8, Issue 1.

Vere, S. (1978). *Inductive learning of relational productions*. Academic Press Inc, University of Illinois at Chicago circle.

Wertheimer, M. 2012. *On perceived motion and figural organization*. Cambridge, Mass: MIT Press.

Learning how to reinterpret creative problems

Kazjon Grace
College of Computing and Informatics
University of North Carolina at Charlotte
Charlotte, NC, USA
k.grace@uncc.edu

John Gero
Krasnow Institute
for Advanced Study
George Mason University
Fairfax, VA, US
john@johngero.com

Rob Saunders
Faculty of Architecture, Design and Planning
Sydney University
Sydney, NSW, Australia
rob.saunders@sydney.edu.au

Abstract

This paper discusses a method, implemented in the domain of computational association, by which computational creative systems could learn from their previous experiences and apply them to influence their future behaviour, even on creative problems that differ significantly from those encountered before. The approach is based on learning ways that problems can be reinterpreted. These interpretations may then be applicable to other problems in ways that specific solutions or object knowledge may not. We demonstrate a simple proof-of-concept of this approach in the domain of simple visual association, and discuss how and why this behaviour could be integrated into other creative systems.

Introduction

Learning to be creative is hard. Experience is known to be a significant influence in creative acts: cognitive studies of designers show significant differences in the ways novices and experts approach creative problems (Kavakli and Gero, 2002). Yet each creative act is potentially so different from every other act that it is complex to operationalise the experience gained and apply it to subsequent acts of creating.

Systems that can, through experience, improve their own capacity to be creative are an interesting goal for computational creativity research as they are a rich avenue for improving system autonomy. While computational creativity research has coalesced over the last decade around quantified ways to evaluate creative output, there have been few attempts to imbue a system with methods of self-evaluation and processes by which it could learn to improve. This research presents one possible avenue for pursuing that goal.

A distinction should be drawn between learning about the various objects and concepts to be used in particular creative acts, which serves to aid those acts specifically, and learning about how to be a better creator more broadly. Knowledge about objects influences future creative acts with those objects, but the generalisability of that knowledge is suspect.

One example of where this learning challenge is particularly relevant is analogy-making, in which every mapping created between two objects is, by the definition of an analogy as a new relationship, in some way unique. Multiple analogies using the same object or objects are not guaranteed to be similar. This makes it very difficult to generalise knowledge about making analogies and apply it to any future analogy-making act.

We propose to tackle this problem of learning to be (computationally) creative by learning ways to interpret problems, rather than learning solutions to problems or learning about objects used in problems. These interpretations can be learnt, evaluated, recalled and reapplied to other problems, potentially producing useful representations. This process is based on the idea that perspectives that have been adopted in the past and have led to some valuable creative output may be useful to adopt again if a compatible problem arises. While even quite similar creative problems may require very different solutions, quite different problems may be able to be reinterpreted in similar ways. We discuss this approach specifically for association and analogy-making but it may hypothetically apply to other components of computational creativity. We develop a proof-of-concept implementation in the domain of computational association, and outline some ways in which this learning of interpretations could be more useful than object- or solution-learning in creative contexts.

Models for how previous experiences can influence behaviour could be a valuable addition to learning in creative systems. A computational model able to learn ways to approach creative problems would behave in ways driven by its previous experiences, permitting kinds of autonomy of motivation and action currently missing from most models of computational creativity. For example, it would be possible to develop a creative system that could autonomously construct aesthetic preferences based on what it has (or has not) experienced, or to learn styles by which it can categorise the work of itself and others, such as described in (Jennings, 2010). A creative system capable using past experiences to influence its behaviour is a key step towards computationally creative systems that are embedded in the kind of rich historical and cultural contexts which are so valuable to human artists and scientists alike.

Learning interpretations in computational association

We have previously developed a model of computational association based on the reinterpretation of representations so as to render them able to be mapped. Our model, along with an implementation of it in the domain of ornamental

design, is detailed in (Grace, Gero, and Saunders, 2012). We distinguish association from analogy by the absence of the transfer process which follows the construction of a new mapping: analogy is, in this view, association plus transfer. Interpretation-driven association uses a cyclical interaction of re-representation and mapping search processes to both construct compatible representations of two objects and produce a new mapping between them. An interpretation is considered to be a transformation that can be applied to the representations of the objects being associated. These transformations are constructed, evaluated and applied during the course of a search for a mapping, transforming the space of that search and influencing its trajectory while the search occurs. This differs from the theory of rerepresentation in analogy-making presented in Yan, Forbus and Gentner 2003 as in our system representations are iteratively adapted in parallel with the search for mappings, rather than only after mapping has failed. This permits interpretation to influence the search for mappings, and for mapping to influence the construction, evaluation and use of interpretations in turn.

The implementation of this model preented here explores the process of *Interpretation Recollection*, through which interpretations that have been instrumental in creating past associations can be recalled to influence a current association problem. This process occurs in conjunction with the construction of interpretations from observations made about the current problem.

In the model interpretation recollection is a step in the iterative interpretation process in which the set of past, successful interpretations is checked for any interpretations appropriate to the current situation. These past interpretations will then be considered for application to the object representations alongside other interpretations that have previously been constructed or recalled. A successful interpretation – one that has previously led to an association – can thereby by reconstructed and reapplied to a new association problem. In this paper we demonstrate that this feature of the interpretation-driven model leads to previous experiences influencing acts of association-making, and claim that this is promising groundwork for future investigations into learning in creative contexts.

In the implementation described in this paper we use simplified approaches to determining the relevance of previously successful interpretations and reapplying them to the current context. The metric for determining appropriateness is straightforward: any previous interpretation which has a non-zero effect on a current object representation is determined to be capable of influencing the course of the current association problem and included. This simplifies the notion of "appropriate for future use" and leads to an obvious scalability issue, but we demonstrate that this very simple approach influences behaviour. More sophisticated methods for determining when and how known interpretations should be reapplied are an area of future investigation.

Experimenting with learnt interpretations

As a preliminary investigation into the potential of interpretation-based creative learning, we will demonstrate that the approach we have developed permits previous experience to influence the behaviour of an association system. To illustrate this we will prime the system to produce different results after having experienced different histories. In our system previously constructed associations can influence new association problems through interpretation learning; past associations can act to "prime" the system to produce particular results on future associations. By demonstrating that an association system's experience with one pair of objects can influence its behaviour associating different objects, we show the advantage of interpretation-based approach to learning. Comparatively an object-based approach to learning would not have permitted generalisation to an unfamiliar pair of objects.

In our experiments the system is exposed to a particular stimulus (either a simple unambiguous association problem or nothing in the case of the control trial) and then attempts to solve an ambiguous association problem that is the same between all trials. Our association system produces many different mappings between any two objects, so changes in the distribution of mappings produced on the second problem is used as an indicator of priming effects.

Three trials were conducted. In the first trial no priming association was performed, in the second trial a priming association between Objects 1 and 2 of Figure 1 was performed, and in the third trial a priming association Objects 1 and 3 of Figure 1 was performed. In each trial an association between Objects 4 and 5, depicted in Figure 2, followed the priming stage. Each trial was performed 100 times, with the system being re-initialised (and re-primed) between each one so that the histories are identical for every association. A distribution of the results of the association between Objects 4 and 5 was produced. All trials were conducted using three relationships: relationships of the relative orientation of shapes, such as '$\sim 45^\circ$ difference in orientation'; relationships of the relative vertical separation of shapes, such as '$\sim$3 units of separation in the Y axis'; and simple binary relationships when two shapes share vertices.

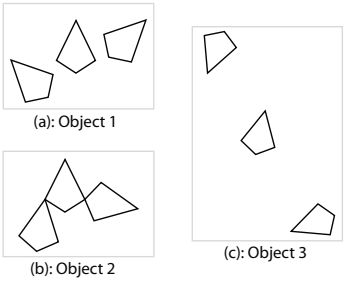


Figure 1: The three objects used in the priming associations. An association between either Objects 1 and 2 or Objects 1 and 3 is used to prime the interpretation system.

The two associations used for priming are designed to repeatably produce a predictable association based on a predictable interpretation - making them well suited to testing the impact of priming an association system with that

interpretation. The system perceives Objects 1 and 2 and constructs a simple association based on equating the pattern of relative rotations between features in Object 1 with the pattern of shared vertices between features in Object 2. In the other trial, the system perceives Objects 1 and 3 and constructs another simple association, this time equating the pattern of relative rotations in Object 1 with the pattern of relative vertical positions in Object 3. These associations are depicted in Figure 3, with the thick dashed lines between features within the objects denoting relationships that were mapped, while solid lines between features joining the two objects denote which features were mapped to each other.

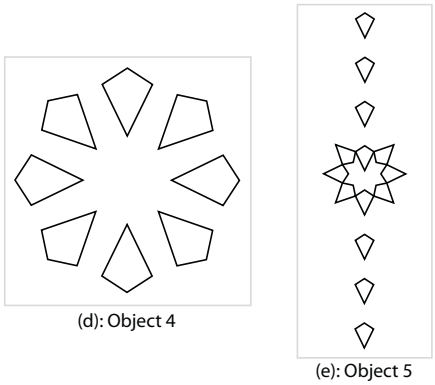


Figure 2: The two objects used in the test association of all three trials, which is used to measure the effects of the priming associations.

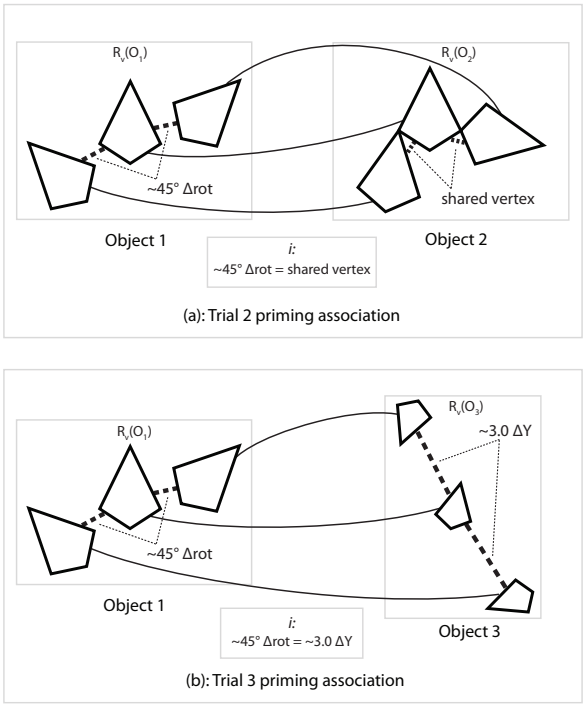


Figure 3: The solutions to the association problems used to prime the system in the second and third trials. These simple problems predictably influence the experiential component of the creative system in ways that can then be measured.

These simple associations effectively prime the system with an interpretation which will predictably bias the dependent association between Objects 4 and 5. This bias provides a proof-of-concept test of experiential influence. Future studies are needed to determine the scope of influences which historical context can exert in creative systems.

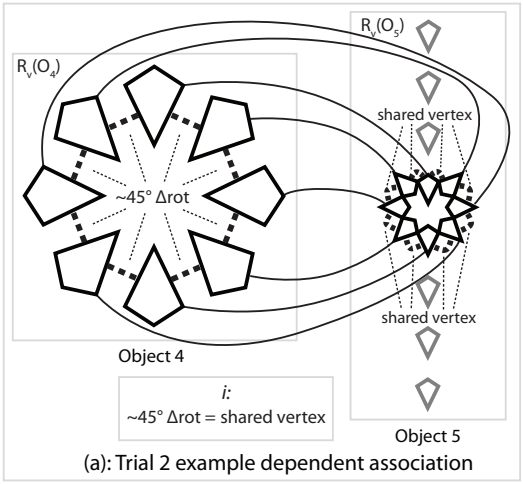
The post-priming association problem used in all three trials is designed to have two dominant solutions. Over many runs the system will produce many other associations in addition to these two, but these will occur relatively often. The two associations can be seen in Figure 4, with association (a) being between the radial arrangement of shapes in Object 4 and the similar arrangement of touching shapes in Object 5, and association (b) being between the same arrangement in Object 4 and the vertically spaced shapes in Object 5.

It is hypothesised that when the system is first primed with the association in Figure 3(a) the solution in Figure 4(a) will be more common (than when unprimed), and that when the system is first primed with the association in Figure 3(b) the solution in Figure 4(b) will instead be more common (than when unprimed). This outcome would demonstrate the feasibility of using interpretation-based learning to enable a creative system's experiences to influence its actions.

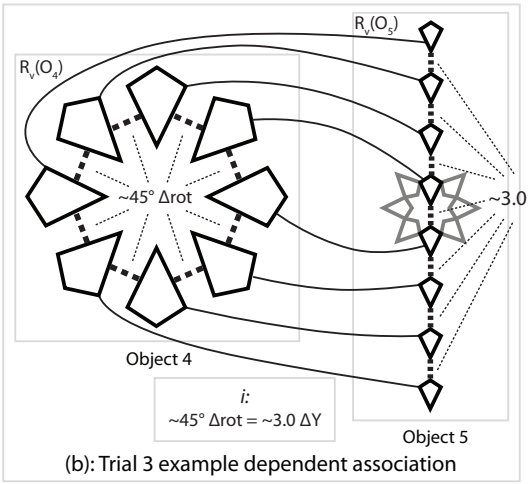
Experimental results

The distribution of associations produced in each trial can be seen in Figure 5. Each of the three bars represents one trial, and each of the three different shading tones represent a different result, with the darkest tone representing the solution seen in Figure 4(a), the middle tone representing Figure 4(b), and the lightest tone representing all other solutions. The latter category included fragmented mappings (those for which the system could not find a complete mapping of all the shapes in Object 4) based on relationships such as 90° and 135° orientation differences as well as similar varieties and combinations of vertical separation and vertex sharing relationships. Although they are irrelevant to this investigation of priming effects, at present this implementation has no way of evaluating associations other than the number of features which are mapped. See (Grace, Gero, and Saunders, 2012) for a discussion of the evaluative capabilities of this model and its current implementation.

It is clear from Figure 5 that priming the association system with previous problems that rely on compatible interpretations leads to a significant influence on the outcome of the association process. Trial 1, in which no priming is performed, serves as a control against which the frequency of different associations can be compared. In Trial 2 the system is primed with a problem that relies on the adoption of an interpretation equating a pattern of rotational relationships with a pattern of shared vertices. The result for Trial 2 clearly shows that the frequency of solutions relying on this interpretation (such as the one seen in Figure 4(a)) has



(a): Trial 2 example dependent association



(b): Trial 3 example dependent association

Figure 4: Two of the possible solutions to the dependent association performed in each trial. Solution (a) uses the same interpretation as used in Figure 3(a), while solution (b) uses the one found in Figure 3(b).

increased significantly, from 17% in the control to 63% in Trial 2. In Trial 3 the system is primed with a problem that relies on equating the same pattern of rotational relationships with a pattern of vertical separation, shown in Figure 4(b). The result for Trial 3 shows a similarly significant increase in frequency, with 36% frequency for the primed trial compared to only 3% in the control.

The difference in absolute frequency of the two associations shown in Figure 4 can be explained by the underlying graph structures and the process for searching them used in our model. The association primed for in Trial 2 is based on the "shared vertex" relationship, which is 50% more common in Object 5's graph representation than the "3.0 difference in the Y axis" relationship used in the interpretation primed for in Trial 3. For information on how our system automatically extracts these and other relationships from vector representations of the objects see (Grace,

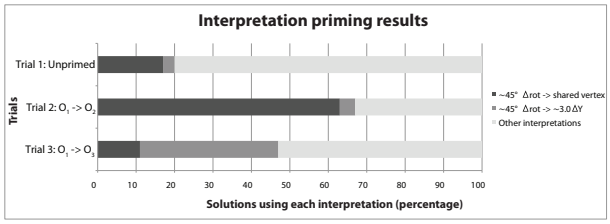


Figure 5: The distribution of association results in each trial, showing the influence of the priming in Trials 2 & 3.

Gero, and Saunders, 2012). The commonality of that relationship makes mappings that involve features connected by that relationship similarly more common, which makes it more likely to be utilised by both the mapping and interpretation processes. This bias makes the vertex-sharing relationships much more likely to feature in associations, but priming the system towards a less common result largely overrides it. This can be seen in the twelve-fold increase in the likelihood of the less-common association as compared to the only three-fold increase in the more common one.

These results show that it is possible for the learning of interpretations to influence the behaviour of a creative system, and demonstrate our model of association's capacity for interpretation learning and experiential influence on behaviour. While the influence on behaviour produced in this implementation is limited, these experiments demonstrate that interpretation learning can influence behaviour on problems significantly different than those previously experienced. This shows the potential for more general learning than is possible by solution- or object-based methods, making this approach a valuable building block for modelling learning computational creativity.

Discussion

The experiments described in this paper are a demonstration that the behaviour of creative systems can be influenced by storing and reusing ways to interpret creative problems. This section discusses the impact on creativity of drawing from experience to reinterpret a problem and the ways interpretation can influence creative acts. For a more general discussion of our model and how it compares to other models see Grace et. al (2012).

Re-using interpretations for creativity?

There is an intuitive objection to the idea of re-using elements of a previous creative process: that process, or at least that element of the larger creative process, cannot by definition be p-creative. While the process may go on to produce p- or h-creative outputs, it will at least partially be based on things that have been experienced previously.

The p-creativity, or lack thereof, of any element of the creative process does not imply any impact on the creativity of the final product, but the objection bears discussion: if drawing on experience will only reduce the creativity of a process, what is its value? Investigations of the diversity

of solutions both with and without priming show that there is no significant reduction in the breadth of solutions produced, only in the order in which the system produces them. This is due to the novelty-favouring behaviour of our model, which over time discounts and eventually discards solutions to a particular problem which have repeatedly arisen. Such intrinsic motivations towards novelty are a necessary component of learning creative systems, balancing the desire to repeat the familiar against the desire to explore the new.

(Suwa, Gero, and Purcell, 1999) propose a third element to Boden's categorisation of creativity (1992), 'situational', or s-creativity, to describe when an object or process is not absolutely new to an agent, but is new within the current situation. This occurs when a familiar idea is considered for the first time in an unfamiliar context, a common outcome of analogy-making and a potent component of experiential learning. This is particularly applicable to the notion of reusing interpretations, which have the potential to transform the solution space of the current problem despite not being a novel process to the agent in question.

Kinds of interpretation and their influence

In the system presented here interpretations are simple transformations that are stored and re-applied verbatim. However, the notion that interpretations can influence future acts does not require that the previously useful interpretation be literally re-applied to the new context. It would be possible to develop a system in which exemplary, prototypical or generalised interpretations could be reconstructed from experience and applied to the current context.

We define interpretations as a transformations applied to the objects being associated, but this need not be a direct transformation of the object representations used by the system. Other elements of the model could be transformed, such as evaluative processes, which would change not the information being used in the creative process but its value metrics. This could lead to experiential influence on aesthetic judgement, similar to the idea of autonomously derived aesthetics proposed by Colton (2011). Alternatively, representational processes of the model could be transformed, for example relaxing thresholds for categorisation or similarity. This could lead to behaviours like satisficing, a common behaviour of human designers in which requirements are changed during the creative act (Simon, 1957).

Conclusions

This paper proposes the notion of interpretation-learning – the storage and recollection of ways to transform problems – as a complement to more familiar models of object- or solution-learning. Interpretation-learning is hypothesised as being of particular utility in creative contexts as each creative problem is unique in its solutions, but potentially not in the ways it can be perceived. These remembered interpretations can be thought of as granting a creative system more autonomy over its decision making than other means of deciding how to interpret problems such as provided heuristics or stochastic processes. We present a simple implementation of a creative system in which past experiences influence behaviour through interpretation, to serve as a proof-of-concept of the notion of interpretation-learning. With this approach demonstrated as feasible and promising, future work can explore its efficiency and effectiveness.

Incorporating learning is emerging as an important component of computational creativity due to growing prominence of desired behaviours like surprise (Maher, 2010), appreciation (Colton, Goodwin, and Veale, 2012) and autopoeisis (Saunders, 2012), which necessarily involve past experience. Learning about specific objects or outcomes is of limited utility in computational creativity, as creative problems are by definition unique. However, learning and recalling different perspectives through which to view objects is one process by which learning in creative contexts could be modelled.

References

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The face and idea models. In *Proceedings of the 2nd International Conference on Computational Creativity*. 90–95.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of the 3rd International Conference on Computational Creativity*. 95–102.

Grace, K.; Gero, J.; and Saunders, R. 2012. Constructing computational associations between ornamental designs. In *Proceedings of the 17th International Conference on Computer Aided Architectural Design Research in Asia*, 37–46.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Mind and Machines* 20:489–501.

Kavakli, M., and Gero, J. S. 2002. The structure of concurrent cognitive actions: A case study of novice and expert designers. *Design Studies* 23:25–40.

Maher, M. L. 2010. Evaluating creativity in humans, computers, and collectively intelligent systems. In *DESIRE10: Creativity and Innovation in Design*.

Saunders, R. 2012. Towards autonomous creative systems: A computational approach. *Cognitive Computation, Special Issues on Computational Creativity, Intelligence and Autonomy* 4:216–225.

Simon, H. 1957. *Models of Man - Social and Rational*. New York: John Wiley and Sons.

Suwa, M.; Gero, J.; and Purcell, T. 1999. How an architect created design requirements. In Goldschmidt, G., and Porter, W., eds., *Design Thinking Research Symposium: Design Representation*, volume 2. MIT Press. 101–124.

Yan, J.; Forbus, K.; and Gentner, D. 2003. A theory of rerepresentation in analogical matching. In Alterman, R., and Kirsch, D., eds., *Proceedings of the Twenty-Fifth Annual Meeting of the Cognitive Science Society*, 1265–1270. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Computational Creativity in Naturalistic Decision-Making

Magnus Jändel
Swedish Defence Research Agency
Stockholm, SE-16490, Sweden
magnus.jaendel@foi.se

Abstract

Creativity can be of great importance in decision-making and applying computational creativity to decision support is comparatively feasible since novelty and value often can be evaluated by a reasonable human effort or by simulation. A prominent model for how humans make real-life decisions is reviewed and we identify and discuss six opportunities for enhancing the process with computational creativity. It is found that computational creativity can be employed for suggesting courses of action, unnoticed situation features, plan improvements, unseen anomalies, situation reassessments and information to explore. For each such enhancement opportunity tentative computational creativity methods are examined. Relevant trends in decision support research are related to the resulting framework and we speculate on how computational creativity methods such as story generation could be used for decision support.

Introduction

Creativity and decision-making Before the battle of Austerlitz 1805, Napoleon deceptively maneuvered to create the impression of weakness and indecision in the French forces. The opposing Russo-Austrian army took the bait, attacked and fell into a carefully prepared trap resulting in a crushing defeat. Detecting deception requires an act of creativity where the reality of the situation is discerned behind a screen of trickery. European history could have taken a different turn with more creative leadership on the Russian and Austrian side. Leaders of today are likewise challenged to be more creative. Given the progress of computational creativity in other fields, it is therefore interesting to pursue its application to decision-making.

Computational creativity for decision support The key problem in computational creativity is how to automatically assess the novelty and creative value of an idea, concept or artifact that has been generated by computational means (Boden, 2009). This is a very difficult problem in art where novelty is judged by comparing to extensive traditions and evaluation would engender implementation of computational esthetic taste. Decision support is fundamentally less challenging. Novelty is often judged against a reasonably short list of options that are known to the decision makers and value is evaluated by analyzing how the idea works in the situation at hand. Computer simulations are increasingly employed for assisting decision makers and it is often quite feasible to use simulations for automatic evaluation of suggested ideas. Given the comparative straightforwardness of applying computational creativity to decision support, it appears that there are surprisingly few applications. Some of these are discussed and put in context after that we have introduced the framework that is the main result of this paper.

Decision-making models Applying computational creativity to any given area of decision-making requires substantial domain knowledge and it is often difficult to see how methods generalize to other domains. Our strategy is therefore to identify generic approaches by analyzing how formal decision-making models can be extended to include computational creativity techniques.

Somewhat simplified, decision-making models can be partitioned into two general classes: rational models and naturalistic models. The former prescribes how decisions ought to be made while the latter describes how people really make decisions. Many naturalistic models surpass, however, their purely descriptive origins and offer some suggestions on how intuitive decision-making can be improved.

In the following two sections we analyze how computational creativity tools can extend rational and naturalistic decision-making models respectively with a strong focus on a particularly prominent naturalistic model.

Rational decision-making models

Rational decision-making models provide methods for how to optimally select an action from a set of alternative actions (Towler 2010). In utility-based decision-making it is for example assumed that each action leads to a set of outcomes and that the probability of each outcome is known or can be estimated. Furthermore, each outcome has a utility which is a real-valued variable and the task of the decision-maker is to select the action that is most likely to optimize the utility of the outcome. Other rational schemes extend this approach to cases with multiple objectives and multiple constraints (Triantaphyllou, 2002). The main US army Military Decision-Making Process (MDMP) is for example essentially a rational process where it is required

that at least three different courses of action should be compared.

Rational decision-making models characteristically give little guidance on how to generate the set of action alternatives although it is tacitly assumed that more alternatives make for better decisions. Since the 1980s, it has been claimed that decision makers typically don´t employ rational models (Kahneman, Slovic, and Tversky 1982) and it appears further that leaders don´t find rational models to be efficient (Yates, Veinott and Patalano 2003) and that generating more alternatives actually can be detrimental for decision quality (Johnson and Raab 2003).

Computational creativity could assist rational decision-making by suggesting criteria, enriching the set of action alternatives, envisioning possible outcomes of actions and suggesting factors that should be considered in mental or computer simulations. The methods that could be employed for this are often quite similar to corresponding methods in naturalistic decision-making which is the focus of this paper.

Extended naturalistic decision-making model

Naturalistic decision-making models are inspired by research in how decisions are made in domains such as business, firefighting and in military areas. Investigations indicate that experienced and effective leaders evaluate the nature of the situation intuitively and rarely consider more than one course of action (Klein 2003).

Figure 1 summarizes a leading naturalistic decision-making model: the recognition-primed model (RPD). For the moment, please ignore the symbols **CC1**, **CC2**, … **CC6**. This paragraph briefly reviews work of Klein and coworkers on RPD (Klein, Calderwood and Clinton-Cirocco 1986; Ross, Klein, Thunholm, Scmitt, and Baxter, 2004; Klein 2008). The experienced decision maker evaluates the state of affairs and will normally recognize a familiar type of situation. Recognition means that the relevant cues or indicators are pinpointed; expectancies on how the situation will appear and unfold are identified; what kind of goals that are reasonable to pursue are recognized and a typically short list or courses of actions are found. In the following we use *course of action* or *action* to designate the conceptual level of a top-level plan that if implemented will consist of a chain of component actions or plan elements. As a reality-check the expectancies are analyzed and compared to available information. Any anomalies found trigger an iteration of the recognition process were more information may be sought and the situation is reassessed, sometimes leading to a major shift in how the situation is perceived. Eventually, the decision maker arrives to a satisfactory anomaly-free situation-recognition and selects the most promising course of action for scrutiny. The consequences of performing the selected course of action is simulated either mentally or by computer. This may lead to that the course of action is rejected and another option is selected for a new round of simulation. Frequently it is found that the course of action is promising but that it has some unwanted consequences. Rather than rejecting the course of action, decision makers will try to repair the plan by modifying the chain of plan elements that implement the course of action. It is implicit in Figure 1 that modified courses of actions are re-simulated. Eventually the decision-maker will find a satisfactory course of action which will be implemented. Note that the RPD process does not include a search for the optimal course of action, the optimal implementation or quantitative utility criteria. If the plan satisfies the recognized goals it is deemed to be ready for implementation.

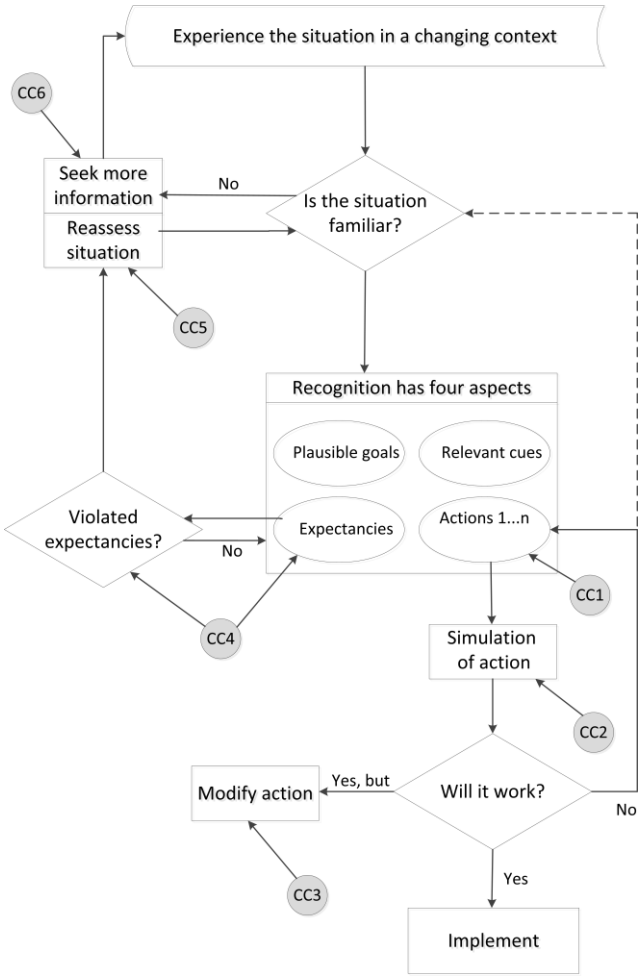


Figure 1. Computational creativity extensions to the recognition-primed model. Filled circles denote computational creativity agents. Everything else in the figure is quoted from Klein (2008).

How can decision makers that use RPD or some similar naturalistic decision-making model take advantage of computational creativity? In Figure 1, we mark six slots where a computational creativity agent could be plugged into the RPD process. The computational creativity agents are called **CC1**, **CC2**, … **CC6** and these symbols are used in the following to highlight where the different computational creativity extensions are mentioned. For each agent we provide a mnemonic tag, discuss in which way it could improve decision-making and provide a sketch of at least

one creativity technology that could be applicable. Finally we provide a speculative example illustrating why creative input could be of great value in the current decision-making phase.

CC1 (proposing actions): Computational creativity could be used for suggesting a broader range of courses of action in a recognized situation. The **CC1** agent would work under the assumption that the situation and the relevant goals are correctly identified and that the creative task is to find unrecognized course of action alternatives that lead towards the "plausible goals" in Figure 1. The decision maker has identified the nature of the situation which will suggest a well-defined search space of actions. Some of the actions in the search space are explicitly known to the decision maker and would hence be found in the list of actions indicated in Figure 1. The RPD process evaluates listed actions. Creative suggestions should hence point to feasible actions that are significantly different from already listed actions. The **CC1** algorithm must define a similarity metric in action space and the list of actions that are known to the decision-maker should be available to the **CC1** agent so that it can avoid searching too close to known actions. Ideally, the **CC1** agent uses a simulation engine for confirming the approximate validity of courses of action but it might also be possible to fall back on human adjudication. Candidate courses of action that are far from known courses of action according to the metric and pass the simulation test are suggested to the decision maker and added to the known list. A government wanting to integrate an island population to the mainland society may for example consider courses of actions such as building a bridge, airport or ferry terminal. The **CC1** agent, realizing that known courses of action all relate to physical connectivity, may suggest courses of actions such as investing in telepresence or locating a new university to the island.

CC2 (proposing features): Simulations are never completely realistic but will always model some aspects of the situation at hand with higher fidelity than others and also ignore many other aspects. The **CC2** agent could suggest features that should be included in computer or mental simulations. Such ideas might be crucial for success since the acuity of the simulation is essential for the quality of the plan that implements the selected course of action. Consider for example a decision maker trying to control flooding caused by a burst dam. The core simulation would be concerned with modeling how actions influence the flow of water. A **CC2** agent searching historical records of floods could come up with the suggestion that modeling the spread of cholera might be important. The **CC2** agent could for example grade the importance of candidate features by measuring how often they are mentioned in news stories on flood-related events.

CC3 (reparing plans): Computational creativity could be used for suggesting how a promising but somewhat flawed plan can be repaired. Assume that simulation has exposed at least one problem with the current course of action and that the decision makers have the mental or computational means for re-planning but are out of ideas. The task of the **CC3** agent is to provide an idea for how the problem can be solved. The main planning process can then use the idea for driving the next iteration of re-planning. Consider a case in which the main planning process is a planning algorithm (Ghallab, Nau, and Traverso 2004) that works by searching for a chain of plan elements that implements the course of action. Each plan element has a set of prerequisites and a set of consequences. The planner searches for chains of plan elements where all prerequisites are satisfied, the consequences at the end of the chains match the goals and the general direction of the plans is consistent with the currently considered course of action. If the planning algorithm fails to find a problem-free plan, the **CC3** agent could suggest a new plan element. This creative output is validated if the planner solves or alleviates the problem by using the suggested action element in a modified plan. The task of the **CC3** agent could be construed as search in the space of possible plan elements where the identified problem may be used for heuristic direction of the search. Note that the **CC3** agent should not be another planner that by explicit planning guarantees that the suggested plan element solves the problem. It is sufficient that suggested plan elements have a high probability of contributing to the solution. The **CC3** agent should obviously be aware of the present set of plan elements that are used by the main planner and avoid suggesting elements that are identical or very similar to currently known plan elements. A government may for example have selected reduction of the national carbon footprint as the chief course of action for environmental protection but fails to find a plan that reaches the target. A **CC3** agent could then suggest enhanced weathering, where crushed rock absorbs carbon dioxide, as a new plan element.

CC4 (identifying anomalies): Computational creativity could be used for identifying anomalous expectancies in the current perspective on the situation. When a decision maker has identified the situation as familiar it is often difficult to notice aspects of the situation that do not fit into the familiar context. It is crucial to find any anomalies since this might trigger a radical reassessment of the situation. The **CC4** agent is best applied when the decision-maker has exhausted the manual search for anomalous expectancies and is ready to proceed with action evaluation. A simple version of the combinatorial approach could explore the space of situation features searching for pairs of features that in combination stand out as anomalous. This could be done by investigating second-order attributes of the features and noting how combinations of attributes interact. The obscure features method (McCaffrey and Spector 2011) might be adapted for this purpose. Simulation methods that are used for evaluating actions could also be applied to examining anomaly candidates with validated anomalies escalated for human consideration. The Russian and Austrian leaders at Austerlitz would have benefited

from a **CC4** agent suggesting that Napoleon´s uncharacteristic eagerness to negotiate and seemingly panicky abandonment of important positions were anomalies deserving serious attention.

CC5 (situation assessment): Supporting reassessment of the situation is the most challenging creative task. Imagine that the decision maker has noted a number of anomalies indicating that the present situation recognition is flawed but no viable alternative interpretations pop up in human minds. People are often locked into habitual trains of thought and this behavior is frequently aggravated by time pressure, fear and group-think. Computational creativity is free from such human frailties and might be able to suggest new ways of looking at the situation. A single idea might be enough for providing the Aha! experience that releases the intuitive power of the decision maker. A simple implementation of a **CC5** agent could use a library of case histories enshrining human expert findings in a broad range of circumstances. A sufficiently small volume of decision-making experience could, as noted by M. Boden (personal communication), advantageously be codified as a check list. **CC5** agents would be needed only in contexts in which the total span of assessment possibilities is large and inscrutable. A police officer leading the investigation of suspected arson in an in-door food market could for example benefit from the suggestion that spontaneous combustion of pistachio nuts might be an alternative perspective on the evidence (see Hill (2010) for further information on spontaneous combustion). The **CC5** agent would in this case use encyclopedic knowledge that pistachio nuts are a kind of food and that pistachio nuts are subject to spontaneous combustion combined with records of historical cases in which suspected arson has been found to be explained by spontaneous combustion.

CC6 (recommending information): Computational creativity could be used for suggesting what kind of information that could support reassessment or resolution of apparently anomalous expectancies. Decision makers often have access to vast archives and abundant streams of news and reports. Selecting what deserves attention is a difficult and sometimes creative task. Decision makers would be biased by their present understanding of the situation so the **CC6** agent might be able to provide a fresh perspective. The task of the **CC6** agent is quite similar to that of the **CC2** agent; it must explore the space of information sources and information aspects for the purpose of identifying novel and valuable pieces of information. A doctor confronted with anomalous symptoms could for example get suggestions from a **CC6** agent regarding which medical tests to apply.

Discussion and conclusions

In this section we will first discuss some current applications of computational creativity to decision support in relation to the framework described in the previous section and then speculate on how selected approaches to computational creativity could be applied to decision support.

Examples of computational creativity in decision support Computer chess is probably the most advanced current application of computational creativity in decision support. Grand masters learn creativity in chess by studying how computers play (Bushinsky 2012). The main reason for this success is that chess is a very complex but deterministic game that readily can be simulated. The complexity of the game makes it possible for a computer to discover solutions that escape the attention of humans and simulation combined with heuristic assessment of positions enable automatic evaluation of computer generated solutions. The main components of creativity - novelty and value - are therefore attainable by chess programs. Referring to Fig. 1, we note that chess players use computational creativity mainly for suggesting courses of action (**CC1**) and repairing plans (**CC3**).

Tan and Kwok (2009) demonstrate how Conceptual Blending Theory (Fauconnier and Turner 2002) can be used for scenario generation intended for defense against maritime terrorism. The scenarios are examples of **CC5** agent output since they assist decision makers in assessing situations that may look peaceful and familiar at first sight but in which creative insights may reveal an insidious attack pattern.

Deep Green is a DARPA research program that aims for a new type of decision support for military leaders (Surdu and Kittka 2008) According to the Deep Green vision commanders should sketch a course of action using an advanced graphical interface while AI assistants work out the consequences and suggests how the plan can be implemented. The AI assistants are guided by thousands of simulations that explore how the situation could evolve and what factors that are important. According to our analysis in the previous section Deep Green seems to include development of **CC2** and **CC3** computational creativity agents although computational creativity is not explicitly mentioned in the Deep Green program.

Emerging tools Creative story generation could be turned into tools for decision support. Story generation techniques that spin a yarn connecting a well-defined initial state with a given final state (Riedl and Sugandh, 2008) could be used by **CC2** agents for suggesting improvements in simulations forecasting the outcome of plans. The **CC2** agent could generate stories that starts with the present situation, implements the course of action under consideration and ends with failure. Analysis of the generated story could give decision makers insights into aspects and circumstances that should be simulated carefully. **CC3** agents could also use the stories for suggesting countermeasures. With a comprehensive domain-related supply of vignettes, story generation might even be used for situation assessment by **CC5** agents. It is interesting to note that the techniques of vignette-based story generation are similar to those of planning algorithms and simulation engines. The difference is in purpose rather than in methodology. Story

generators aim for novelty, planners for optimality and simulations for sufficiently realistic modeling of some relevant aspects of reality.

It can be difficult for decision makers to fully understand the ramifications of goals that have been adopted by opponents or partners. This may cause errors in situation recognition and in identifying relevant expectancies in the current situation assessment. Agent-based story generation where an open-ended story evolves driven by conflicting goals (Meehan 1976) could be useful for both **CC4** and **CC5** decision support agents. Such stories could give a fresh perspective from a different point of view and help identifying anomalies and possibly inspire reassessment of the situation.

Consider a **CC3** agent that, as discussed in the previous section, is tasked with coming up with new plan elements for the purpose of repairing a failed plan. Li et al. (2012) extends conceptual blending (Fauconnier and Turner, 2002) to incorporate goals with application to algorithms for generating hypothetical gadgets engineered to fulfill the goals. This methodology could be applied to algorithms for **CC3** agents in which the goals are derived from the needs of the jammed planning process and generated "gadgets" would be plan elements with prerequisites that can be fulfilled in the context of the problem-ridden plan and consequences designed to be instrumental for unjamming the planning process.

Jändel (2013) describes information fusion systems extended with computational creativity agents of type **CC5**. The agents aid in uncovering deceit by comparing generic deception strategies to the present situation and guide the fusion process to explore alternative situation assessments.

Future applications There are many research opportunities in the confluence of computational creativity and naturalistic decision-making both with respect to algorithms for the six types of agents indicated in this paper and for research into the effect and efficiency of computational creativity in various domains of decision-making.

Pioneering areas of application will probably be in high-stake strategic decision-making where time and resources are at hand and leaders are willing to go to great lengths in order to minimize risks and ensure decision quality. Bridgehead applications will therefore most likely be in fields such as defense strategy, major economic and environmental decisions and strategic business planning. As methods and tools evolve and the level of automation increases computational creativity will increasingly be applied also to operative and tactical decision-making.

Acknowledgments

This research is financed by the R&D programme of the Swedish Armed Forces.

References

Boden, M. 2009. Computer models of creativity. AI Magazine 30(3):23–34.

Bushinsky, S. 2009. Deus ex machina a higher creative species in the game of chess. AI Magazine 30(3):63–69.

Fauconnier, G., and Turner, M. 2002. The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic Books.

Ghallab, M.; Nau, D.S.; .and Traverso, P. 2004. Automated Planning: Theory and Practice. Morgan Kaufmann.

Hill, L. G. 2010. ShockWave Science and Technology Reference Library, V 5, Non-Shock Initiation of Explosives. Springer.

Johnson, J., and Raab, M. 2003. Take the first: Option-generation and resulting choices. Organizational Behavior and Human Decision Processes 91:215229.

Jändel, M. 2013. Computational creativity for counterdeception in information fusion. Unpublished, submitted to 16th Int. Conf. on Information Fusion.

Kahneman, D.; Slovic, P.; and Tversky, A. 1982. Judgement under uncertainty: Heuristics and biases. Cambridge University Press.

Klein, G.; Calderwood, R.; and Clinton-Cirocco, A. 1986. Rapid decision-making on the fireground. In Proceedings of the Human Factors and Ergometrics Society, 576–580.

Klein, G. 2003. Intuition at work. New York: Doubleday.

Klein, G. 2008. Naturalistic decision making. Human Factors 50:456–460.

Li, B.; Zook, A.; Davis, N.; and Riedl, M. . 2012. Goal driven conceptual blending: A computational approach for creativity. In Proceedings of the 2012 International Conference on Computational Creativity.

McCaffrey, T., and Spector, L. 2011. How the obscure features hypothesis leads to innovation assistant software. In Proceedings of the Second International Conference on Computational Creativity.

Meehan, J. 1976, The Metanovel: Writing stories by computer. Ph.D. Dissertation, Yale.

Riedl, M. O., and Sugandh, N. 2008. Story planning with vignettes: Toward overcoming the content production bottleneck. In Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling, ICIDS '08, 168–179. Berlin, Heidelberg: Springer.

Ross, K.; Klein, G.; Thunholm, P.; Scmitt, J.; and Baxter, H. 2004. The recognition-primed decision model. Military Review July-August, 6–10.

Surdu, J. R., and Kittka, K. 2008. The deep green concept. In Proceedings of the 2008 Spring simulation multiconference, SpringSim '08, 623–631. San Diego, CA, USA: Society for Computer Simulation International.

Tan, K.-M. T., and Kwok, K. 2009. Scenario generation using double-scope blending. In AAAI Fall Symposium.

Towler, M. 2010. Rational decision making: An introduction. Wiley.

Triantaphyllou, E. 2002. Multi-Criteria Decision Making Methods: A Comparative Study. Kluwer.

Yates, J.; Veinott, E.; and Patalano, A. 2003. Hard decisions, bad decisions: On decision quality and decision aiding. In Schneider, S., and Shanteau., eds., Emerging perspectives on judgment and decision research. Cambridge University Press. 13–63.

Nobody's A Critic: On The Evaluation Of Creative Code Generators – A Case Study In Videogame Design

Michael Cook, Simon Colton and Jeremy Gow
Computational Creativity Group, Imperial College, London

Abstract

Application domains for Computational Creativity projects range from musical composition to recipe design, but despite all of these systems having computational methods in common, we are aware of no projects to date that focus on program code as the created artefact. We present the Mechanic Miner tool for inventing new concepts for videogame interaction which works by inspecting, modifying and executing code. We describe the system in detail and report on an evaluation based on a large survey of people playing games using content it produced. We use this to raise issues regarding the assessment of code as a created artefact and to discuss future directions for Computational Creativity research.

Introduction

Automatic code generation is not an unusual concept in computer science. For instance, many types of machine learning work because of an ability to generate specialised programs in response to sets of data, e.g., logic programs (Muggleton and de Raedt 1994). Also, evolutionary systems can be seen to produce code either explicitly, in the case of genetic programming, or implicitly through evolutionary art software that uses programmatic representations to store and evaluate populations of artworks. Moreover, in automated theory formation approaches, systems such as HR (Colton 2002) generate logic programs to calculate mathematical concepts. These programs are purely for representation, however, rather than in pursuit of creative programming. In software engineering circles, 'metaprogramming' is used to increase developer efficiency by expanding abstract design patterns, or to increase adaptability by reformatting code to suit certain environments. None of these instances of code generation fully embrace the act of programming for what it is – a creative undertaking. There can be no field better placed to appreciate programming in this way than Computational Creativity.

Building software that can generate new software, or modify its own programming, opens up huge new areas for Computational Creativity, as well as enriching all existing lines of research by allowing us to reflect on our systems as potential artefacts of code generators or modifiers themselves. We attempt here to highlight some of these future opportunities and challenges by describing the design of a prototype system, *Mechanic Miner* (Cook et al 2013), which designs a particular videogame element – game mechanics – by inspecting, modifying and executing Java game code.

Mechanic Miner produced game mechanics for *A Puzzling Present*, a platform game released in December 2012 and downloaded more than 5900 times. This game included survey and logging code to assess, among other things, the quality of the mechanics generated by Mechanic Miner in terms of perceived enjoyability and the challenge in using them. In analysing the data and evaluating the system, however, we have noticed issues with current notions of assessment within Computational Creativity research, and how they interact with the idea of evaluating a creative system whose output is program code. We explore these issues below. In this paper we make the following contributions:

- We describe the development of a creative system that generates code as its output.
- We report on the first large-scale experimental evaluation of interactive computationally-created artefacts.
- We discuss issues involving the assessment of creative systems working in media with a high barrier to entry.

The rest of this paper is organised as follows: in *Mechanic Miner – Overview* we describe Mechanic Miner in full, detailing how it generates and evaluates new game mechanics through code. In *A Puzzling Present – Evaluation Through Play* we describe *A Puzzling Present*, a game designed and released using mechanics invented by Mechanic Miner. We discuss the difficulties in evaluating interactive code, how a balance can be struck between presenting a survey and offering a natural experience to the user, and present some results from our survey. In the section *Creativity in Code Generation*, we highlight issues for the future of code generation, as well as promising opportunities for Computational Creativity. in *Related Work* we briefly describe previous approaches to mechanic generation and highlight why code generation is necessary to advance in this area. Finally, in *Conclusions* we review our achievements and reflect on where our work with game mechanics will lead next.

Mechanic Miner – Overview

Definitions

Many conflicting definitions exist for game mechanics, as described, for instance, in (Sicart 2008), (Kelly 2010) and (Cook 2006). For our purposes here, we define a game mechanic as a piece of code that is executed whenever a button is pressed by the player that causes a change in the game's state. How a game mechanic is defined in code will vary

from game to game, depending on the architecture of the game engine, the way the game has been coded within that engine, and the idiosyncrasies of the individuals who wrote the rest of the game code. For example, below is a line of code from a game written in the *Flixel* game engine. Executing the code causes the player character to jump, by adding a fixed value to its velocity (the player's gravity will counteract this change over time and bring the character to the ground again).

```
player.velocity.y -= 400;
```

Mechanic Miner generates artefacts within a subspace of game mechanics, which we have called *Toggleable Game Mechanics* (TGMs). A TGM is an action the player can take to change the state of a variable. That is, given a variable v and a modification function f with inverse f^{-1}, a TGM is an action the player can take which applies $f(v)$ when pressed the first time, and $f^{-1}(v)$ when pressed a second time. The action may not be perfectly reversible; if v is changed elsewhere in the code between the player taking actions f and f^{-1}, the inverse may not set v back to the value it had when f was applied to it. For instance, if v is the player's x co-ordinate, and the player moves around after applying f, then their x co-ordinate will not return to its original value after applying f^{-1}, as it was modified by the player moving.

Generation Mechanic Miner is written in Java, and therefore able to take advantage of the language's built-in Reflection features that allow program code to inspect and explore other code[1]. For example, the following code retrieves a list of fields of a given class:

```
MyClass.getClass().getFields()
```

Such Field objects can be manipulated to yield their name, their type, or even passed to objects of the appropriate type to find the value of that field within the object. Java has similar objects to represent most other language features, such as `Method`s and generic types. Given the definition of a TGM above, we can see that Reflection allows software to store the location of a target field at runtime, and dynamically alter its value. Using the Reflections library, Mechanic Miner can therefore obtain a list of all classes currently loaded, and iterate through them asking for their available fields. It can use information on the type of each field to conditionally select modifiers that can be applied to the field.

Java's Reflection features do not provide encapsulation for primitive operations such as mathematical operators, assignment or object equality. To solve this problem, we created custom classes to represent these operations, which enabled Mechanic Miner to select modifiers for a field that could be applied during evaluation. Thus, a TGM is composed of a `java.lang.Field` object, and a type-specific `Modifier`. For example, a mechanic that doubled the x co-ordinate of the player object would use the `org.flixel.FlxSprite` object's `x` field, and an `IntegerMultiplyModifier` defined as follows:

[1]We further extended this core functionality by employing the Reflections library from http://code.google.com/p/reflections.

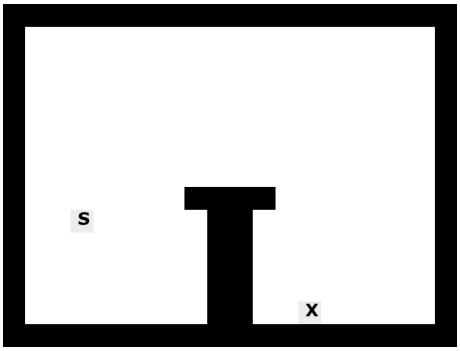


Figure 1: A sample level used to evaluate mechanics.

```
public void apply(Field f){
 if(toggled_on){
  f.setValue(f.getValue()*coefficient);
 } else{
  f.setValue(f.getValue()/coefficient);
 }
}
```

Where `coefficient` is set to 2 in the case of doubling, but can be set by Mechanic Miner to an arbitrary value as it evaluates potential mechanics. Note the use of a boolean flag, `toggled_on`, to retain the state of the TGM so that its effect can be reversed. Modifiers were selected to give a coverage of key operations that might be performed on fields, such as inverting the value of a boolean field, adding or multiplying values for a numerical field, or setting numerical fields to exact values (such as zeroing a field, and then returning it to its original value). Future extensions we plan to the generation process will allow for the use of mathematical discovery tools such as HR (Colton 2002) that could invent calculations which transform the values of the fields.

Evaluation In order to evaluate generated mechanics, we need strong criteria that describe the properties that desirable mechanics should have. In the version of Mechanic Miner described here, we focus purely on the *utility* of a mechanic (that is, whether it affords the player new possibilities when playing the game) rather than how fun the mechanic is to use, how easy it is to understand, or how appropriate it is for the context. Utility is not only easy to define, but can be defined in absolute terms, which provides a solid target for a system to evaluate towards.

To illustrate how utility can be identified by Mechanic Miner, consider the game level shown in Figure 1. The player starts in the location marked 'S' and must reach the location marked 'X', and when they do, we say that the player has *solved* the game level. The game operates similar to a simple game such as *Super Mario*; the player is subject to gravity, but can move left and right as well as jumping a small distance up. Under these rules alone, the level is not solvable because the central wall is too high and impedes progress. Therefore, if we were to add a new game

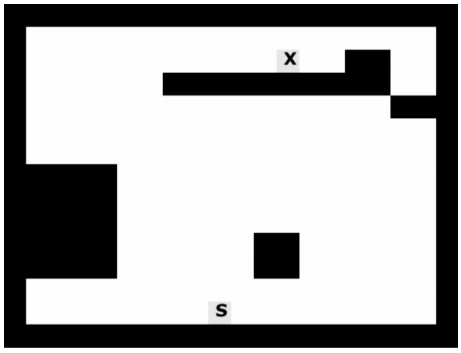


Figure 2: A level generated by Mechanic Miner for the 'gravity inversion' mechanic. The player starts in the 'S' position and must reach the exit, marked 'X'.

mechanic such as the inversion of gravity, and as a result the level were to become solvable, we could conclude that the new mechanic had expanded the player's abilities, and allowed them to solve a level of this type.

This idea is central to Mechanic Miner's evaluation of mechanics – it uses a solver to play game levels in a breadth-first fashion, trying legal combinations of button presses while remaining agnostic to what mechanics the buttons relate to. It will continue to search for combinations of button presses until it finds at least one solution; at this point it continues looking for combinations of that length, completing the breadth first expansion of this depth, and will then return a list of all paths that led to a solution. Hence it can try arbitrary mechanics without knowing in advance what the associated code does when executed. This enables it to firmly conclude whether the mechanic has contributed to the player's abilities by assessing which areas of the level are accessible that were not previously, which in turn enables it assess the level itself.

Level Generation Mechanic Miner's ability to simulate gameplay in order to evaluate mechanics can also be applied in reverse to act as a fitness function when generating levels for specific mechanics using evolutionary techniques. Representing a level design as a 20x15 array of blocks that are either solid or empty, we can evaluate the fitness of a level with respect to a mechanic M by playing the level twice – once with only the basic controls available, and once with M added to the controls. If the level is solvable with M, but *not* solvable without it, then the level is given a higher fitness. Using a binary utility function as our primary evaluation criteria strengthens the system's ability to provide exact solutions to the problem – either the level is completed, or it is not. In order to have a gradient between the two so that the evolutionary level designer can progress towards good levels, we moderate the fitness based on what proportion of the level was accessible. Thus, over time, levels that are more accessible emerge until eventually the exit is reachable from the start position (and thus the level is solvable).

Figure 2 shows a level generated for use with a mechanic called *gravity inversion*. Activating the mechanic would cause gravity to pull the player towards the ceiling instead of the floor. Activating it again would reverse the effect. Note that the level is not solvable without this mechanic, as the platforms are too high to jump onto.

The simulation-driven approach to level design allowed for the resulting software to be highly parameterised. Information such as the minimum number of distinct actions required to solve a level (where each button press is considered a distinct action) or the number of times a mechanic must be used, allowed the system to generate levels with different properties. It also allows the system to remain blind to the mechanic it is designing for. This allows Mechanic Miner to exploit created mechanics without having a human intervene and describe aspects of the mechanic to it, giving it greater creative independence as it is theoretically able to discover a wholly new mechanic in a H-creative way, and generate levels for that mechanic without any assistance. We can view this within the creativity tripod framework of (Colton 2008), which advocates implementing skill, appreciation and imagination in software. In particular, we see the ability to use output from one system to inspire creative work in another without external assistance as an example of skill as well as an appreciation of what makes a game mechanic useful to the player. We also claim that simulating player behaviour is in some sense *imagining* how they would play.

Illustrative Results

Below are examples of mechanics generated by Mechanic Miner. All of the effects can be reversed by the player:

- An ability to increase the player's jump height, allowing them to leap over taller obstacles.
- An ability to rubberise the player, making them able to bounce off platforms and ceilings.
- An ability to turn gravity upside down, sucking the player upwards.

These mechanics are evident in commercially successful games, such as Cavanagh's *VVVVVV* which featured gravity inversion as a core mechanic. Bouncing was an unexpected result for us, as we had no idea it was in the space of possibilities, although it has been featured in some games developed in other engines, particularly Nygren's *NightSky*. Cavanagh has received multiple nominations in the Independent Games Festival (IGF), and NightSky was shortlisted for Excellence In Design and the Grand Prize in the 2009 IGF.

Novel game mechanics are highly prized in game design circles. Many international design awards have tracks for innovative gameplay or mechanics (such as the IGF Nuovo Award[2]) and game design events often centre around the creation of unique methods of interaction (such as the Experimental Gameplay Workshop[3]). Mechanic Miner's ability to reinvent existing but niche mechanics is encouraging, given the small design space the system currently has access to.

As well creating mechanics, Mechanic Miner was also able to find exploits in the supplied game code, and use

[2]http://www.igf.com/

[3]http://www.experimental-gameplay.org/

them to create emergent gameplay – something which we had not anticipated as a capability of the system. One mechanic, which teleported the player a fixed distance left or right, was used by Mechanic Miner to design levels which at first glance had no legal solution. After inspecting the solution traces produced by the simulator, it became clear that the mechanic was being used in an innovative way to take advantage of a weakness in the code that described the player's jump. Jumping checked if the player's feet were in contact with a solid surface. By teleporting inside a wall, this check would be passed, and the player could jump upwards. Repeated applications of this technique allowed the player to jump up the side of walls – complicated exploitation of code, more commonly seen in high-end gameplay by *speedrunners*[4], i.e., gamers who compete over finding exploits in popular videogames to help them complete the games in the shortest time possible. For example, speed runs of the popular puzzle game *Portal* involve the abuse of 3D level geometry to escape the level's boundaries and pass through solid walls.

A Puzzling Present - Evaluation Through Play

To evaluate some of the mechanics and levels designed by the Mechanic Miner system, we developed a short compilation game featuring hand-selected mechanics, titled *A Puzzling Present* (APP). APP was released in late December 2012 on the Google Play store and desktop platforms[5]. The objective was to conduct a large-scale survey of players in order to gain feedback on the types of mechanic generated by the system, in addition to evaluating different metrics for level design. However, we were also conscious that interruptions to play, or overt presentation of the software as an experiment rather than a game, may deter players from completing levels or giving feedback and/or change the nature of the experiment, which is to ask their opinion on games, not surveys. In designing APP, we therefore made several trade-offs to balance these two factors.

All play sessions were logged in terms of which buttons the player presses, at what times, which can be used to fully replay a given player's attempt at a level. In addition to this, upon starting the game for the first time, the player was asked to opt-in to short surveys after each level. These took the form of two multiple-choice rating tasks on a 1-4 scale, evaluating enjoyability and difficulty. Figure 3 shows the survey screen. This presented itself to the player upon reaching the exit to a level, assuming the player had agreed to respond to surveys, although even in this case, they could continue without responding to the survey.

75614 sessions were recorded in total, over 5933 unique devices. When asked to opt-in to surveys, 60.7% of users agreed. Those who opted-in contributed 63.4% of the total session count. 92.3% of sessions played by opt-ins resulted in at least one of the two questions being answered, with 89.9% of sessions resulting in both questions being answered. Although the survey questions provided a rich source of data, by allowing us to gain qualitative evaluations

Figure 3: Survey screen from *A Puzzling Present*

of the levels and game mechanics, the log data (which is recorded for all players) is equally valuable, and so allowing players who did not wish to participate in the survey to continue to play the game (or those who initially agreed to change their minds later) we gained an additional 32,000 level traces which we otherwise might have lost.

APP contained thirty levels, split into sets of ten that share a common mechanic. The three game mechanics are those described in the *Illustrative Results* section above: higher jump, bouncing and gravity inversion. Each level required the game mechanic to be used to complete it, but were generated using differing metrics for difficulty expressed through evolutionary parameters within the level designer. These were broken down as follows: two levels used a baseline setting determined through experimentation ('Baseline'); two levels put stricter requirements on minimum reaction times needed ('Faster Reaction'); two levels selected for longer paths from start to exit ('Longer Path'); two levels selected for more mechanic use in the shortest solutions ('Higher Mechanic Use'); and two levels selected for longer action chains in the solution. This provided a variety of the levels for the player to test, and allowed us to analyse feedback data to assess these metrics for future use. In order to mitigate bias or fatigue introduced as a result of experiencing certain levels or sets of levels before others, the order in which a particular player experienced the levels was randomised when the game was first started up. This was done by first randomising the order of the game mechanics, and then randomising the order of the ten levels within that set, thereby ensuring that all levels which share a mechanic are experienced together, to provide a more cohesive experience.

Figure 4 shows the mean difficulty and fun ratings for the nth level played as the people progressed through the 30 levels. These mean ratings remained fairly consistent throughout the game, with the exception of the 30th level. As levels were presented randomly, we assume this is an effect of the very low number of people still playing at this point. This consistency indicates that learning or fatigue did not seem to have much effect on player experience. This may be down to the interactivity of the artefact in question, and raises the question of whether the evaluation of created artefacts is more consistent when the survey participants are interactively engaged. We discuss this later as future work.

[4]Such as the community at http://speeddemosarchive.com/

[5]Download from www.gamesbyangelina.org/downloads/app.html.

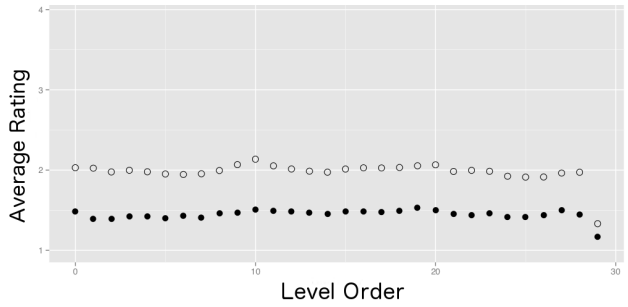


Figure 4: Mean fun (white circles) and difficulty (black circles) ratings for the nth level played.

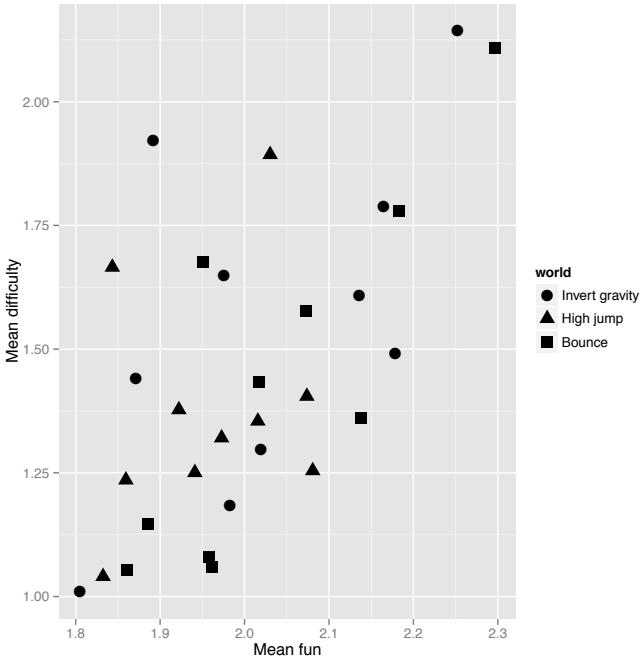


Figure 5: Mean level fun and difficulty, broken down by 'world' (a group of levels that share a mechanic).

The number of players completing a given set *(world)* of ten levels for a certain mechanic is consistent across the three game mechanics; 2259 completed World one, 2151 completed World two and 2219 completed World three. The data show no bias towards players not completing any particular one of the three worlds, suggesting that players left due to general fatigue with the system as a whole, rather than the content generated by Mechanic Miner. This may be down to the human-designed elements of the game that were common throughout the three worlds – such as the interface, control scheme, or artwork – and therefore not attributable to the output of Mechanic Miner.

Under statistical analysis of the survey scores, we found a moderate and highly significant rank correlation between mean difficulty and enjoyability (Spearman's $\rho = 0.56$, $p = 0.002$). The relationship between the difficulty of a level and the perceived enjoyability of a level is an interesting one to consider. While we might expect an inverse relationship for an audience who are easily frustrated with games, we also see many examples of games in which challenge correlates to an enjoyable game. We postulate that the correlation between mean difficulty and enjoyability exists here because the levels are, on average, *too* easy – the average difficulty rating across all levels is just 1.45, on a scale of 1 to 4 – and so an increase in difficulty was welcomed as it made the levels more interesting. A later study, with improved difficulty metrics to give a broader spread of skill levels, would help confirm this hypothesis.

Group	Mean Fun	Mean Difficulty
High Jump	1.96	1.38
Invert Gravity	2.02	1.55
Bounce	2.03	1.42
Baseline	1.96	1.30
Faster Reaction	2.01	1.51
Longer Path	1.95	1.20
Higher Mechanic Use	2.03	1.60
Longer Solution	2.06	1.66

Figure 6: Mean level fun and difficulty, broken down by game mechanic and level design parameters.

The mean fun and difficulty by world mechanic and level generation metric are shown in Table 6. Variations in mean fun are very small between groups, whereas mean difficulty shows greater separation, especially between the metrics. An analysis of variance (ANOVA) showed highly significant ($p < 0.001$) separate main effects for fun and difficulty with respect to both factors. There was also a significant interaction between mechanic and metric, which we do not report here. Post-hoc Tukey's HSD tests suggested the following significant differences between groups: a) the mechanics Invert Gravity and Bounce are more fun than High Jump; b) the metrics Fast Reaction, High Use and High Actions are more fun than Baseline and Longer Path; c) all differences in mean difficulty between mechanics, and between metrics, are significant.

Creativity In Code Generation

Nobody's a Critic

Many different approaches to assessing creativity in software have been proposed over the last decade of Computational Creativity research. Ritchie (2007) suggests that the creativity of a system might be established by considering what the system produces, evaluating the artefacts along such lines as *novelty*, *typicality* and *quality*. This leads to the proposal of ratios between sets of novel artefacts produced by a system, and sets that are of high quality, for instance. While this is helpful in establishing the performance of a given system, it presupposes both a minimum level of understanding in those assessing the system, and a direct connection between the means of interaction with the artefact, and the generated work itself.

In the case of software – particularly interactive media

whose primary purpose is entertainment – we are not guaranteed either of these. The consumers of software, such as those that evaluated *A Puzzling Present*, are often laypeople to the world of programming, even if they are highly experienced in interacting with software. More importantly, there is a disconnect between the presentation of code through its execution within a game environment, and the nature of the generated code itself. All software designed for use by the general public – from word processors to video games – presents a metaphorical environment in which graphics, audio and systems of rules come together to present a cohesive, interactive system with its own internal logic, symbols, language and fiction. In *A Puzzling Present* in particular, generated game mechanics operated on obscure variables hidden away within a complex class structure. To the interacting player, this is simply expressed as objects moving differently on-screen. This disconnect makes it hard for any user to properly evaluate the generated code itself, because they are not engaging with the underlying representation or mechanics of the software they are using.

Other approaches to evaluation consider the process of creation itself as crucial to the perception of creativity. In (Colton, Charnley, and Pease 2011) the authors propose the FACE model that considers elements of the creative process such as the generation of contextual information (which the authors call *framing*) and the use and invention of aesthetic judgements that affect creative decision-making. This focus on the process is a promising alternative to the artefact-heavy assessment methods that are more common in Computational Creativity, but problems abound here also, since in order to judge the creative process, a person must be able to comprehend that process to some degree.

As noted in (Johnson 2012), the majority of the systems in Computational Creativity have focused on 'old media' application domains, such as the visual arts, music and poetry. Although the skill ceiling for these media is undeniably high, they have very low barriers to entry. Most people have drawn pictures as children, attempted to crack new jokes, or hummed improvised ditties to themselves. While they may not exhibit even a small percentage of the virtuosity present at the top end of the medium in question, by engaging in the creation of artefacts, they can appreciate the process and are better placed to comment on it – or indeed they feel so, even if this is not the case. As a result, creative systems operating in the realm of old media often find truth in the term 'everyone's a critic'. By contrast, programming is a skill that is only recently being taught below university level in the western world; therefore, asking the general public to assess the creativity of a code generator by commenting on its creative process is unlikely to result in a useful or fair assessment.

This phenomenon – where nobody is a critic – makes it hard to apply existing thinking on the evaluation of creative systems to large-scale public surveys.

Speaking In Code

If neither the artefact-centric nor the process-centric approach is suitable to assess creative code generators, this begs the question of how we can proceed in assessing these systems on a large scale. We believe the key may be one particular element of the model described in FACE model of (Colton, Charnley, and Pease 2011), namely *framing* information that describes an artefact and the process that created it, as explored further in (Charnley, Pease, and Colton 2012).

Figure 7: Framing information in *Stealth Bastard*.

Code is not designed to be read by people. Extensive education is needed to understand the basics of programming structure and organisation, including additional time spent on learning specific languages. Even experienced programmers do not rely on these skills alone to understand program code – instead they leave plain-English comments so that others, and they themselves, will be able to understand the meaning of code long after it has been written. In interactive media, the need to explain features legibly and correctly situated within the (possibly fictional) context of the software is especially integral to the user's understanding and enjoyment of a piece of software. Video games, for instance, rely on their ability to create an immersive environment where all functionality is communicated through the fiction of the game world in question. The arcade game *Space Invaders* is not about co-ordinates overlapping and numbers being decremented – it is about shooting missiles at aliens and protecting your planet from attack.

This all amounts to a clear need to build into creative code generation systems the ability to explain the function of code it produces. This could be done either by annotating and describing the function of the raw code itself or, in the case of presenting artefacts to a layperson for assessment or consumption, by describing the function of the code in terms of the metaphors and context dictated by the software the code is part of. In the latter case, this poses interesting problems more akin to creative natural language generation. Videogames, for example, must describe the functionality of game mechanics in terms of what they enable the player to do within the game world - Figure 7 shows the *Stealth Bastard* game (Biddle 2012) explaining how to complete a level. Note the use of a physical verb (*enter*), a symbolic noun (*exit*) and a reference to meta-game objectives (*completing a level*). These are concepts unrelated to the technical specifics of game code, but crucial to the player's understanding of the thematic and ludic qualities of the game.

The generation of textual descriptions of both the creative process *and* the generated code is crucial in enabling these systems to be assessed by the general public. It will also become more important in autonomously creative systems that generate code for use in interactive contexts, where the meaning of the code must be conveyed clearly to a user. This

is a highly-prized feature of human-designed software[6] and is crucial in autonomously creative systems where artefacts are not subject to curation prior to their use.

Beyond Software

Considering program code as an artefact produced by a creative system allows us to reconsider existing creative systems as potential code generators themselves. Modules within creative systems might be able to integrate criteria such as those described in (Ritchie 2007) into a process of self-exploration and modification – where new code is created for generative submodules, and evaluated according to its ability to produce content along axes such as novelty, typicality or quality. Code generation should not be thought of, therefore, as a distinct strand of Computational Creativity that runs alongside other endeavours in art, poetry and the like. Instead, it should be viewed as a new lens through which to view existing takes on Computational Creativity, and a new way to improve the novelty and ingenuity of creative systems of all kinds.

If generic notions of novelty or typicality for code can be developed, then they can be applied across mediums to great effect. Comparisons of code segments have been explored within verification and software engineering approaches (Bonchi and Pous 2012; Turon et al. 2013), but for the purposes of Computational Creativity, a significantly different approach will be required, as we consider the ludic, aesthetic and semantic similarities in the output of a piece of code, rather than its raw data. If this can be done, creative software will no longer need to be considered static, instead empowered with the ability to generate new functionality within itself; creative artefacts will no longer need to be considered as finished when they leave a piece of software, but could improve and iterate upon their designs in response to use; and creative software will no longer be considered simply executing code written by humans, but instead be seen to be a collaborator in its own creation.

Related Work

The generation of game mechanics is closely related to the design of game rules in the more abstract sense. METAGAME (Pell 1992) is an early example of a system that attempted to generate new game rulesets. This worked by varying existing rulesets from well-known boardgames such as chess and checkers, using a simple grammar that could express the games as well as provide room for variation. Grammar-based approaches to ruleset generation are common in this area, perhaps most prominently seen in Ludi (Browne and Maire 2010) which evolved boardgame rulesets from a grammar of common operations, or work in (Togelius and Schmidhuber 2008) and (Cook and Colton 2012) which present similar work for realtime videogames.

Grammar-based approaches work well because they explore spaces of games that are defined by common core concepts; but are naturally limited by the nature of the human-designed grammar as a result. An alternative approach that can cover a broader space is to use annotated databases of mechanical components, and then assemble them to suit a particular design problem. Work in (Nelson and Mateas 2007) uses this approach to design games around simple noun-and-verb input, while (Treanor et al. 2012) use an annotated database approach to develop games that represent a human-defined network of concepts.

Smith and Mateas (2010) present an alternative approach, describing a generator of game rulesets without an evaluative component. The system they describe uses answer set programming to define a design space through a set of logical constraints. Solutions to these constraints describe game rulesets, therefore if constraints are chosen to restrict solutions to a certain space of good games, solving them will yield high-quality games. These criteria can be narrowed down by adding further constraints to the answer set program. This can be seen as somewhat related to grammatical approaches – higher-level concepts are defined by hand (such as 'character movement' or 'kill all') which are then selected for use later. This has similar limitations to the grammatical approaches, in that it is dependent on external input to define its initial language, and that this restricts the novelty of the system as a result. The future work proposed in (Smith and Mateas 2010) was to focus more on programmatic modification, however, which would have further distinguished the approach.

Conclusions and Further Work

We have described Mechanic Miner, a code modification system for generating executable content for videogames, and A Puzzling Present, a game which we released built using content generated by Mechanic Miner. We showed that code can be used as both a source material and a target domain for Computational Creativity research, and that it can lead to greater depth than working with metalevel abstractions of target creative domains, offering surprise and novelty even on a small scale. Through evaluation of gameplay responses, we drew conclusions about the presentation of creative artefacts to large audiences for evaluation. Finally, we raised the issue of how created artefacts can be evaluated by an audience which, in general, has no experience in the domain the artefacts reside within.

This work has also highlighted several areas of future work needed to expand the concepts behind Mechanic Miner to prove the worth of the approach in generating more sophisticated mechanics and games. These include work to expand the expressiveness of the code generation, so that it can include higher-level language concepts such as method invocation, expression sequences, control flow and object creation. This will lead to a large expansion of the design space, which will raise issues of efficiency and evaluation, also bearing further investigation.

We will also be using our experimental results to tune both our existing metrics for level and mechanic design, and to drive further development in systems such as Mechanic Miner, to increase their autonomy and their ability to seek out novel content. We are particularly interested in how different difficulty metrics can be combined to produce a diverse set of game content.

[6] E.g., as promoted in Apple's *Human Interface Guidelines*

We will also consider the looming problem of code generation's relationship with metaphorical gameplay. Game designer and critic Anna Anthropy describes games as "an experience created by rules" (Anthropy 2012). The way in which this experience is created, however, is deeply grounded in the player's ability to connect the systems inside a game with the real world. In *Super Mario*, for instance, eating a mushroom makes you larger, and conveys extra speed and jumping power. In the game's code, this is simply a collision of two objects, and some state changes. Notions such as size visually indicating strength or ability, or the idea that consuming food can improve your strength, are fundamentally connected to real-world knowledge, and less evident simply by looking at code. Discovering ways that software can discover these relationships for itself will be a major hurdle in developing code generators capable of designing meaningful game content, but also a gateway to an unprecedented level of creative power for software, and an opportunity to bring art, music, narrative and mechanics together in a more meaningful way than ever before.

The field of Computational Creativity was founded on the belief that computers could be used to simulate, enhance and investigate aspects of creativity, and researchers have created many complex pieces of software by hand. We believe that the time is ripe to move this a step further, and to turn the ideas we have developed on our own creations; to reconsider our artificial artists, composers and soup chefs as pieces of code that can be assessed, altered and improved at the same level of granularity that they were created. In order to do so, however, we may need to challenge some assumptions we hold about certain creative mediums and the relationship the general public has with them.

Acknowledgements

We would like to thank the reviewers for their input and suggestions, particularly regarding the discussion section.

References

Anthropy, A. 2012. *Rise of the Videogame Zinesters: How Freaks, Normals, Amateurs, Artists, Dreamers, Drop-outs, Queers, Housewives And People Like You Are Taking Back An Art Form*. Seven Stories Press.

Biddle, J. 2012. Stealth bastard deluxe. Digitally distributed via http://www.stealthbastard.com.

Bonchi, F., and Pous, D. 2012. Checking NFA equivalence with bisimulations up to congruence. In *Proceedings of the 40th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '13.

Browne, C., and Maire, F. 2010. Evolutionary Game Design. *IEEE Transactions on Computational Intelligence and AI in Games* 2(1):1–16.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in Computational Creativity. In *Proceedings of the Third International Conference on Computational Creativity*.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational Creativity Theory: The FACE and IDEA models. In *Proceedings of the Second International Conference on Computational Creativity*.

Colton, S. 2002. *Automated Theory Formation in Pure Mathematics*. Springer.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposiumo on Creative Intelligent Systems*.

Cook, M., and Colton, S. 2012. Initial results from co-operative co-evolution for automated platformer design. In *Proceedings of the Applications of Evolutionary Computation (EvoGames workshop)*.

Cook, M.; Colton, S; Raad, A; and Gow, J 2013. Mechanic Miner: Reflection-Driven Game Mechanic Discovery and Level Design. In *Proceedings of the Applications of Evolutionary Computation (EvoGames workshop)*.

Cook, D. 2006. What are game mechanics? http://www.lostgarden.com/2006/10/what-are-game-mechanics.html.

Johnson, C. 2012. The creative computer as romantic hero? or, what kind of creative personae do computational creativity systems exemplify? In *Proceedings of the Third International Conference on Computational Creativity*.

Kelly, T. 2010. Game dynamics vs game mechanics. http://www.whatgamesare.com/2010/12/game-dynamics-vs-game-mechanics.html.

Muggleton, S., and de Raedt, L. 1994. Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19(20).

Nelson, M. J., and Mateas, M. 2007. Towards automated game design. In *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence*.

Pell, B. 1992. Metagame in symmetric, chess-like games. In *Heuristic Programming in Artificial Intelligence 3: The Third Computer Olympiad*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Sicart, M. 2008. Defining game mechanics. *Game Studies* 8(2).

Smith, A., and Mateas, M. 2010. Variations forever: Flexibly generating rulesets from a sculptable design space of mini-games. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*, 273–280.

Togelius, J., and Schmidhuber, J. 2008. An experiment in automatic game design. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*.

Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the Third Workshop on Procedural Content Generation in Games*.

Turon, A. J.; Thamsborg, J.; Ahmed, A.; Birkedal, L.; and Dreyer, D. 2013. Logical relations for fine-grained concurrency. In *Proceedings of the 40th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '13.

A model for evaluating interestingness in a computer-generated plot

Rafael Pérez y Pérez, Otoniel Ortiz
Departamento de Tecnologías de la Información
Universidad Autónoma Metropolitana, Cuajimalpa
Av. Constituyentes 1054, México D. F.
{rperez/oortiz}@correo.cua.uam.mx

Abstract

This paper describes a computer model for evaluating the interestingness of a computer-generated plot. In this work we describe a set of features that represent some of the core characteristics of interestingness. Then, we describe in detail our computer model and explain how we implemented our first prototype. We assess four computer-generated narratives using our system and present the results. For comparison reasons, we asked a group of subjects to emit an opinion about the interestingness of the same four stories. The outcome suggests that we are in the right direction, although much more work is required.

Introduction

Evaluation is a core aspect of the creative process and if we are interested in building creative systems we need to develop mechanisms that allow them to evaluate their own outputs. The purpose of this project is to contribute in that direction.

This paper describes a model for evaluating the interestingness of a computer generated plot. It is part of our research project in computer models of narrative generation. Some time ago we developed a computer model of narrative generation (Pérez y Pérez and Sharples 2001; Pérez y Pérez 2007). Our model distinguished three core characteristics: coherence, novelty and interestingness. To test our model we built an agent that generated plots. Now, we are interested in developing a model to evaluate the coherence, novelty and interestingness of a computer-generated narrative. So, our storyteller agent will be able to evaluate its own outputs. In this way, we expect to understand better how the evaluation process works and, as a consequence, how the creative process works. Due to space limitations this document only discusses the central features of our model for the evaluation of interestingness (the reader can find some published work describing the main characteristics of our model for evaluation of novelty in Pérez y Pérez et al. 2011).

We are aware that human evaluation of interestingness is a very complex task and we are far from understanding how it works. Nevertheless, we believe that computer models, like the one we describe in this text, can provide some light in this challenging aspect of human creativity.

Related Work

There have been several discussions about how to assess computational creativity. For example, Ritchie (2007) suggests criteria for evaluating the products of a creative process (the process is not taken into consideration); in general terms such criteria evaluate how typical and how valuable the product is. Colton (2008) considers that skill, imagination and appreciation are characteristics that a computer model needs to be perceived to have. Jordanous (2012) suggests to have a set of human experts that evaluate characteristics like Spontaneity and Subconscious Processing, Value, Intention and Emotional Involvement, and so on, in a computer generated product. All these are interesting ideas, although some are too general and difficult to implement (e.g. see Pereira et al. 2005). Some work has been done in evaluation of plot generation:

> A computer model might be considered as representing a creative process if it generates knowledge that does not explicitly exist in the original knowledge-base of the system and which is relevant to (i.e. is an important element of) the produced output. Note that this definition involves inspection of both the output of the program and its initial data structures... we refer to this type of creativity as computerised creativity (c-creativity) (Pérez y Pérez and Sharples 2004).

Peinado et al. (2010) also have worked in evaluation of stories, although they work was oriented to asses novelty. An area that some readers might consider related to this work is interactive drama and drama managers. A good example of this type of systems is the work by Weyhrauch (1997). However, rather than evaluating the plot and the creative process, Drama managers focus in evaluating the user's experience while playing the game. Some other systems might employ different techniques, e.g. case base systems (Sharma et al. 2010), but the goal is the same: to provide a pleasant experience to the user.

Description of the Model

This work describes a model to evaluate the interestingness of a computer generated plot. Such a plot is known as the new story or the new narrative. For the purpose of this project, we consider a narrative interesting when it is recounted in a correct manner and when it generates new knowledge. A story is recounted in a correct manner when it follows the classical Aristotelian structure of a story: introduction, development, climax and resolution (or setup, conflict and resolution). Some previous work has shown the relation between the Aristotelian structure

and the evaluation of interestingness in computer generated plots (Pérez y Pérez and Sharples 2001). We are particularly interested in evaluating the opening and the closure of a story. We consider that a story has a correct opening when at the beginning there are no active dramatic tensions in the tale and then the tension starts to grow. We consider that a story has a correct closure if all the dramatic tensions in the story are solved when the last action is performed. An important characteristic of the recountal of a story is the introduction of unexpected obstacles. In this work an obstacle is unexpected when the story seems to finish (final part of the resolution section) and then new problems arise.
Following Pérez y Pérez and Sharples (2004) we believe that the generation of new knowledge contributes to consider a narrative interesting. Some studies in motivation, curiosity and learning seem to support this claim (e.g. see Deckers 2005). In the same way, writers have pointed out how good narratives are a source of new knowledge (e.g. see Lodge 1996). In this work a new story generates new knowledge when:

- It generates knowledge structures that did not exist previously in the knowledge base of the system and that can be employed to build novel narratives.
- It generates a knowledge widening, i.e. when existing knowledge structures incorporate unknown information obtained from the new story. This information can be employed to build novel narratives.

Our computer model of evaluation is based on expectations. So, the assessment of the new knowledge structures and the knowledge widening is performed by analysing how much the new story modifies the knowledge base; then, comparing if such modifications satisfied the given expectations. In the same way, the evaluations of unexpected obstacles and the correctness of the narrative's recountal are performed by analysing the structure of the new narrative; then, assessing if such a structure fulfils there expectations. Finally, all these partial results are considered to obtain a final evaluation of interestingness. The following lines elaborate these ideas.

Generating Original Structures

One of the key aspects of c-creativity is the generation of novel and relevant knowledge structures. That is, a storyteller must develop narratives that increment its knowledge base (in this work we focus on how the knowledge base of the evaluator is incremented). Thus, a storyteller must include mechanisms that allow: 1) incorporating within its knowledge base the new information generated by its outputs, i.e. it must include a feedback process; 2) comparing its knowledge base before and after feeding back a new tale (an interesting point for further discussions is to compare the processes that different systems might employ to perform these tasks).
In this way, the first part of the model focuses in determining the proportion of new structures. It requires a parameter known as the Minimal Value of New Structures (Min-NS); it represents the minimum amount of new structures expected to be created by the new story. In this way, the Proportion of New Structures (PNS) is defined by the ratio between the number of new structures (NNS) created by the new narrative and the Minimal Value of New Structures (Min-NS). If the number of new structures is bigger than its minimal value, the Proportion of New Structures is set to 1.

$$PNS = \begin{cases} \dfrac{NNS}{\text{Min-NS}} & \text{IF } NNS \leq \text{Min-NS} \\ 1 & \text{IF } NNS > \text{Min-NS} \end{cases}$$

Besides calculating the number of new structures, it is necessary to determine how novel they are, i.e. to verify if they are similar to the information that already exists in the knowledge base. With this purpose we define a parameter known as the Limit of Similitude (LS) that represents the maximum percentage of alikeness allowed between two knowledge structures.
So, all those new structures that are too alike to already existing structures must be eliminated. In other words, one must get rid of all new structures that are at least LS% equal to any existing structure. The number of surviving structures is known as the Original Value (O-Value) and they represent new structures that are not similar to any old structures. Like in the previous case, the model requires an expected Minimum Original Value (Min-OV) to calculate the Proportion of the Original Value (POV). And, like in the previous case, this proportion never can be bigger than 1.

$$POV = \begin{cases} \dfrac{\text{O-Value}}{\text{Min-OV}} & \text{IF O-Value} \leq \text{Min-OV} \\ 1 & \text{IF O-Value} > \text{Min-OV} \end{cases}$$

So, POV represents in which percentage the new narrative satisfies the expected number of original new structures.
The Novelty of the Knowledge Structures (NKS) is defined as the ratio between the O-Value and the number of new structures (NNS).

$$NKS = \frac{\text{O-Value}}{NNS}$$

It represents which percentage of the new structures is original. In this way, if the O-Value is identical to the number of new structures the NKS is equal to 1 (100%). That means that all new structures satisfy the requirement of novelty.
A variant of the process of creation of knowledge structures is known as knowledge widening. It occurs when existing knowledge structures incorporate within its own structure unknown information obtained from the new story. This concept is inspired by Piaget's ideas about accommodation and assimilation (Piaget 1952). So, the

model requires knowing the number of unknown information incorporated into the knowledge base; we refer to it as the number of new elements. So, in order to calculate the Proportion of Knowledge Widening (PKW) it is necessary to know the Number of New Elements (NNE) and an expected Minimum value of New Elements (Min-NE).

$$PKW = \begin{cases} \dfrac{NNE}{Min\text{-}NE} & \text{IF } NNE \leq Min\text{-}NE \\ 1 & \text{IF } NNE > Min\text{-}NE \end{cases}$$

Thus, PNS, POV, NKS and PKW provide information to evaluate how much new knowledge is generated.

Analysing the Story's Structure

We defined earlier that a story is recounted in a correct manner when it follows the classical Aristotelian structure: setup, conflict and resolution. The story's structure in this work is represented by the graphic of the curve of the dramatic tensions in the tale. Tensions represent conflicts between characters. When the number of conflicts grows the value of the tension rises; when the number of conflicts decreases the value of the tensions goes down; when the tension is equal to zero all conflicts have been solved. Thus, we analyse the characteristics of the graphic of tension to evaluate the presence of unexpected obstacles and how well recounted the story is. In this way, our evaluation model requires a mechanism to depict the dramatic tension in the tale.

There are four basic cases of graphics of tensions that we consider in this work: one complete curve (see figure 1-a); several complete curves (see figure 1-b); one incomplete curve (see figure 1-c); several incomplete curves (see figure 1-d). It is also possible to find combinations of these cases. A curve is defined as complete when its final amplitude is zero; that is, all tensions are resolved. By contrast, the final amplitude of an incomplete curve never gets the value of zero.

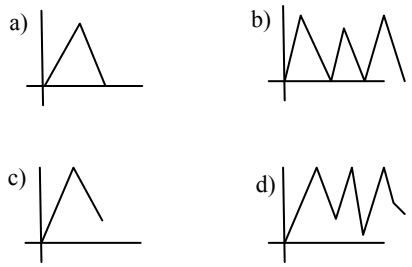


Figure 1. Examples of graphics of tensions.

The peak of a curve represents the climax of a narrative; if we have a sequence of curves we refer to the peak with the highest amplitude as the main climax. So, in a sequence, first the story reaches a situation with high levels of tensions, after that tensions start to loosen up and then they rise again; this cycle can be repeated. Each peak is a climax; each loosen up is a resolution of such a climax. We refer to the situation where a narrative has a resolution and then tensions start to rise again as *reintroducing-complications*.

We can find variations of the basic graphics of tensions we enumerated earlier. For example, the deepness of each valley in a sequence of incomplete curves might be different for each instance; in the same way, the amplitude of the peaks of sequences of complete or incomplete curves might change between them; and so on.

The difference between having a single curve and having a sequence of curves is that in the former there is only one high point in the story while in the latter we have two or more high points, i.e. new characters' obstacles are initiated reintroducing in this way complications.

The difference between a sequence of complete curves and a sequence of incomplete curves is that in the former all tensions are solved before new tensions arises; in the later new tensions emerge before the current ones are worked out. An incomplete curve is very similar to a complete curve if the fall of the tensions is close to 100% with respect to its peak, i.e. if the amplitude is close to the value of zero. On the other hand, if the fall of the tensions is close to 0% with respect to its peak, i.e. if the amplitude is close to the value of its peak, we practically do not have an incomplete curve. In this work we appreciate narratives that seem to end and then reintroduce new problems for the characters. In other words, we want narratives where all tensions are solved (complete curves) or are almost solved (incomplete curves with deep valleys) and then they rise again. This formula can be observed in several examples of narratives like films, television-series and novels (nevertheless, the model allows experimenting with different values of valley's profundity).

Thus, different graphics of tensions produce different characteristics in the narrative. We hypothesize that a story that includes more curves of tensions is more exciting than a story that includes fewer curves because the former reintroduces more complications. However, too many curves make the story inadequate. So, it is necessary to find a balance. In this way, our model requires to set a number that represents the perfect amount of complete curves that a story should comprise. We refer to this number as the Ideal Value of Complete Curves (Ideal-CC). So, because we can calculate the number of complete curves (Num-CC) in any new narrative and because we have defined an ideal number for them, it is possible to estimate how close the number of curves is to its ideal value. We refer to this number as the Proportion of Complete Curves (PCC):

$$PCC = \begin{cases} \dfrac{Num\text{-}CC}{Ideal\text{-}CC} & \text{IF } NumCC \leq Ideal\text{-}CC \\ 1 - \dfrac{Num\text{-}CC}{Ideal\text{-}CC} & \text{IF } Ideal\text{-}CC < NumCC \leq Ideal\text{-}CC \cdot 2 \\ 0 & \text{IF } NumCC > Ideal\text{-}IC \cdot 2 \end{cases}$$

It is important to explain how the NUM-CC is calculated. As it is going to be explained some lines ahead, a story must include at least one complete curve to be considered as properly recounted. But this curve itself does not reintroduce problems. The reintroduction of complications occurs when the current ones are sorted out and then new complications (i.e. new complete curves) emerge. In this way, NUM-CC only registers those complete curves that actually reintroduce new conflictive situations.

The process to calculate the incomplete curves is a little bit different. The goal is to calculate how close the set of incomplete curves are to its ideal value. Remember that too many curves or too few curves produce inadequate results. It is necessary to know the number of incomplete curves (Num-IC) and the Ideal Value of Incomplete Curves (Ideal-IC) to calculate the Proportion of Incomplete Curves (PIC):

$$PIC = \begin{cases} \dfrac{\text{Num-IC}}{\text{Ideal-IC}} & \text{IF NumIC} \leq \text{Ideal-IC} \\ 1 - \dfrac{\text{Num-IC}}{\text{Ideal-IC}} & \text{IF Ideal-IC} < \text{NumIC} \leq \text{Ideal-IC} \cdot 2 \\ 0 & \text{IF NumIC} > \text{Ideal-IC} \cdot 2 \end{cases}$$

Now, it is necessary to analyse each of the curves to see how close they are to its ideal value. One starts getting the amplitude of the first peak and the amplitude of the bottom part of its valley; the ratio between the valley and the peak indicates the percentage, with respect to its peak, that the valley needs to be expanded to reach zero. So, if the peak's amplitude is 10 and the valley's is 4, the valley needs to be expanded 40% to reach zero. The process is repeated for all incomplete curves. The summation of these results is known as the Summation of Incomplete Curves (SIC):

$$SIC = \begin{cases} \sum_{i=1}^{\text{Num-IC}} \dfrac{\text{Amplitude-Valley}_i}{\text{Amplitude-Peak}_i} & \text{If Num-IC} > \text{Ideal-IC} \\ & \text{If Num-IC} \leq \text{Ideal-IC} \\ \left[\sum_{i=1}^{\text{Num-IC}} \dfrac{\text{Amplitude-Valley}_i}{\text{Amplitude-Peak}_i} \right] + (\text{Ideal-IC} - \text{Num-IC}) & \end{cases}$$

Notice that, if the number of incomplete curves is minor to the ideal value of incomplete curves, the difference between them is added to the summation. So, the value of SIC represents how far the set of incomplete curves is from its ideal value. So, if SIC ≈ 0 the new narrative totally satisfies the requirement for reintroducing complications (all curves have deep valleys); if SIC ≈ Ideal-IC the valleys are so small that practically we do not have incomplete curves.

Now, given an Ideal Number of Incomplete Curves (Ideal-IC), it is possible to calculate in what percentage the amplitude of all incomplete curves is similar to its ideal value. We refer to this value as the Total Amplitude of Incomplete Curves (TAI), which is defined as follows:

$$TAI = \begin{cases} \dfrac{\text{Ideal-IC} - \text{SIC}}{\text{Ideal-IC}} & \text{IF SIC} \leq \text{Ideal-IC} \\ 0 & \text{IF SIC} > \text{Ideal-IC} \end{cases}$$

If SIC > Ideal-IC we have too many incomplete curves whose amplitudes do not provide useful information for the evaluation.

Regarding the recountal of a story, we consider that a narrative follows the classical Aristotelian structure when its graph of tension includes at least one complete curve, i.e. the tension at the beginning and at end of the story is zero, and at least once the value of the tensions between these two points is different to zero. So, in this project we analyse if the story under evaluation has an adequate opening and adequate closure in terms of tensions. A story has an adequate opening (A-Opening) when the tension in the story goes from zero at the beginning of the story to some value greater than zero at the first peak.

$$\text{A-Opening} = \frac{\text{Amplitude First Peak - Amplitude (t=1)}}{\text{Amplitude First Peak}}$$

In this way, because our goal is to have a continue tension growing from zero to the first peak, this formula indicates which percentage of this goal is achieved.

One common mistake, particularly between inexperienced writers, is to finish a story leaving loose ends. Thus, following Pérez y Pérez and Sharples, a story "should display an overall integrity and closure, for example with a problem posed in an early part of the text being resolved by the conclusion" (Pérez y Pérez and Sharples 2004). In this way, in order to have an Adequate Closure (A-Closure) all conflicts must be worked out at the end of the story. That is, the value of the tension in the last action must be equal to zero. So, it is necessary to perform a similar process to the one employed to calculate the incomplete curves: one needs to get the amplitude of the curve's main peak, the amplitude of the bottom part of the last valley, and then calculate in what percentage the tension goes down. If the final amplitude of the curve is zero, i.e. if it goes down 100%, the Adequate Closure is set to 1; if the curve goes down 30% the Adequate Closure is set to 0.3; and so on.

$$\text{A-Closure} = 1 - \frac{\text{Amplitude Last Valley}}{\text{Amplitude Main Peak}}$$

Calculation of Interestingness

Thus, our model employs the following characteristics:

- Proportion of new structures (PNS)
- Proportion of the Original Value (POV)
- Novelty of the Knowledge Structures (NKS)
- Proportion of Knowledge Widening (PKW)

- Adequate Opening (A-Opening)
- Adequate Closure (A-Closure)

- Proportion of Complete Curves (PCC)
- Proportion of Incomplete Curves (PIC)
- Total Amplitude of Incomplete Curves (TAI),

The first six characteristics (PNS, POV, NKS, PKW, A-Opening and A-Closure) are known as the core characteristics (CoreC); the last three are known as the complementary characteristics (ComplementaryC). This distinction emerges after talking to some experts in science of human communication that pointed out to us that a story can be interesting even if there are no reintroductions of complications (that is, even if there are no extra complete or incomplete curves). The experts agreed that the reintroduction of problematic situations might add interest to the story, but they are not essential to it. So, we decided that they would complement the evaluation of the core characteristics (a kind of extra points).

It is necessary to set a weight for each of the core characteristics. The sum of all weights must be equal to 1. Thus, the Evaluation of Interestingness (I) is equal to the summation of the value of each core characteristic (CoreC) multiplied by its weight (W):

$$I = \sum_{i=1}^{6} CoreC_i \cdot W_i$$

The Complement (Com) is equal to the summation of the value of each complementary characteristic multiplied by its complementary weigh (w). The sum of all complementary weights ranges from zero to 1.

$$Com = \sum_{i=1}^{3} ComplementaryCi \cdot wi$$

Thus, the total value of interest (TI) is giving by

$$TI = \begin{cases} 1 & \text{if } (I + Com) > 1 \\ I + Com & \text{if } (I + Com) \leq 1 \end{cases}$$

If we combined the values obtained from the correct recountal of a story and the reintroduction of complications, then we can calculate a parameter that we referred to as excitement (E):

$$E = A\text{-}Closure \cdot W + A\text{-}Openning \cdot W + PCC \cdot w + PIC \cdot w + TAI \cdot w$$

Thus, E assigns a value to the increments and decrements of tension during the story.

Implementation of the Prototype

We have implemented a prototype to test our model. Our prototype evaluates the interestingness of four stories generated by our storyteller. Details of our computer model for plot generation can be found in (Pérez y Pérez and Sharples 2001; Pérez y Pérez 2007). In this document we only mention two characteristics that are important to learn in order to understand how the prototype of the evaluator works:

1. Our plot generator employs a set of stories, known as the previous stories, to construct its knowledge base. Such narratives are provided by the user of the system. Any new story generated by the storyteller can be included as part of the previous stories.
2. As part of the process of developing a new story the storyteller keeps a record of the dramatic tension in the story. The following are examples of situations that trigger tensions: when the life of a character is at risk; when the health of a character is at risk; when a character is made a prisoner; and so on. Every tension has assigned a value. So, each time an action is performed the system calculates the value of all active tensions and records it. With this information the storyteller graphs the curve of tension of the story (see figure 3).

Now we explain some details of the implementation of the prototype for the evaluation of interestingness. The model includes several parameters that provide flexibility. The first step is to set those parameters. We start with the expected or ideal values: Minimal Value of New Structures (Min-NS), Minimum value of New Elements (Min-NE), Minimum Original Value (Min-OV), Ideal Value of Complete Curves (Ideal-CC) and Ideal Value of Incomplete Curves (Ideal-IC). To determine the value of these parameters we employ the previous stories as a reference. (The previous stories employed in this work were made long time before this project started. They represent well-formed and interesting narratives. So, they are a good source of information). The process works as follows. We select seven previous stories. With six of them we create the knowledge base; the 7th is considered a new story (as if it had been produced by our storyteller). Then, we analyse how many new structures, new elements, new original value structures, and new complete and incomplete curves are generated by the 7th previous story and record these results. We repeat the same process for each of the previous stories. Then, after eliminating the highest and lowest values, we calculate the means of each result obtained. Following this procedure we conclude that the parameters should be set as follows: Min-NS = 7; Min-NE = 4; Min-OV = 5; Ideal-CC = 1; Ideal-IC = 1. That is, in average each previous story generates seven new knowledge structures, four new elements, five original structures, one complete curve and one incomplete curve.

The next step is to set the weights. Based on empirical experience of experts in human communication, the weight of the generation of new knowledge is set to 50% and the weight of the correctness of the way the narrative is recounted is set to the other 50%.

The characteristics that define the generation of new knowledge are: Proportion of new structures (PNS), Proportion of the Original Value (POV), Novelty of the Knowledge Structures (NKS) and Proportion of Knowledge Widening (PKW). Table 1 shows their assigned weights. We considered Novel knowledge structures more important than Knowledge Widening structures. The correctness of the way the narrative is recounted is defined by the parameters A-Opening and A-Closure. Both are important and both received the same weight. Finally, the LS was set to 85%.

Regarding the complementary parameters and weights, they contribute with a maximum extra value of 10% distributed as follows: 5% for the complete curves and 5% for TAI. This decision is based on our own experience.

Core Characteristic	Weight
Proportion of new structures (PNS)	10
Proportion of the Original Value (POV)	10
Novelty of the Knowledge Structures (NKS)	15
Proportion of Knowledge Widening (PKW)	15
Adequate Opening (A-Opening)	25
Adequate Closure (A-Closure)	25

Complementary Characteristic	Weight
Proportion of Complete Curves (PCC)	5
Proportion of Incomplete Curves (PIC)	0
Total Amplitude of Incomplete Curves (TAI)	5

Table 1. Weights of the characteristics

Finally, if the value of the correct recountal of the story (A-Closure + A-Opening) does not reach at least 50% of its highest possible value, the story is considered as unsatisfactory. In this way we avoid evaluating stories that lack enough quality (the reader must remember that in this paper we do not evaluate coherence; that is a different part of the project. However, this constraint in the prototype helps to avoid processing pointless stories).

Testing the Model

To test our model our storyteller generated four narratives known as short-1, short-2, long-1 and long-2 (see Figure 2). Figure 3 shows their graphics of tension. The following lines describe the main characteristics of each narrative.

Short-1 lacks an introduction; it starts with a violent action. One gets the impression that everything occurs very fast. It is not clear what happens to the virgin once she escapes and has an accident. Also it is unclear the fate of the enemy.

Short-2 has a brief introduction and then the conflict starts to grow (the killing of the knight). The end is tragic and all tensions are sorted out.

Long-1 has a nice long introduction. The conflict between the princess and the lady grows nicely and slowly until it reaches a climax. However, at the end, we do not know the destiny of the characters. Who got the knight? So, the story has an inadequate conclusion.

SHORT 1	SHORT 2
The enemy kidnapped the virgin The virgin laugh at the enemy The enemy attacked the virgin The virgin wounded the enemy The virgin ran away The virgin had an accident ***The End***	Jaguar knight was a citizen The artist prepared to sacrifice the jaguar knight The jaguar knight became free The jaguar knight fought the artist The artist killed the jaguar knight The artist committed suicide ***The End***
LONG 1	**LONG 2**
Jaguar knight was a citizen The princess was a citizen The princess was fond of jaguar knight The princess fell in love with jaguar knight The lady was in love with jaguar knight The princess got jealous of the lady The jaguar knight was in love with the princess The princess attacked the lady The lady wounded the princess The lady ran away The lady had an accident ***The End***	Jaguar knight was a citizen The enemy was a citizen The enemy got intensely jealous of jaguar knight The enemy attacked jaguar knight The jaguar knight fought the enemy The enemy wounded jaguar knight The enemy ran away The enemy went to Texcoco lake The enemy did not cure jaguar knight The farmer prepared to sacrifice the enemy The enemy ran away The jaguar knight died because of its injuries ***The End***

Figure 2. Four computer-generated stories

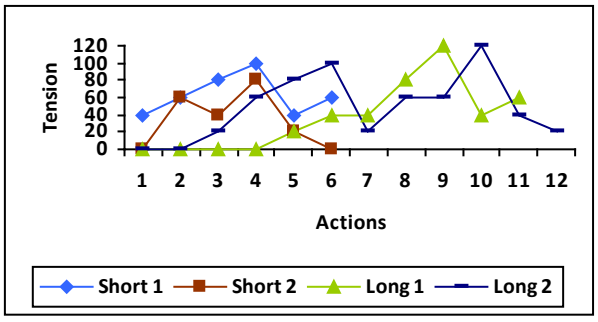


Figure 3. Graphics of Tensions for the four stories

Long-2 starts introducing the characters of the narrative. The tension grows fast until the story reaches a climax when the enemy wounded the knight. The tension decreases when the enemy decides to run off; however, it increases again when the enemy returns and the farmer attempts to kill him. Finally, he escapes again and the knight dies.

Based on our personal taste, our favourite narrative was short-2, then long-2, long-1 and finally short-1. We evaluated these four stories with our prototype. Table 2 shows the results; figure 4 shows the normalised values for the following features: generation of new knowledge, adequate closure, excitement and the total value of interestingness. Against our prediction, the system selected Long-2 as the most interesting story. There were two main reasons that explained why Long-2 beat short-2: 1) Long-2 generated more knowledge structures than Short-2; 2) Long-2's complements were slightly better evaluat-

ed than Short-2's. So, Short-2 obtained the second best result.

	Long-1	Long-2	Short-1	Short-2
PNS	10	8.57	4.29	4.29
POV	0	10	6	6
NKS	0	15	15	15
PKW	3.8	3.75	11.25	3.75
A-Op	25	25	15	25
A-Clo	13	20.83	10	25
I	**51.25**	**83.15**	**Unsatisfactory**	**79.04**
Com	3.4	3.15	3	1.65
TI	**54.6**	**86.30**	**Unsatisfactory**	**80.69**
E	41.4	48.98	28	51.65

Table 2. Numerical values of the evaluation.

In third place was Long-1; it did not produce any original structure and therefore its characteristic NKS got a value of zero. Also, its closure was poor. In last place was Short-1. The system evaluated Short-1 as an unsatisfactory story; i.e., it did not satisfy the minimum requirements of a correct recountal of a story (as we can see in table 2, the opening only got 15 points and the closing 10!). Nevertheless, we included the value of Short-1's closure and excitement in figure 4.

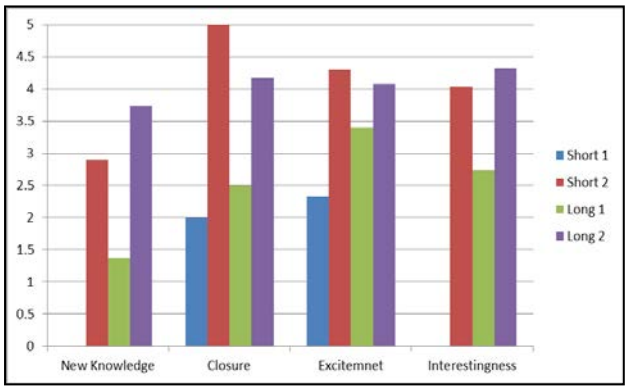


Figure 4. Graphics of the results of the evaluation.

We thought it could be interesting to compare the opinion of a group of subjects about the four stories under analysis to the results generated by our computer evaluator. Thus, we decided to make a survey by applying two questionnaires: 22 subjects answered questionnaire 1 and 22 subjects answered questionnaire 2; 25% were females and 75% were males; 13% had a PhD degree, 29% had a master degree, 27% had a bachelor's degree and 29% had other types of degree. We decided to group the narratives by their length. So, the first questionnaire included the two short narratives while the second questionnaire included the two long narratives. In both questionnaires we asked subjects to evaluate the adequateness of the closure and the interestingness of the stories. Subjects could rank each feature with a value ranging from 1 to 5, where 1 represented the lowest assessment and 5 the highest one. Figure 5 shows the results of the evaluation of interestingness. Short-2 was considered the most interesting narrative; Long-2 seemed to be in the second position followed close behind by Short-1 and Long-1. These last results were not conclusive. We were surprised that Short-1 was not clearly in the last position. We speculated that human capacity of filling gaps when reading a narrative might contribute to this result. Although our computer agent calculated a higher evaluation to Long-2 than to Short-2, both stories got a very similar score (the difference was less than 6%; c.f. with the score of Long-1). So, we felt that subjects' opinion about these two narratives was close to the results we obtained from our computer prototype. However, by contrast, our system clearly rejected Short-1 and left Long-1 in a clear third position while subjects' evaluation was unclear.

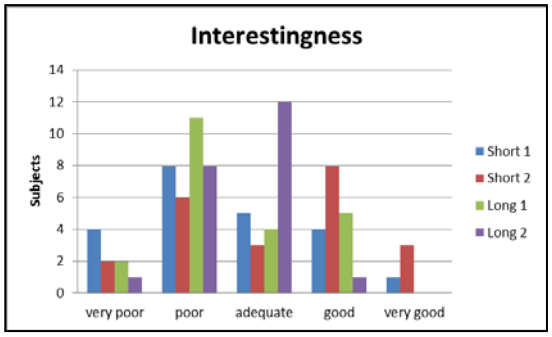


Figure 5. Subjects' evaluation of interestingness.

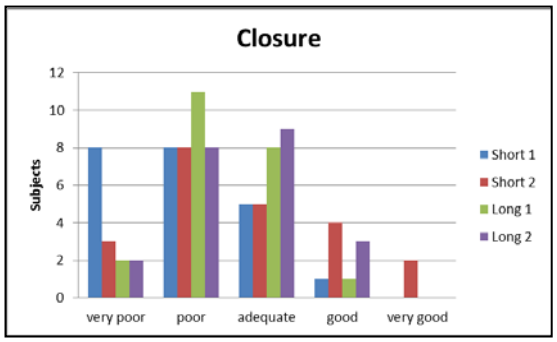


Figure 6. Subjects' evaluation of closure.

Figure 6 shows the results for the evaluation of closure. Subjects ranked Short-2 as the story with the best closure, followed by Long-2, Long-1 and Short-1. There was a total coincidence between the computer agent evaluation and the human evaluation.

Discussion and Conclusions

This paper reports a computer model for the evaluation of interestingness. It is part of a bigger project that attempts to evaluate the interestingness, coherence and novelty of computer generated narratives. The model presented in this paper emphasises two properties: generation of new knowledge and the correctness of the recountal of a story. Regarding the generation of new knowledge, we developed a process to calculate how much new information was produced by a computer generated story. In the same way, motivated by Piaget ideas about accommodation and assimilation, we defined two different types of knowledge structures: new knowledge and widening knowledge. We went further by identifying those new knowledge structures which were very different to the existing ones. Regarding the recountal of a

story, we worked on previous research that had illustrated the relation between the dramatic tension of a story and its interestingness. In this work we expanded this idea by analysing the opening and closure of a story, and verifying if new obstacles were introduced along the plot. Thus, we have been able to create a model that allows a computerised agent to perform a detailed evaluation of the stories it produces.

The implementation of our prototype has allowed testing the ideas behind the model. We are satisfied with the results. But we are more excited about what we are expecting to achieve with this new characteristic. The capacity to evaluate its own outputs allows a storyteller to distinguish positive and negative qualities in a narrative and therefore to learn from its own creative work; it also incorporates the possibility of evaluating and learning from narratives generated by other systems. In our case, we expect that our storyteller agent will be able to determine autonomously which stories, either produced by itself, by other systems or by humans, should become part of its set of previous stories. That is our next goal.

We have compared the results produced by our automatic evaluator to the results obtained from a questionnaire answered by a group of 44 human evaluators. In general terms, the results obtained from both approaches were similar. This suggests that the subjects that answered the questionnaire might consider acceptable the outputs produced by our system. Nevertheless, it is intriguing why the story Short-1 got a relative high evaluation from the subjects. We need to analyse further this result and see if we require adjusting our model.

As it has been showed in this work, we consider the generation of new knowledge an important characteristic of computational creativity. So, it is not enough to evaluate the creative-product and/or the creative-process, as it has been suggested by some researchers. We believe that it is also necessary to considerate how much such products and/or processes modify the characteristics of the storyteller agent and the evaluator agent (that in our case is the same)**.** So, any evaluation process must consider this aspect. This idea is inspired by the fact that, any creative act performed by humans will influence their future creative acts. We need to represent this feature in our computer models.

The qualities that make a story interesting, coherent and novel are complex and many times overlap each other. Our work seems to illustrate part of this overlapping complexity. For example, the generation of new structures might be employed to evaluate novelty; the adequate opening and closure might be employed to evaluate coherence; however, at the same time, they are essential elements to evaluate interestingness. This seems to confirm our idea that a general model of evaluation of narratives at least must contemplate coherence, novelty and interestingness. We are currently working on producing such a general model.

Hopefully this model will be useful not only for those working in plot generators but also to those researchers working in similar areas (e.g. interactive fiction). We are aware that many features not considered in this work might contribute to make a story interesting (e.g. suspense, intrigue). As mentioned earlier, human evaluation is very complex and we do not comprehend yet how it works. Nevertheless, we expect this research contributes to understand better the mechanisms behind it.

References

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. Creative Intelligent Systems: Papers from the AAAI Spring Symposium. 14–20.

Deckers, L. 2005. Motivation Biological, Psychological, and Enviromental. Pearson.

Jordanous, A. 2012. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. Cognitive Computation, 4(3): 246-279

Lodge, D. 1996. The practice of writing: essays, lectures, reviews and a diary. London: Secker & Warbug.

Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating machine creativity. In Weber, R. and von Wangenheim, C. G., eds., *Case-based reasoning: Papers from the workshop programme at ICCBR 01Vancouver*. Canada 129–137.

Peinado, F.; Francisco, V.; Hervás R. and Gervás, P. 2010. Assessing the Novelty of Computer-Generated Narratives Using Empirical Metrics. *Minds and Machines*. 20(4):565-588.

Pereira, F. C.; Mendes, M.; Gervás, P., and Cardoso, A. 2005. Experiments with assessment of creative systems: An application of Ritchie's criteria. In Gervás, P. Veale, T. and Pease, A., eds., *Proceedings of the workshop on computational creativity*, 19th international joint conference on artificial intelligence.

Pérez y Pérez, R. and Sharples, M. 2001 MEXICA: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13(2):119-139.

Pérez y Pérez, R. and Sharples, M. 2004. Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge Based Systems Journal*. 17(1):15-2.

Pérez y Pérez, R. 2007. Employing Emotions to Drive Plot Generation in a Computer-Based Storyteller. Cognitive Systems Research 8(2): 89-109.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Perez y Perez, R., Ortiz, O., Luna, W. A., Negrete, S., Peñaloza, E., Castellanos, V., and Ávila, R. 2011. A System for Evaluating Novelty in Computer Generated Narratives. In *Proceedings of the Second International Conference on Computational Creativity*, México City, México, pp. 63-68

Piaget, J. 1952. The Origins of Intelligence in Children. London: Routledge and Kegan Paul, 1936 (French version published in 1936, translation by Margaret Cook published 1952).

Sharma, M., Ontañón, S., Mehta, M. and Ram, A. 2010. Drama Management and Player Modeling for Interactive Fiction Games. *Computational Intelligence Journal*, 26(2): 183-211.

Weyhrauch, P. 1997. Guiding Interactive Drama. PhD Dissertation, School of Computer Science, Carnegie Mellon University.

A Model of Heteroassociative Memory: Deciphering Surprising Features and Locations

Shashank Bhatia and **Stephan K. Chalup**
School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan, NSW 2308 Australia
shashank.bhatia@uon.edu.au, stephan.chalup@newcastle.edu.au

Abstract

The identification of surprising or interesting locations in an environment is an important problem in the fields of robotics (localisation, mapping and exploration), architecture (wayfinding, design), navigation (landmark identification) and computational creativity. Despite this familiarity, existing studies are known to rely either on human studies (in architecture and navigation) or complex feature intensive methods (in robotics) to evaluate surprise. In this paper, we propose a novel heteroassociative memory architecture that remembers input patterns along with features associated with them. The model mimics human memory by comparing and associating new patterns with existing patterns and features, and provides an account of surprise experienced. The application of the proposed memory architecture is demonstrated by identifying monotonous and surprising locations present in a Google Sketchup model of an environment. An inter-disciplinary approach combining the proposed memory model and isovists (from architecture) is used to perceive and remember the structure of different locations of the model environment. The experimental results reported describe the behaviour of the proposed surprise identification technique, and illustrate the universal applicability of the method. Finally, we also describe how the memory model can be modified to mimic forgetfulness.

Introduction

Within the context of evaluating computational creativity, measures of accounting surprise and identifying salient patterns have received great interest in the recent past. Known by different names, the problem of accounting surprise has been applied in various research areas. Specifically, the problem of identifying locations that stimulate surprise has important applications in areas such as robotics, architecture, data mining and navigation. Robotics researchers, while aiming towards robot autonomy, intend to identify locations that can potentially serve as landmarks for the localisation of a mobile robot (Cole and Harrison 2005; Siagian and Itti 2009). Architects, on the other hand, intend to design building plans that comprise sufficient salient/surprising locations in order to support way-finding by humans (Carlson et al. 2010). Lastly, navigation experts mine existing maps to identify regions/locations that can serve to better communicate a route to the users (Xia et al. 2008; Perttula, Carter, and Denoue 2009). Common to all these applications is the underlying question, the problem of identifying patterns from raw data that appeal or stimulate human attention. While the aim of these applications is same, the underlying measure of accounting surprise that each one follows has been designed to suit only the respective application. There are no domain-independent methods available that are flexible enough to be adaptable universally. Itti (2009) and Baldi (2010) rely on Bayesian statistics, and their method would require considerable domain-specific alteration, as can be seen in (Ranganathan and Dellaert 2009; Zhang, Tong, and Cottrell 2009). On one hand, designing methods that are domain-independent having capacity of comparing multi-dimensional data is a challenging task. On other hand, the use of dimensionality reduction techniques to limit or reduce dimensionality are known to cause bias. The reduction of dimensions would depend on methods employed, and different methods may assign varying weights to each dimension (Brown 2012). This makes surprise measurement, which includes comparing multi-dimensional patterns, a challenging problem.

Commonly known as outlier detection, novelty detection, saliency detection etc., the question of detecting a "surprising event" has been raised in the past (Baldi and Ittii 2010). Specifically, the methods that provide a domain-independent approach for discovering inherent surprise in perceived patterns aim for information maximisation. In an information-theoretic sense, patterns that are rare are known to contain maximum information (Zhang, Tong, and Cottrell 2009). In a more formal sense, patterns that lead to an increase in entropy are deemed as unique, and are known to cause surprise (Shannon 2001). Another argument in the literature is about the frequency of occurrence of such patterns. An event/pattern that has a lower probability of occurrence/appearance, is deemed rare. Therefore, various proposals have been made that compare probabilities (Bartlett 1952; Weaver 1966) and identify the pattern with the lowest probability value. These techniques were further refined to consider the probabilities of all other patterns as well (Weaver 1966; Good 1956; Redheffer 1951). Most recent developments use Bayesian statistics to compare the probabilities of the occurrence of patterns or features extracted from them. Baldi and Ittii (2010) proposed to employ a dis-

tance metric to measure the differences between prior and posterior beliefs of a computational observer, and argued its interpretation to be that of an account of surprise. The authors proposed the use of Kullback-Leibler divergence (Kullback 1997) as the distance metric, and discussed its advantages over Shannon's entropy (Shannon 2001). They demonstrated the use of their proposed method by identifying surprising pixels from an input image. The complex mathematical constructs of modelling surprise that exist in the literature are difficult to adapt, and therefore have not found their applications across different domains.

The concept of surprise can also be understood through its relationship to memory. Something that has not been observed stimulates surprise. In this setting, if a computational agent remembers the percepts presented to it, a measure of surprise can be derived. Baldi and Ittii (2010) follow this idea, but their perceptual memory is in the form of a probabilistic model. The patterns that are already observed compose the prior model, and the model obtained after adding new percepts is the posterior. As noted previously, most often the patterns/features to be evaluated are available in the form of a vector quantity (Brown 2012). Conversion of this multi-dimensional quantity into a probabilistic model not only requires specific expertise, but is also sensitive to the method employed to update the model's parameters. Even after substantial effort in design, the memory is sensitive to the parameters employed for the model. These shortcomings of the state-of-the-art methods form one part of motivation behind the current paper.

Another aspect that is ignored in most contemporary methods is the associative nature of memory. Human memory has a natural tendency to relate/associate newly perceived objects/patterns with those perceived in the past. Recent research in cognitive science supports the influence perceptual inference has on previous memory (Albright 2012). A classical example is the problem of handwritten digit recognition. Multiple handwriting patterns corresponding to the same digit are labelled and associated via the same label. Since the memory is always trying to associate new patterns with previous experience, it is obvious that a strong association will lead to lower surprise and vice versa. This property of association, though well-recognised, has not been incorporated in the state-of-the-art methods of measuring surprise. This forms the second motivation of the current paper.

Inspired by the discussed shortcomings of existing methods, this paper presents a computational memory framework that can memorise multi-dimensional patterns (or features derived from them) and account for inherent surprise after attempting to associate and recall a new pattern with those already stored in the memory. The uniqueness of the memory model is two-fold. Firstly, it can be employed without converting the perceived patterns into complex probabilistic models. Secondly, for the purpose of accounting surprise, the memory model not only aims to match and recall the new pattern, but also attempts to associate its characteristics/features before deeming it surprising. To illustrate these advantages and their usage, the proposed method is employed to identify monotonous and surprising structural features/locations present in an environment. Noted previously, this is an important problem in the field of robotics as well as architecture, and therefore we use a Google Sketchup (Trimble 2013) based architectural model for the demonstration. An isovist - a way of representing visible space from a particular location (Benedikt 1979) - is used for the purpose of perceiving a location in the form of a multi-dimensional pattern. This paper points towards the methods of extracting isovists from respective environments (section: Spatial Perception), and provides details of the neural network based memory architecture (section: Associative memory). Experimental results compare the degree to which identified monotonous locations associate with each other, and illustrate the isovist shape of those that stimulate computational surprise (section: Experiments & Results). Additionally, we describe how the proposed memory model can be modified to mimic forgetfulness, thereby forgetting patterns that have not been seen in a given length of time. To conclude, the paper provides a discussion on prospective applications of the proposed framework, and demonstrates its universality by evaluating its performance in a classification task, on various pattern classification datasets (section: Conclusions & Discussoins).

Spatial Perception

This work utilises multi-dimensional Isovist patterns to perceive/represent a location. Conceptually, an isovist is a geometric representation of the space visible from a point in an environment. If a human were to stand at a point and take a complete 360° rotation, all that was visible forms an isovist. In practice, however, this 3D visible space is sliced horizontally to obtain a vector that describes the surrounding structure from the point of observation, also known as the vantage point. This 2D slice is essentially a vector composed of lengths of rays projected from the vantage point, incident on the structure surrounding the point. Therefore, if a 1° resolution was utilised, an isovist would be a 360-dimensional vector, $\vec{I} = [r_1, r_2, \ldots, r_{360}]$ where r_θ represents the length of the ray starting from the vantage point, and incident on the first object intersected in the direction θ. This way, an isovist records a profile of the surrounding structure (illustrated in figure 1). In an environment, multiple isovist can be generated from different vantage points. Each isovist can be represented as a 360-dimensional pattern describing the structure visible from the vantage point. In this paper, an indexed collection of isovist patterns extracted from an existing model of the environment is used.

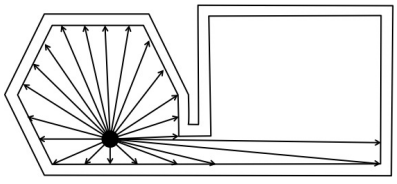


Figure 1: A hypothetical 2D plan of an environment, showing a vantage point (black dot) and the corresponding isovist generated from the vantage point.

Isovist Extraction

The method of extraction of isovists employed in this paper is derived from our previous work (Bhatia, Chalup, and Ostwald 2012), where we employed a Ruby script that executes on the Google Sketchup platform and extracts 3D isovists from a Google Sketchup model. This records the isovists while using the "walk through" tool provided in Google Sketchup. The "walk through" tool allows a user to walk through a 3D model of an architectural building plan. However, in this work, we utilise modified version of the Ruby script that extracts a 2D slice of the perceived 3D isovist. The model of a famous architectural building, Villa Savoye, is used to extract the isovist and identify the surprising locations present. The building is known for uniqueness of its structure, and therefore provides good examples for the evaluation of surprising locations.

Inputs and association patterns

An isovist records a spatial profile, and can be used to memorise a location by a computational memory. This is an advantage while trying to recognise/identify a location by its isovist; however, becomes a drawback when the aim is to infer surprise through association. A simple example is the case of two rectangular rooms that are similar in shape, but have different side lengths. While the isovists recorded at the central point of these rooms would have a large difference, the number of straight edges, and the angles they make, remain the same (90°). Therefore, for the purpose of associating and finding similarities between two locations, in this paper we employed a 3-dimensional feature vector derived from isovist pattern. We compute *(i)* Area of the isovist, *(ii)* Eccentricity value, and *(iii)* Circularity value to form the elements of the 3-dimensional associated feature pattern. This feature pattern is used to associate two isovist patterns. The perceived isovist pattern, therefore, comprises a 360-dimensional vector, and the derived associated pattern is a 3-dimensional feature vector. The isovist of a location and the feature vector are presented as a pair to the memory model proposed in this paper. The memory model remembers essential patterns and computes surprise after associating new patterns and comparing existing ones. Due to the association task that the memory performs, such memories are known as Associative Memories (Palm 2013).

Associative Memory

Associative memories are computational techniques, capable of storing a limited set of input and associated pattern pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$. Depending on the size of the input vector x_i, its associated pattern y_i, and methods of association, various types of such memories are proposed. Kosko (1988) was the first to introduce Bidirectional Associative Memories (BAM), which provides a two way association search mechanism termed Heteroassociation. A BAM can take either the input or associated pattern as its input and has the capacity to recall the respective association. Despite the utility BAM can offer, its usage has been limited due to many existing challenges, such as limited capacity and conditions of instability. Importantly, existing variations of BAM can only memorise binary patterns. Many other variations of BAM have been offered, however, and the present note is provided only as a basis for the following discussion and is by no means an exhaustive account of the developments on this topic. A detailed review can instead be found in (Palm 2013). The proposed memory model offers similar functionality without requirement for input patterns to be binary in nature.

Overview of the architecture

The architecture of the proposed memory model consists of two memory blocks, and can be divided into three parts. *(a)* **Input Memory Block (IMB):** block that stores input patterns, *(b)* **Associated Memory Block (AMB):** block that stores associated feature vectors/patterns, *(c)* **Association Weights:** a matrix that maintains a mapping between the two memory blocks. Complete architecture of the memory is represented in figure 2.

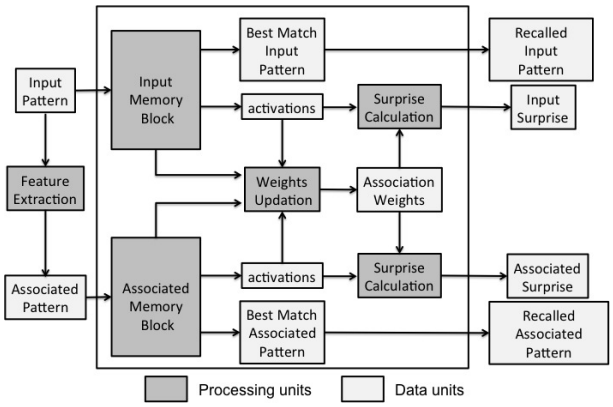


Figure 2: Memory Architecture: Comprise two memory blocks and association weights, all linked through one or more data/processing units presented in white and grey colour respectively.

The memory blocks are the storage units responsible for memorising input and associated patterns. This memory model in concept works similar to traditional BAMs except that it provides additional many-to-many mapping functionality on real-valued vectors. Input patterns (which in the case of this application are isovist vectors) when presented to the memory model are compared in two respects: *(a)* similarity of shape, and *(b)* similarity of the features derived from them. The detailed construction and working of each block and the overall memory model is provided in the following subsections.

Memory Blocks

The smallest unit of storage in this memory model is a Radial Basis activation unit, also known as a Radial Basis Function (RBF). Typically, a RBF is a real valued function with its response monotonically decreasing/increasing with distance from a central point. The parameters that describe a RBF include the central point c, distance metric $\|\cdot\|_d$ and

the shape of the radial function. A Gaussian RBF with Euclidean distance metric and centre c_i is defined in equation 1. The parameters c_i and radius σ_i decide the activation level of the RBF unit. Any input x lying inside the circle centred at c_i having a radius less than or equal to σ_i will result in a positive activation, with the level of activation decreasing monotonically as the distance between the input and the centre increases.

$$\phi_i(x) \quad = \quad \exp\left(-\frac{(x-c_i)^2}{\sigma_i^2}\right) \qquad (1)$$

The realisation of a memory element in our approach is done by saving the input as the centre c_i, and adjusting the value of the radius σ_i to incorporate values that lie close to each other. Mathematically, this memory element will have $\phi_i(x) > 0$ activation for all values of x that fall in a σ_i neighbourhood of the point c_i defined in equation 2. Further, $\lim_{x \to c_i} \phi_i(x) = 1$. This condition ensures that the activation unit with the centre c_i closest to the current input x activates the most.

$$B(c_i; \sigma_i) \quad = \quad \{x \in X \mid d(x, c_i) < \sigma_i\} \qquad (2)$$

In a collection of multiple RBF units, with each having a different centre c_i and radius σ_i, multiple values can be remembered. If an input x is presented to this collection, the unit with highest activation will be the one that has the best matching centre c_i. Or in other words, for the presented input value, the memory block can be said to recall the nearest possible value c_i. For one input pattern, there will be one corresponding recall value. This setting of multiple RBF units can thus work as a memory unit. The Memory Blocks described previously comprise multiple RBF units. As an example, a memory block comprising n RBF units can be represented with figure 3.

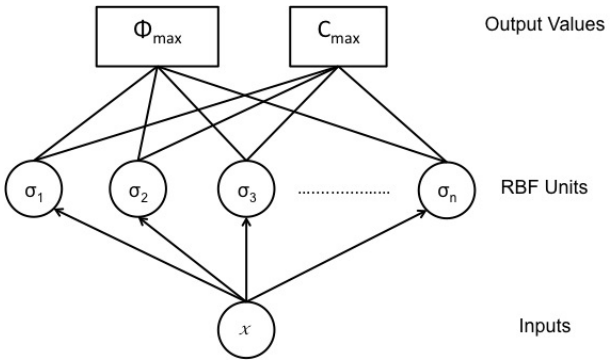


Figure 3: RBF Memory Block: Each RBF unit stores one data value in the form of centre c_i; the range of values for which the unit has positive activation are defined by the values of σ_i according to equation 2. c_{max} is the value that the memory recalls as the best match to the input, and ϕ_{max} represents the confidence in the match.

So far we have described the use of the RBF unit as a memory block having a scalar valued centre c_i. In order to memorise a multi-dimensional pattern (in this application an isovist pattern, comprising 360 ray-lenghts), we modify the traditional RBFs to handle a multi-dimensional input isovist vector $\vec{x}$ by replacing its scalar valued centre with a 360-dimensional vector $\vec{c_i}$. While Euclidean distance and dot product of two multi-dimensional vectors are also scalar and do not disrupt the working of standard RBFs, their capacity to capture the difference in shape between two isovist patterns is minimal. Therefore, in order to account for difference in shape, we replace the Euclidean distance metric by Procrustes Distance (Kendall 1989). The procrustes distance is a statistical measure of shape similarity that accounts for dissimilarity between two shapes while ignoring factors of scaling and transformation. For two isovist vectors $\vec{x}_m$ and $\vec{x}_n$, the procrustes distance $\langle \vec{x}_m, \vec{x}_n \rangle_p$ first identifies the optimum translation, rotation, reflection and scaling required to align the two shapes, and finally provides a minimised scaled value of the dissimilarity between them. An example of two similar and non-similar isovists with their procrustes-aligned isovists is shown in figure 4. Utilising procrustes distance with the multidimensional centre $\vec{c_i}$, we term this Multidimensional Procrustes RBF, which is defined as:

$$\phi_i(\vec{x}) \quad = \quad \exp\left(-\frac{\langle \vec{x}, \vec{c_i} \rangle_p^2}{\sigma_i^2}\right) \qquad (3)$$

Procrustes distance provides a dissimilarity measure ranging between 0 and 1. A zero procrustes distance therefore leads to maximum activation and vice versa. A multidimensional procrustes RBF has the capacity to store a multi-dimensional vector in the form of its centre. It is important to note that for the application described in this paper, the difference between two multi-dimensional vectors, viz. the isovists, was recorded using procrustes distance. However, in general the memory model can be adapted for any suitable distance metric, or used with the simple Euclidean distance. The use of procrustes distance as a distance metric was adapted specifically for the purpose of the application of identifying surprising locations in an environment.

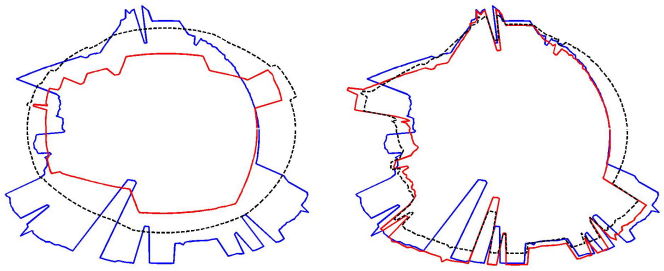


Figure 4: Two isovist pairs (illustrated in red and blue) and corresponding aligned isovists (black dashed), one with a high procrustes distance (left) and other with a low procrustes distance (right).

IMB and AMB IMB and AMB are in principle collections of one or more multidimensional-procrustes RBF and multidimensional RBF units respectively, grouped together as a block (such as the one represented in figure 3). Each

block is initialised with a single unit that stores the first input vector (for IMB) and derived features (for AMB). The feature vector employed to associate two input patterns (in this application isovists) comprise *(i)* area, *(ii)* circularity, *(iii)* eccentricity, together making up a 3-dimensional vector. Initially, each block is created with a single memory unit having a default radius 0.1. Thereafter, the memory block adapts one of the two behaviours. For new patterns that lie far from the centre, the memory block grows by incorporating a new RBF unit having its centre same as the presented pattern. On the other hand, for patterns that lie close to existing patterns, the radii of the RBF units are adjusted in order to obtain positive activation. Adjustment of the radii is analogous to adjustments of weights performed during the training of a Neural Network. The procedure followed to expand or adjust the radii can be understood by following algorithms 1 & 2. Consider a memory block comprising k neural units, with their centres $\vec{c_1}, \vec{c_2}, \ldots, \vec{c_k}$ and radii $\sigma_1, \sigma_2, \ldots, \sigma_k$ and the distance metric $\langle\cdot\rangle_d$. Let the model be presented with a new input vector $\vec{x}$. The algorithm 1 first computes $\langle\cdot\rangle_d$ distance (procrustes distance in the case of an isovist block) between each central vector and the presented pattern, and compares the distance with prespecified best and average match threshold values Θ_{best} and $\Theta_{average}$. If the distance value is found as $d \leq \Theta_{best}$, the corresponding central vector is returned - as this signifies that a similar pattern already exists in memory. However, in the case where $\Theta_{avg} \leq d < \Theta_{best}$, the radius of the corresponding best match unit is updated. This updating ensures that the memory responds with a positive activation when next presented with a similar pattern.

Algorithm 1 Memory Block Updation

Require: $\vec{x}$, $[c_1, c_2, \ldots, c_k]$, Θ_{best}, Θ_{avg} , Σ
for all center vectors c_i **do**
  $d_i(\vec{x}) \Leftarrow \langle \vec{x}, \vec{c_i} \rangle_d$
end for
$bestScore \Leftarrow \min_i (d_i)$
$bestIndex \Leftarrow argmin\,(d_i)$
$blockUpdated \Leftarrow$ false
if $(\Theta_{best} \leq bestScore)$ **then**
  $\vec{r} \Leftarrow \vec{c}_{bestIndex}$
  $blockUpdated \Leftarrow$ true
else if $(\Theta_{avg} \leq bestScore < \Theta_{best})$ **then**
  if $(\sigma_{bestIndex} < \Sigma)$ **then**
    $[\vec{c}_{bestIndex}, \sigma_{bestIndex}] \Leftarrow$ computeCenter()
    $blockUpdated \Leftarrow$ true
  end if
end if
if $(blockUpdated ==$ false$)$ **then**
  add new neural unit center with
  $\vec{c}_{k+1} = \vec{x}$
  $\sigma_{k+1} = 0.1$
end if

The network expands on the presentation of patterns that cannot be incorporated by adjusting the weights/radii of the RBF units. This feature provides three advantages over the traditional BAMs. The first is that there is no a-priori training required by the memory block. The memory is updated as new patterns are presented, and the training is online. Secondly, adjustment of weights ensures that similar patterns are remembered through a common central vector, thereby reducing the number of neural units required to remember multiple patterns. Despite the averaging process, a high level of recall accuracy is guaranteed by maintaining all radii $\sigma_i \leq \Sigma$. The values of Θ_{best}, Θ_{avg} and Σ are application specific parameters that require adjustment. However, for the purpose of associating and remembering isovists, in our application we determined these using equations $4, 5, 6$. Here, D_{ij} is a $n \times n$ matrix containing $\langle\cdot\rangle_p$ distances between all central vectors; $std(D_{ij})$ stands for standard deviation.

Algorithm 2 Center vector and radius calculation

Require: $\vec{c}_{bestIndex}, \Theta_{best}, \vec{x}$
$\vec{c}_{old} \Leftarrow \vec{c}_{bestIndex}$
$\vec{c}_{bestIndex} \Leftarrow (\vec{c}_{bestIndex} + \vec{x})\,/2$
$d_{new} \Leftarrow \frac{\left(\langle \vec{x}, c_{best\vec{I}ndex} \rangle_d\right)^2}{-2 \cdot \log(\Theta_{best})}$
$d_{old} \Leftarrow \frac{\left(\langle \vec{c}_{old}, c_{best\vec{I}ndex} \rangle_d\right)^2}{-2 \cdot \log(\Theta_{best})}$
$\sigma_{bestIndex} \Leftarrow \max(d_{new}, d_{old})$

$$
\begin{aligned}
D_{ij} &= \langle \vec{c_i}, \vec{c_j} \rangle_p \\
S_d &= \sum_{i \neq j}^{n} D_{ij} \\
\Theta_{best} &= \frac{percentile(S_d, 95)}{S_d} \qquad (4) \\
\Theta_{avg} &= \frac{percentile(S_d, 50)}{S_d} \qquad (5) \\
\Sigma &= \frac{\min(std(D_{ij}))}{\max(std(D_{ij}))} \qquad (6)
\end{aligned}
$$

Association Weights Association weights act as a separate layer of the network architecture, and play the role of mapping the input patterns with their associated features. For a case of m isovist patterns and n associated feature vectors stored in IMB and AMB respectively, the association weights would comprise a $(m \times (n+1))$ matrix. The first column of the matrix contains the indices of each central vector $\vec{c_i}$ and the remaining columns contain mapping weights. On initialisation, the mapping weights are set to zero. Once each memory block is updated, the corresponding best match index obtained as an output of the memory block is used to configure the values of the matrix. Let q be the index returned from IMB, and r be the index obtained from AMB. The weight updation simply increments the value at the q^{th} row and $r + 1^{th}$ column of the weight matrix. If such a row or column does not exist (signifying a new addition to the memory block), a new row/column is added. During the use of the memory model to recall the associated vector from the presented input vector, assuming an index p was returned, the p^{th} row is selected, and the index of the column containing the highest score is obtained.

Let this index be k. If the highest score in k^{th} column this implies that for AMB, the centre of the k^{th} activation unit is most strongly associated with the current input. This kind of mapping look-up can be performed vice versa as well and provides an efficient bi-directional many-to-many mapping functionality, which is hard to implement in traditional memory models.

Surprise Calculation The Kullback-Leibler (KL) divergence (Kullback 1997) is a measure of difference between two probabilistic models of current observations. To estimate KL divergence, an application specific probabilistic model of the current data is required, and in most cases the design of such a model requires specific expertise. In our approach, each memory model computes the surprise without having the need to train/estimate or design any probabilistic model. This is achieved by using activation scores that each memory unit outputs on presentation of a pattern. These scores are obtained through RBF activation units. Each score in principle is therefore a probabilistic estimate of the similarity between the input vector and the centre of the corresponding memory unit. Exploiting this property, we measure the KL divergence on activation scores. On presentation of a new input vector $\vec{x}$ to a memory block, the activation scores are first computed. Since these scores are calculated before the block updates (using algorithm 1 & 2), they are termed a-priors, $A = [a_1, a_2, \ldots, a_n]$. Post the execution of algorithm 1, the memory block would either remain the same (in the case of best match), or change one of its radius values (for average match), or lastly may have an additional neural unit (no match). Accordingly, the activation scores obtained after the updating might be different from the a-priors. Scores obtained after the updating of memory are termed posteriors, $P = [p_1, p_2, \ldots, p_m]$. If $n < m$, the a-priors are extrapolated with the mean value of A to ensure $m = n$, and finally the KL-divergence or the surprise encountered is computed as:

$$S = \sum_{i=1}^{m} \ln\left(\frac{p_i}{a_i}\right) \cdot p_i \tag{7}$$

Here a_i and p_i are a-prior and posterior activation scores respectively. IMB and AMB each provides an estimate of the surprise encountered by each block. Surprise value from IMB indicates the surprise in terms of shape of the isovist (in the current application), and one from AMB indicate the surprise encountered in terms of associated features. Overall surprise is an average of the two surprise values. Illustration of the surprise values returned from AMB along with the values in the input vector are presented in figure 5a. Calculation of surprise in the memory model has two advantages, one that the user does not need to meticulously design of a probabilistic model and second that the surprise calculation is independent of the number of dimensions of the input vector.

Forgetfulness in memory

In order to imitate human memory more closely, one additional functionality that can be added in the presented memory model is the property of forgetting. The principal of "out of sight is out of mind" can be implemented in the presented memory model by the use of a bias value for each memory unit. Diverting from the traditional use of bias values, in our approach a bias value is used to adjust the activation score in such a way that the most recently perceived or activated memory unit attains a tendency to have higher activation score, and vice versa. This is achieved by decrementing the bias values of the units that were not recalled. In this way, if a pattern is presented once to the memory and is never recalled, that pattern will have the lowest bias. The effect of low bias will be low levels of activation, and therefore a low recall rate. This feature is an important consideration when evaluating "what causes surprise" and is therefore programmed as an optional configuration that can be used in the current memory model. However, for the current evaluation of surprising locations, it is assumed that the perceiver will not forget any location that was presented earlier.

Experiments & Results

Deciphering surprising structures

The isovist patterns extracted from the Google Sketchup models along with the feature vector (described earlier) were presented one at a time to the memory model. For the present application, the values of Θ_{best} and Θ_{avg} were appropriately selected to ensure that the change in size of the location, viz. the value of area, does not contribute to the value of AMB surprise. This was deliberately designed to serve the purpose of the present application, viz. deciphering surprising locations. The aim in our application was to consider a location surprising largely based on the surprise caused by its shape (isovist) and, to a limited extent, by the associated features. Hence only regions that differ in shape as well as in the values of derived features tend to be most surprising. The plot in figure 5(a) illustrates the values of surprise (ordinate) obtained from IMB and AMB for each isovist index (abscissa). As evident, the values of IMB surprise are initially very high, since the memory model has not been exposed to any isovist patterns. As the memory is presented with more isovist patterns (represented by increasing index of isovists), the surprise initially fluctuates, and then gradually decreases. On the other hand, AMB surprise always retains low values due to the low value of match thresholds Θ_{best} and Θ_{avg} chosen for AMB. However, despite low match thresholds, the AMB surprise was highest at two locations where the associated feature values peaked (illustrated in figure 5(b)). Again, this sudden drift was surprising and was very well captured by the computed surprise shown in the same plot. The view of the location corresponding to locations with highest and lowest surprise values are presented in figure 5(c). The views are recorded from the Google Sketchup model.

Forgfullness demonstration

The behaviour of IMB and AMB surprise - while having the forgetting behaviour enabled - can be very well verified from figure 6(a) and (b). Figure 6(a) presented the IMB and AMB surprise values obtained with the same experiment comprising 300 isovists. Unlike figure 5(a), this time the gradual de-

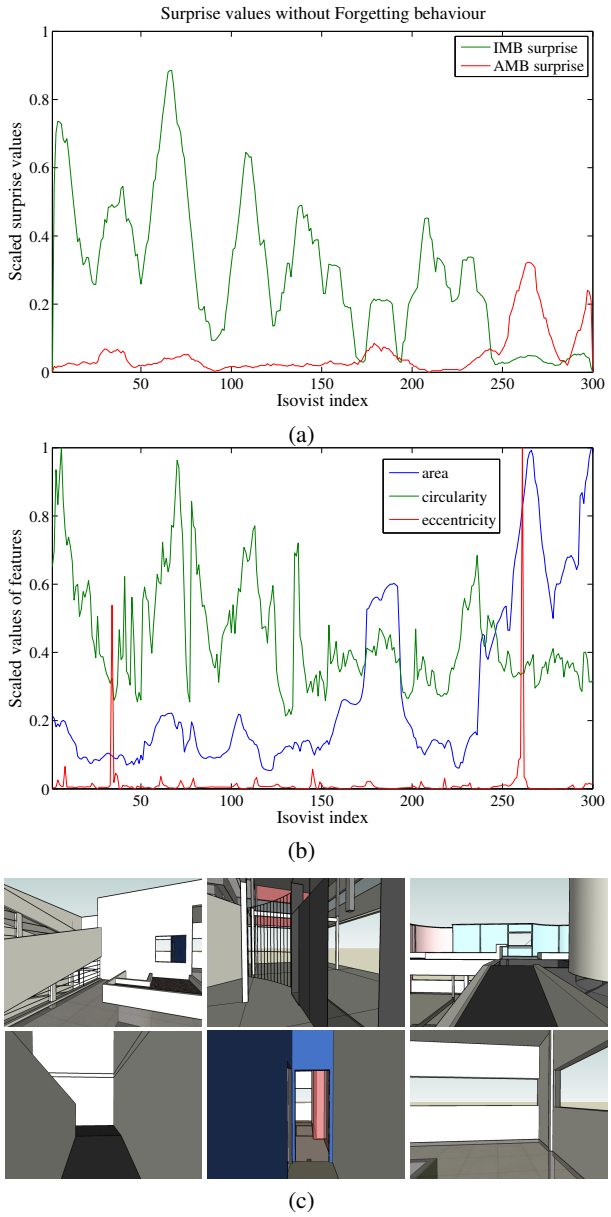


Figure 5: The figure illustrates the results of surprise evaluation of IMB and AMB, without forgetfulness behaviour. 5(a) presents scaled values of IMB and AMB surprise, and 5(b) presents scaled values of associated features. 5(c) illustrates the view from identified high surprise (top row), and low surprise (bottom row) locations. It was discovered that surprise values were high at transitions between two locations, and low surprise was identified at locations with monotonous passages and rooms.

crease in the values of IMB surprise is not noticed. Regular peaks demonstrate that despite prior exposure to similar isovists or features, both IMB and AMB evaluate high surprise. This is because each memory block is implementing the forgetting behaviour (described earlier). As a result, they forget what was previously remembered, and hence cause higher values of surprise. The general trend in the difference of surprise values with forgetting and without forgetting behaviour is illustrated in figure 6(b). The white region between the two curves is the difference between overall surprise values. Remembering all patterns without forgetting causes the surprise values to gradually reduce. In comparison to the values of surprise with forgetting behaviour, these cause fewer peaks. Additionally, the thick red and green curves present smoothened values of overall surprise with forgetting and without forgetting behaviour respectively. These again provide the reader with the general trend each one follows.

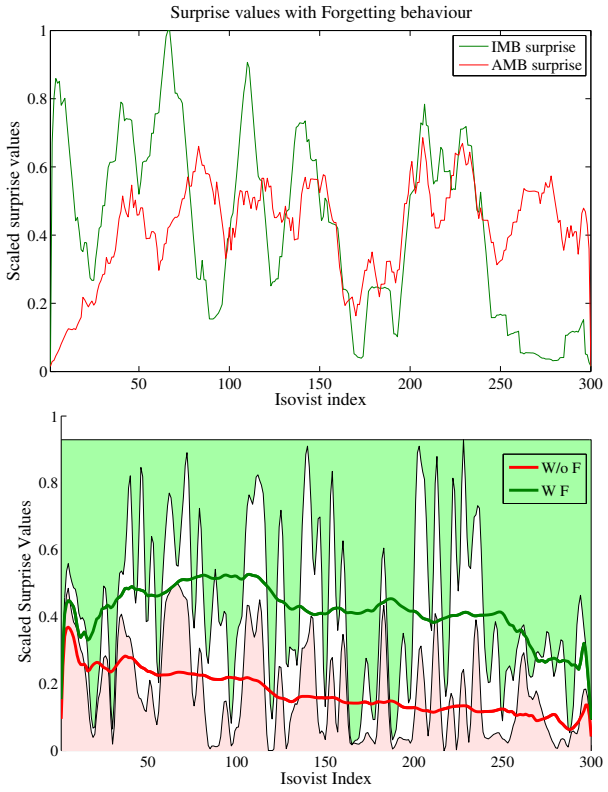


Figure 6: Comparison of the results of surprise evaluation with forgetfulness either enabled or disabled. 6(a) presents individual IMB and AMB surprise values, and 6(b) presents the difference between overall surprise experienced in the two cases. This is shown by the two shaded regions. Additionally, 6(b) also represents smoothened values of overall surprise in case of forgetting enabled (WF) and disabled (W/o F). Surprise values of IMB and AMB were found to attain more frequent peaks in the memory with forgetfulness, as it tends to "forget" previously presented patterns.

Conclusions & Discussion

In this paper, we presented a computational model of associative memory that is capable of remembering multi-dimensional real-valued patterns, performing bi-directional

association, and importantly, mimicking human memory by providing an account of surprise stimulated. The memory model is constructed using collections of multi-dimensional RBF units with procrustes distance as the metric for comparison between input and centre. The unique feature of the presented memory model is that it masks the complex requirement of probabilistic modelling required otherwise in the current literature for computing surprise. Additionally, the presented memory model, while providing similar functionality to BAM has capacity to remember real-valued patterns without issues concerning stability. Furthermore, similar to the working of human memory, the presented memory model can be configured to forget patterns that are not recalled over long periods of time, thereby implementing the rule, "out of sight is out of mind".

The use of the memory model is demonstrated by identifying locations within an architectural building model that has variations in structure, which stimulates surprise. An isovist - a way of representing the structural features of a location - is used to represent the shape of a surrounding environment. Experimental results reveal and confirm the expected behaviour of surprise computation in two ways. First, from the application point of view, the identified high surprise locations were found to exist near transitions between two smaller parts of the "Villa-Savoye" house. This would be expected when the shape of the region where a person/agent enters changes its shape drastically. Second, the expected difference between the surprise values obtained from two experiments with forgetfulness behaviour enabled and disabled was verified (figure 6(b)). While the values of overall surprise continued to spike in the memory with forgetfulness, a gradual decrease was observed in the memory without forgetfulness. These two results verify the behaviour of surprise computation and the forgetfulness behaviour of the proposed memory model, and the technique employed for surprise computation.

Acknowledgments

The 3D model used for the test was obtained from the Google Sketchup Warehouse (Villa Savoye by Keka 3D Warehouse 2007). The authors are grateful to R. Linich for language review.

References

Albright, T. D. 2012. On the Perception of Probable Things: Neural Substrates of Associative Memory, Imagery, and Perception. *Neuron* 74(2):227–245.

Baldi, P., and Ittii, L. 2010. Of bits and wows: A Bayesian theory of surprise with applications to attention. *Neural Networks* 23(5):649–666.

Bartlett, M. S. 1952. The statistical significance of odd bits of information. *Biometrika* 39:228–237.

Benedikt, M. 1979. To take hold of space: isovists and isovist fields. *Environment and Planning B: Planning and Design* 6(1):47–65.

Bhatia, S.; Chalup, S. K.; and Ostwald, M. J. 2012. Analyzing Architectural Space: Identifying Salient Regions by Computing 3D Isovists. *In Proceedings of 46th Annual Conference of the Architectural Science Association, ASA 2012*.

Brown, D. C. 2012. Creativity, Surprise & Design: An Introduction and Investigation. *The 2nd International Conference on Design Creativity (ICDC2012)* 1:75–86.

Carlson, L. A.; Hölscher, C.; Shipley, T. F.; and Conroy-Dalton, R. 2010. Getting Lost in Buildings. *Current Directions in Psychological Science* 19(5):284–289.

Cole, D., and Harrison, A. 2005. Using Naturally Salient Regions for SLAM with 3D Laser Data. In *In Proceedings of International Conference on Robotics and Automation, Workshop on SLAM*.

Good, I. J. 1956. The surprise index for the multivariate normal distribution. *The Annals of Mathematical Statistics* 27(4):1130–1135.

Itti, L., and Baldi, P. 2009. Bayesian surprise attracts human attention. *Vision Research* 49(10):1295–1306.

Kendall, D. G. 1989. A survey of the statistical theory of shape. *Statistical Science* 4(2):87–89.

Kosko, B. 1988. Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics* 18(1):49–60.

Kullback, S. 1997. *Information theory and statistics*. Dover Publications.

Palm, G. 2013. Neural associative memories and sparse coding. *Neural Networks* 37:165–171.

Perttula, A.; Carter, S.; and Denoue, L. 2009. Kartta: extracting landmarks near personalized points-of-interest from user generated content. *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* 72.

Ranganathan, A., and Dellaert, F. 2009. Bayesian surprise and landmark detection. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, 2017–2023. IEEE.

Redheffer, R. M. 1951. A note on the surprise index. *The Annals of Mathematical Statistics* 22(1):128–130.

Shannon, C. E. 2001. A Mathematical Theory of Communication. *ACM SIGMOBILE Mobile Computing and Communications Review* 5(1):3–55.

Siagian, C., and Itti, L. 2009. Biologically Inspired Mobile Robot Vision Localization. *IEEE Transactions on Robotics* 25(4):861–873.

Trimble. 2013. Google Sketchup. *Retrieved from http://sketchup.google.com/intl/en/*.

Weaver, W. 1966. Probability, rarity, interest, and surprise. *Pediatrics* 38(4):667–670.

Xia, J. C.; Arrowsmith, C.; Jackson, M.; and Cartwright, W. 2008. The wayfinding process relationships between decision-making and landmark utility. *Tourism Management* 29(3):445–457.

Zhang, L.; Tong, M. H.; and Cottrell, G. W. 2009. SUNDAy: Saliency using natural statistics for dynamic analysis of scenes. In *Proceedings of the 31st Annual Cognitive Science Conference*, 2944–2949.

Computational Models of Surprise in Evaluating Creative Design

Mary Lou Maher[1], Katherine Brady[2], Douglas H. Fisher[2]
[1]Software Information Systems, University of North Carolina, Charlotte, NC
[2]Electrical Engineering & Computer Science, Vanderbilt University, Nashville, TN
m.maher@uncc.edu, katherine.a.brady@vanderbilt.edu, douglas.h.fisher@vanderbilt.edu

Abstract

In this paper we consider how to evaluate whether a design or other artifact is creative. Creativity and its evaluation have been studied as a social process, a creative arts practice, and as a design process with guidelines for people to judge creativity. However, there are few approaches that seek to evaluate creativity computationally. In prior work we presented *novelty*, *value*, and *surprise* as a set of necessary conditions when identifying creative designs. In this paper we focus on the least studied of these – surprise. Surprise occurs when expectations are violated, suggesting that there is a temporal component when evaluating how surprising an artifact is. This paper presents an approach to quantifying surprise by projecting into the future. We illustrate this approach on a database of automobile designs, and we point out several directions for future research in assessing surprising and creativity generally.

Evaluating Creativity and Surprise

As we develop partially and fully automated approaches to computational creativity, the boundary between human creativity and computer creativity blurs. We are interested in approaches to evaluating creativity that make no assumptions about whether the creative entity is a person, a computer, or a collective intelligence of human and computational entities. In short, we want a test for creativity that is not biased by the form of the entity that is doing the creating (Maher and Fisher 2012), but the test should be flexible enough to allow for many forms of creative output. Ultimately, such tests will imbue artificial agents with an ability to assess their own designs and will inform computational models of creative reasoning. Such tests will also inform the design of cognitive assistants that collaborate with humans in sophisticated, socially-intelligent systems.

Evaluating creativity by the characteristics of its results has a long history, including contributions from psychology, engineering, education, and design. Most descriptions of creative designs include *novelty* (sufficiently different from all other designs) and *value* (utilitarian and/or aesthetic) as essential characteristics of a creative artifact (Csikszenmihalyi & Wolfe, 2000; Amabile, 1996; Runco, 2007; Boden, 2003; Wiggins, 2006; Cropley & Cropley, 2005; Besemer & O'Quin, 1987; Horn & Salvendy, 2003; Goldenberg & Mazursky, 2002; Oman and Tumer, 2009; Shah, Smith, & Vargas-Hernandez, 2003).

Surprise is an aspect of creative design that is rarely given attention, even though we believe that it is distinct from novelty and value: a design can be both novel and valuable, but not be surprising. It may be tempting to think that surprise simply stems directly from its "novelty" or difference relative to the set of existing and known artifacts, but we believe that while surprise is related to novelty, it is distinct from novelty as that term is generally construed. In particular, surprise stems from a violation of expectations, and thus surprise can be regarded as "novelty" (or sufficient difference) in a space of projected or expected designs, rather than in a space of existing designs.

In earlier work, Maher and Fisher (2012) presented novelty, value, and surprise as essential and distinct characteristics of a creative design. They also forwarded computational models based on clustering algorithms, which were nascent steps towards automating the recognition of creative designs. This paper takes a closer look at surprise, adding an explicit temporal component to the identification of surprising designs. This temporal component enables a system to make projections about what designs will be expected in the future, so that a system can subsequently assess a new design's differences from expectations, and therefore judge whether a new design deviates sufficiently from expectations to be surprising.

AI Approaches for Assessing Surprise

There is little work on assessing surprise in computational circles; but there has been some, which we survey here.

Horvitz et al (2005) develop a computational model of surprise for traffic forecasting. In this model, they generate probabilistic dependencies among variables, for example linking weather to traffic status. They assume that when an event has less than 2% probability of occurring, it is marked as surprising. They temporally organize the data, grouping incidents into 15-minute intervals. Surprising events in the past are collected in a case library of surprises that is used to identify when a surprising event has occurred. Though related, the concept of rarity as an identifier of something surprising is not the same as difference ("novelty") as an interpretation of surprise – for example, perhaps the rare event differs on only one or two dimen-

sions from other events, and it is these slight differences that make the event rare, and thus surprising.

An important characteristic of the Horvitz et al model is that it makes time explicit, by grouping events into temporal intervals.

A possible limitation of considering rarity as an interpretation of surprise is that as rare events recur, as they are apt to do, many observers would regard them as less surprising. So conditioning surprise by prior precedent might be a very desirable addition to the model. Indeed, Rissland (2009) advances a case-based approach to reasoning about rare and transformative legal cases, where the first appearance of a rare case is surprising and transformative, but subsequent appearances of similar, but still rare events, are neither transformative, nor surprising.

While Rissland's research is not concerned with computational assessment of surprise per se, it recognizes that there are certain legal precedents that radically alter the legal landscape. Rissland calls such precedents 'black swans,' which are rare, perhaps only differing from past legal cases in "small" ways, but they are surprising nonetheless. Importantly, as cases that are similar to the black swan surface, these 'grey cygnets' (as she calls them) are covered by the earlier black swan precedent; a grey cygnet is not transformative and not surprising. The general lesson for approaches to assessing surprise is that rarity may not be enough, because over any sufficient time span the recurrence of rare events is quite likely! But of course, an observer's memory may be limited to a horizon, so that when time intervals are bounded by these horizons, rarity may in fact be a sufficient basis for assessing surprise.

Itti and Baldi (2004) describe a model of surprising features in image data using a priori and posterior probabilities. Given a user dependent model M of some data, there is a P(M) describing the probability distribution. P(M|D) is the probability distribution conditioned on data. Surprise is modeled as the distance d between the prior, P(M), and posterior P(M|D) probabilities. In this model, time is not an explicit attribute or dimension of the data. There are only two times: before and now.

Ranasinghe and Shen (2008) develop a model of surprise as integral to developmental robots. In this model, surprise is used to set goals for learning in an unknown environment. The world is modeled as a set of rules, where each rule has the form: Condition → Action → Predictions. A condition is modeled as: Feature → Operator → Value. For example, a condition can be feature1 > value1 where greater than is the operator. A prediction is modeled as: Feature → Operator. For example, a prediction can be feature1 > where it is expected that feature1 will increase after the action is performed. Comparisons can detect the presence or absence of a feature, and the change in the size of a feature (<, ≤, =, ≥, >). If an observed feature does not match its predicted value, then the system recognizes surprise. This model does not make any explicit reference to time and uses surprise as a flag to update the rule base.

Maher and Fisher (2012) have used clustering algorithms to compare a new design to existing designs, to identify when a design is novel, valuable, and surprising. The clustering model uses distance (e.g., Euclidean distance) to assess novelty and value of product designs (e.g., laptops) that are represented by vectors of attributes (e.g., display area, amount of memory, cpu speed). In this approach, a design is considered surprising when it is so different from existing designs that it forms its own new cluster. This typically happens when the new design makes explicit an attribute that was not previously explicit, because all previous designs had the same value for that attribute. Maher and Fisher use the example of the Bloom laptop, which has a detachable keyboard (i.e., detachable keyboard = TRUE), where all previous laptop designs had value FALSE along what was a previously implicit, unrecognized attribute. Thus, like one of Rissland's black swans, the Bloom transformed the design space.

In Maher and Fisher, the established clusters of design are effectively representing the expectation that the next new design will be associated with one of the clusters of existing designs, and when a new design forms its own cluster it is surprising and changes our expectations for the next generation of new designs.

Maher and Fisher (2012) focused on evaluation of creativity on the part of an observer, not an active designer. Brown (2012) investigates many aspects of surprise in creative design, such as who gets surprised: the designer or the person experiencing or evaluating the design. Brown (2012) also presents a framework for understanding surprise in creative design by characterizing different types of expectations, active, active knowledge, and not active knowledge, as alternative situations in which expectations can be violated in exploratory and transformative design.

To varying extents, many of the computational approaches above model surprise as a deviation from expectation, where the expectation is an expected value that is estimated from data distributions or a prediction made by simulating a rule-based model. In these, however, there is no explicit representation of time as a continuum, nor explicit concern with projecting into the future.

Recognizing Surprising Designs

Our approach to projecting designs into the future assumes that each product design is represented by a vector of ordinal attributes (aka variables). For each attribute, a mathematical function of time can be fit to the attribute values of existing (past) designs, showing how the attribute's values have varied with time in the past. This best fitting function, obtained through a process of regression, can be used to predict how the attribute's values will change in the future as well. Our approach to projecting into the future is inspired by earlier work by Frey and Fisher (1999) that was concerned with projecting machine learning performance curves into the future (thereby allowing cost benefit analyses of collecting more data for purposes of improving prediction accuracy), and it was not concerned with creativity and surprise assessment per se. While Frey and Fish-

er used a variety of functional forms, most notably power functions, as well as linear, logarithmic, and exponential, we have thus far only used linear functions (i.e., univariate linear regression) for projecting designs into the future for purposes of surprise assessment.

In this paper we focus on regression models for recognizing a surprising design: a regression analysis of the attributes of existing designs against a temporal dimension is used to predict the ”next” value of the attributes. The distance from the observed value to the predicted value identifies a surprising attribute-value pair.

We illustrate our use of regression models for identifying surprising designs in an automobile design dataset, which is composed of 572 cars that were produced between 1878 and 2009 (Dowlen, 2012). Each car is described by manufacturer, model, type, year, and nine numerically-valued attributes related to the mechanical design of the car. In this dataset only 190 entries contain values for all nine attributes. These complete entries all occur after 1934 and are concentrated between 1966 and 1994. A summary of the number of designs and the number of attributes in our dataset is shown in Table 1.

Table 1: List of the mechanical design attributes and the number of automobile design records with an entry for each of the nine attributes in our dataset.

Attribute	Number of Designs
Engine Displacement	438
Bore Diameter	407
Stroke Length	407
Torque Force	236
Torque Displacement	235
Weight	356
Frontal Area	337
Maximum Speed	345
Acceleration	290

A variety of linear regression models are considered. The first model uses linear regression over the entire time period of the design data and fits a line to each attribute as a function of time. The results for one attribute, maximum speed, are shown in Figure 1. This analysis identifies the outliers, and therefore potentially surprising designs. For example, the Ferrari 250LM had a surprising maximum speed in 1964, and the Bugati Type 41 Royale has a surprising engine size (another attribute, and another regression analysis) in 1995.

This first model works well for identifying outliers across a time period but does not identify trendsetters (or ‘black swans’ as Rissland might call them) since data points that occurred later in the timeline were included in the regression analysis when evaluating the surprise of a design. A trendsetter is a surprising design that changes the expectations for designs in the future, and is not simply an outlier for all time. In other words, using the entire time line to identify surprising automobile designs does not help us identify those designs that influenced future designers. A design that is an outlier in its own time, but inspires future generations of designers to do something similar can only be found if we don’t use designs which came out after the model being measured in the training data.

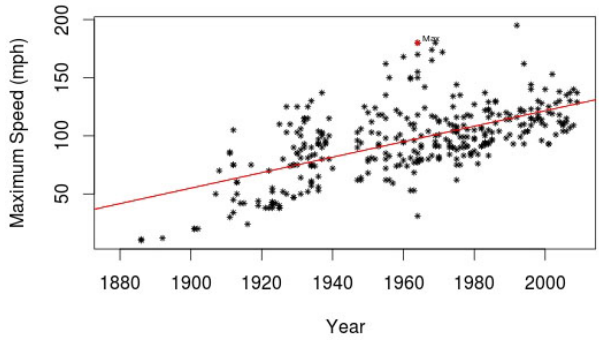


Figure 1. Regression analysis for maximum speed over the entire time period of car design data.

Thus, we considered a second strategy that performs a linear regression only on previously created designs and measures surprise of a new design as the distance from that design’s attribute value to the projection of the line at the year of the design in question. This second regression strategy, where the time period used to fit the line for a single attribute was limited to the time before each design was released (see Figure 2), found roughly the same surprising designs as the first model (over the entire time period) for most attributes, but there were two exceptions: torque displacement and maximum speed. In these exceptions, outliers earlier in time were sufficiently extreme so as to significantly move the entire regression line from before the early outliers to after, whereas in other cases the rough form of the regression lines created over time did not change much.

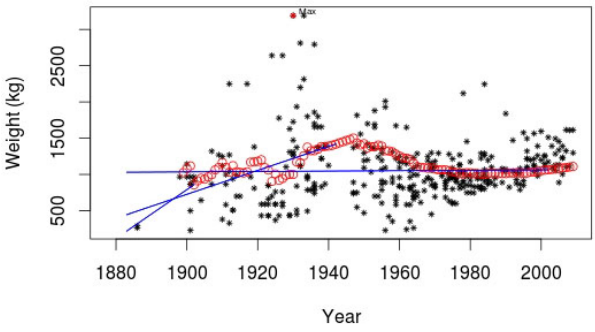


Figure 2: Using strategy 2, linear models are constructed using all previous-year designs. The circles show the predicted (or projected) values for EACH year from the individual regression lines; the dots show actual values. We show three sample regression lines, each ending at the year (circle) it is intended to predict, but there is actually one regression line for each year.

When training this second model, designs from every previous year were weighted equally for predicting future designs. Thus, outliers in the beginning of the dataset perpetually shifted the model and skewed the surprise measurements for all subsequent designs. And why shouldn't they – these early designs correspond roughly to what Rissland called black swans, which understandably diminish the surprise value of subsequent 'grey cygnets'. However, it is also the case that when using model 2, taking into account all past history, that a large mass of 'bland' designs earlier can exaggerate the perceived surprise of a design, even when that design is in the midst of a spurt of like designs.

These observations inspired a third linear regression strategy that makes predictions (or sets expectations) by only including designs within a specified time range before the designs being measured. We use a sliding window, rather than disjoint bins. In either case though, limited time intervals can mimic perceptions of surprise when the observer has a limited memory, only remembering up to a myopic horizon into the past.

The window (aka interval) size used for the cars dataset was ten years. This number was chosen because histograms of the data revealed that all ten-year periods after 1934 contained at least one design with all nine attributes while smaller periods were very sparsely populated in the 1950s. Larger window sizes converged to the second regression model as window size increased.

In general, the size of windows has a large influence on the results. Though we won't delve into the results of this final strategy here, its sensitivity has appeal. In fact, relative to our longer-term goal of modeling human surprise, this sensitivity to window size may map nicely on to different perceptions by people with different experiences. An older adult may have a very different surprise reaction than a young person, depending on past experience. In general, the selection of an appropriate range of years for the third regression model can be correlated with typical periods of time over which a person can remember. That is, if we want to compare our computational model of surprise with human expectations, we should use time intervals that are meaningful to people rather than based on the distribution of data. People will be surprised when expectations based on a time period relevant to their personal knowledge and experience of a series of designs is not met, rather than on the entire time period for all designs.

Directions for Further Research

This paper presents an approach to evaluating whether a design is surprising, and therefore creative, by including a temporal analysis of the conceptual space of existing designs and using regression analysis projected into the future to identify surprising designs. There are a number of directions we plan to follow.

1. We want to further develop the regression models, and in particular move beyond linear regression, to include other functional forms such as polynomial, power, and logarithmic. After all, a design might be regarded as surprising if we used linear regression to project into the future, but not at all surprising if we used a higher-order polynomial regression into the future! Identifying means of distinguishing when one functional form over another is most appropriate for regression will be a key challenge.

2. We want to move beyond our current univariate assessments of surprise through univariate regression, to holistic, multivariate model assessments of surprise through multivariate regression. We can apply multivariate regression methods to designs as a function of time, or combine our earlier work on clustering approaches (Maher and Fisher, 2012) with our regression approaches, perhaps by performing multivariate regression over multivariate summaries of design clusters (e.g., centroids).

3. We have thus far been investigating novelty and value (Maher and Fisher, 2012) and surprise as decoupled characteristics of creativity, but an important next step is to consider how measures of these three characteristics can be integrated into a single holistic measure of creativity, probably parameterized to account for individual differences among observers.

4. Assessments of creativity are conditioned on individual experiences; such individual differences in measures of surprise, novelty, and value are critical – surprise to one person is hardly so to another. We made a barest beginning of this study in Maher and Fisher (2012), where we viewed clustering as the means by which an agent organized its knowledge base, and against which creativity would be judged. The methods for regression that we have presented in this paper will allow us to build in an "imagining" capacity to an agent, adding expectations for designs that do not yet exist to the knowledge base of agents responsible for assessing creativity.

5. In all the variants that we plan to explore, we want to match the results of our models in identifying surprising designs to human judgments of surprise, and of course to assessments of creativity (novelty, value, surprise) of the designs, generally.

6. Finally, our work to date assumes that designs are represented as attribute-value vectors; these propositional representations are clustered in Maher and Fisher (2012), or time-based regression is used in this paper. We want to move to relational models, however, perhaps first-order representations and richer representations still. Relational representations would likely be required in Rissland's legal domain, if in fact that domain were formalized.

A domain that we find very attractive for exploring relational representations is the domain of computer programs, which follow a formal representation and for which a number of well established tools exist for evaluating novelty, value, and surprise. For example, consider that tools for identifying plagiarism in computer programs measure "deep" similarity between programs, and can be adapted as novelty detectors), and for assessing surprise as well.

An ability to measure creativity of "generic" computer programs will allow us to move into virtually any (computable) domain that we want. For example, consider mathematical reasoning in students. In an elementary

course, we can imagine seeing a large number of programs that are designed to compute the variance of data values, as composed of two sequential loops – the first to compute the mean of the data, and the subsequent loop to compute the variance given the mean. These programs will be very similar at a deep level. Imagine then seeing a program that computes the variance (and mean) with ONE loop, relying on a mathematical "simplification." These are the kinds of assessments of creativity that we can expect in more sophisticated relational domains, all enabled by capabilities to assess computer programs.

Acknowledgements: We thank our anonymous reviewers for helpful comments, which guided our revision.

References

Amabile, T. 1982. Social psychology of creativity: A consensual assessment technique. Journal of Personality and Social Psychology 43:997–1013.

Amabile, T. 1996. Creativity in Context: Update to "The Social Psychology of Creativity". Boulder, CO: Westview Press.

Besemer, S., and O'Quin, K. 1987. Creative product analysis: Testing a model by developing a judging instrument. Frontiers of creativity research: Beyond the basics 367–389.

Besemer, S. P., and O'Quin, K. 1999. Confirming the three-factor creative product analysis matrix model in an American sample. Creativity Research Journal 12:287–296.

Boden, M. 2003. The Creative Mind: Myths and Mechanisms, 2nd edition. Routledge.

Brown, D. C. 2012. Creativity, surprise and design: An introduction and investigation. In The 2nd International Conference on Design Creativity (ICDC2012), 75–84.

Cropley, D. H., and Cropley, A. J. 2005. Engineering creativity: A systems concept of functional creativity. In Creativity Across Domains: Faces of the muse, 169–185. Hillsdale, NJ: Lawrence Erlbaum.

Cropley, D. H.; Kaufman, J. C.; and Cropley, A. J. 1991. The assessment of creative products in programs for gifted and talented students. Gifted Child Quarterly 35:128–134.

Cropley, D. H.; Kaufman, J. C.; and Cropley, A. J. 2011. Measuring creativity for innovation management. Journal Of Technology Management & Innovation

Csikszentmihalyi, M., and Wolfe, R. 2000. New conceptions and research approaches to creativity: Implications of a systems perspective for creativity in education. International handbook of giftedness and talent 2:81–91.

Dowlen, C. 2012. Creativity in Car Design – The Behavior At The Edges. A. Duffy, Y. Nagai, T. Taura (eds) Proceedings of the 2nd International Conference on Design Creativity (ICDC2012), 253-262.

Forster, E., and Dunbar, K. 2009. Creativity evaluation through latent semantic analysis. In Proceedings of the Annual Conference of the Cognitive Science Society, 602–607.

Frey, L., and Fisher, D. 1999. Modeling decision tree performance with the power law. In Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, 59–65. Ft. Lauderdale, FL: Morgan Kaufmann.

Goldenberg, J., and Mazursky, D. 2002. Creativity In Product Innovation. Cambridge University Press.

Horn, D., and Salvendy, G. 2003. Consumer-based assessment of product creativity: A review and reappraisal. Human Factors and Ergonomics in Manufacturing & Service Industries 16:155–175.

Horvitz, E.; Apacible, J.; Sarin, R.; and Liao, L. 2005. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. In Proceedings of the 2005 Conference on Uncertainty and Artificial Intelligence. AUAI Press.

Itti L. and Baldi P. (2004). A Surprising Theory of Attention, *IEEE Workshop on Applied Imagery and Pattern Recognition.*

Maher, M. L., and Fisher, D. 2012. Using AI to evaluate creative designs. In A. Duffy, Y. Nagai, T. Taura (eds) Proceedings of the 2nd International Conference on Design Creativity (ICDC2012), 45-54.

Oman, S., and Tumer, I. 2009. The potential of creativity metrics for mechanical engineering concept design. In Bergendahl, M. N.; Grimheden, M.; Leifer, L.; P., S.; and U., L., eds., Proceedings of the 17th International Conference on Engineering Design (ICED'09), Vol. 2, 145–156.

Ranasinghe, N., and Shen, W.-M. 2004. A surprising theory of attention. In IEEE Workshop on Applied Imagery and Pattern Recognition.

Ranasinghe, N., and Shen, W.-M. 2008. Surprise-based learning for developmental robotics. In Proceedings of the 2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems.

Rissland, E. (2009). Black Swans, Gray Cygnets and Other Rare Birds. In L. McGinty and D.C. Wilson (Eds.): ICCBR 2009, LNAI 5650, pp. 6–13, 2009. Springer-Verlag Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007%2F978-3-642-02998-1_2?LI=true#page-1

Runco, M. A. 2007. Creativity: Theories and Themes: Research, Development and Practice. Amsterdam: Elsevier. Shah, J.; Smith, S.; and Vargas-Hernandez, N. 2003. Metrics for measuring ideation effectiveness. Design Studies

24:111–134.

Wiggins, G. 2006. A preliminary framework for description, analysis and comparison of creative systems. Knowledge-Based Systems 16:449–458.

Less Rhyme, More Reason:

Knowledge-based Poetry Generation with Feeling, Insight and Wit

Tony Veale
Web Science & Technology Division, KAIST / School of Computer Science and Informatics, UCD
Korean Advanced Institute of Science & Technology, South Korea / University College Dublin, Ireland.
Tony.Veale@gmail.com

Abstract

Linguistic creativity is a marriage of form and content in which each works together to convey our meanings with concision, resonance and wit. Though form clearly influences and shapes our content, the most deft formal trickery cannot compensate for a lack of real insight. Before computers can be truly creative with language, we must first imbue them with the ability to formulate meanings that are worthy of creative expression. This is especially true of computer-generated poetry. If readers are to recognize a poetic turn-of-phrase as more than a superficial manipulation of words, they must perceive and connect with the meanings and the intent behind the words. So it is not enough for a computer to *merely generate* poem-shaped texts; poems must be driven by conceits that build an affective worldview. This paper describes a conceit-driven approach to computational poetry, in which metaphors and blends are generated for a given topic and affective slant. Subtle inferences drawn from these metaphors and blends can then drive the process of poetry generation. In the same vein, we consider the problem of generating witty insights from the banal truisms of common-sense knowledge bases.

Ode to a Keatsian Turn

Poetic licence is much more than a *licence to frill*. Indeed, it is not so much a licence as a *contract*, one that allows a speaker to subvert the norms of both language and nature in exchange for communicating real insights about some relevant state of affairs. Of course, poetry has norms and conventions of its own, and these lend poems a range of recognizably "poetic" formal characteristics. When used effectively, formal devices such as alliteration, rhyme and cadence can mold our meanings into resonant and incisive forms. However, even the most poetic devices are just empty frills when used only to disguise the absence of real insight. Computer models of poem generation must model more than the frills of poetry, and must instead make these formal devices serve the larger goal of meaning creation.

Nonetheless, is often said that we "eat with our eyes", so that the stylish presentation of food can subtly influence our sense of taste. So it is with poetry: a pleasing form can do more than enhance our recall and comprehension of a meaning – it can also suggest a lasting and profound truth. Experiments by McGlone & Tofighbakhsh (1999, 2000) lend empirical support to this so-called *Keats heuristic*, the intuitive belief – named for Keats' memorable line "Beauty is truth, truth beauty" – that a meaning which is rendered in an aesthetically-pleasing form is much more likely to be perceived as truthful than if it is rendered in a less poetic form. McGlone & Tofighbakhsh demonstrated this effect by searching a book of proverbs for uncommon aphorisms with internal rhyme – such as "*woes unite foes*" – and by using synonym substitution to generate non-rhyming (and thus less poetic) variants such as "*troubles unite enemies*". While no significant differences were observed in subjects' ease of comprehension for rhyming/non-rhyming forms, subjects did show a marked tendency to view the rhyming variants as more truthful expressions of the human condition than the corresponding non-rhyming forms.

So a well-polished poetic form can lend even a modestly interesting observation the lustre of a profound insight. An automated approach to poetry generation can exploit this symbiosis of form and content in a number of useful ways. It might harvest interesting perspectives on a given topic from a text corpus, or it might search its stores of common-sense knowledge for modest insights to render in immodest poetic forms. We describe here a system that combines both of these approaches for meaningful poetry generation.

As shown in the sections to follow, this system – named *Stereotrope* – uses corpus analysis to generate affective metaphors for a topic on which it is asked to wax poetic. *Stereotrope* can be asked to view a topic from a particular affective stance (e.g., view *love negatively*) or to elaborate on a familiar metaphor (e.g. *love is a prison*). In doing so, *Stereotrope* takes account of the feelings that different metaphors are likely to engender in an audience. These metaphors are further integrated to yield tight conceptual blends, which may in turn highlight emergent nuances of a viewpoint that are worthy of poetic expression (see Lakoff and Turner, 1989). *Stereotrope* uses a knowledge-base of conceptual norms to anchor its understanding of these metaphors and blends. While these norms are the stuff of banal clichés and stereotypes, such as that *dogs chase cats* and *cops eat donuts*. we also show how *Stereotrope* finds and exploits corpus evidence to recast these banalities as witty, incisive and poetic insights.

Mutual Knowledge: Norms and Stereotypes

Samuel Johnson opined that "Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it." Traditional approaches to the modelling of metaphor and other figurative devices have typically sought to imbue computers with the former (Fass, 1997). More recently, however, the latter kind has gained traction, with the use of the Web and text corpora to source large amounts of shallow knowledge as it is needed (e.g., Veale & Hao 2007a,b; Shutova 2010; Veale & Li, 2011). But the kind of knowledge demanded by knowledge-hungry phenomena such as metaphor and blending is very different to the specialist "book" knowledge so beloved of Johnson. These demand knowledge of the quotidian world that we all tacitly share but rarely articulate in words, not even in the thoughtful definitions of Johnson's dictionary.

Similes open a rare window onto our shared expectations of the world. Thus, the *as-as*-similes "as hot as an oven", "as dry as sand" and "as tough as leather" illuminate the expected properties of these objects, while the *like*-similes "crying like a baby", "singing like an angel" and "swearing like a sailor" reflect intuitons of how these familiar entities are tacitly expected to behave. Veale & Hao (2007a,b) thus harvest large numbers of *as-as*-similes from the Web to build a rich stereotypical model of familiar ideas and their salient properties, while Özbal & Stock (2012) apply a similar approach on a smaller scale using Google's query completion service. Fishelov (1992) argues convincingly that poetic and non-poetic similes are crafted from the same words and ideas. Poetic conceits use familiar ideas in non-obvious combinations, often with the aim of creating semantic tension. The simile-based model used here thus harvests almost 10,000 familiar stereotypes (drawing on a range of ~8,000 features) from both *as-as* and *like*-similes. Poems construct affective conceits, but as shown in Veale (2012b), the features of a stereotype can be affectively partitioned as needed into distinct *pleasant* and *unpleasant* perspectives. We are thus confident that a stereotype-based model of common-sense knowledge is equal to the task of generating and elaborating affective conceits for a poem.

A stereotype-based model of common-sense knowledge requires both features and relations, with the latter showing how stereotypes relate to each other. It is not enough then to know that cops are tough and gritty, or that donuts are sweet and soft; our stereotypes of each should include the cliché that *cops eat donuts*, just as *dogs chew bones and cats cough up furballs*. Following Veale & Li (2011), we acquire inter-stereotype relationships from the Web, not by mining similes but by mining questions. As in Özbal & Stock (2012), we target query completions from a popular search service (Google), which offers a smaller, public proxy for a larger, zealously-guarded search query log. We harvest questions of the form "*Why do Xs <relation> Ys*", and assume that since each relationship is presupposed by the question (so "why do bikers wear leathers" presupposes that everyone knows that *bikers wear leathers*), the triple of subject/relation/object captures a widely-held norm. In this way we harvest over 40,000 such norms from the Web.

Generating Metaphors, N-Gram Style!

The Google n-grams (Brants & Franz, 2006) is a rich source of popular metaphors of the form *Target is Source*, such as "politicians are crooks", "Apple is a cult", "racism is a disease" and "Steve Jobs is a god". Let *src(T)* denote the set of stereotypes that are commonly used to describe a topic T, where commonality is defined as the presence of the corresponding metaphor in the Google n-grams. To find metaphors for proper-named entities, we also analyse n-grams of the form *stereotype First [Middle] Last*, such as "*tyrant* Adolf Hitler" and "*boss* Bill Gates". Thus, e.g.:

src(racism) = {*problem, disease, joke, sin, poison, crime, ideology, weapon*}

src(Hitler) = {*monster, criminal, tyrant, idiot, madman, vegetarian, racist, …*}

Let *typical*(T) denote the set of properties and behaviors harvested for *T* from Web similes (see previous section), and let *srcTypical*(T) denote the aggregate set of properties and behaviors ascribable to T via the metaphors in *src*(T):

$$(1)\quad srcTypical(T) = \bigcup_{M \in src(T)} typical(M)$$

We can generate conceits for a topic *T* by considering not just obvious metaphors for *T*, but *metaphors of metaphors*:

$$(2)\quad conceits(T) = src(\mathrm{T}) \cup \bigcup_{M \in src(T)} src(M)$$

The features evoked by the conceit *T as M* are given by:

$$(3)\quad salient(T,M) = [srcTypical(\mathrm{T}) \cup typical(\mathrm{T})] \cap [srcTypical(\mathrm{M}) \cup typical(\mathrm{M})]$$

The degree to which a conceit *M* is apt for *T* is given by:

$$(4)\quad aptness(\mathrm{T},\mathrm{M}) = \frac{|salient(T,M) \cap typical(\mathrm{M})|}{|typical(\mathrm{M})|}$$

We should focus on apt conceits $M \in conceits(T)$ where:

$$(5)\quad apt(\mathrm{T},\mathrm{M}) = |salient(T,S) \cap typical(\mathrm{M})| > 0$$

and rank the set of apt conceits by *aptness*, as given in (4).

The set *salient* (*T,M*) identifies the properties / behaviours that are evoked and projected onto *T* when *T* is viewed through the metaphoric lens of *M*. For affective conceits, this set can be partitioned on demand to highlight only the *unpleasant* aspects of the conceit ("you are such a baby!") or only the *pleasant* aspects ("you are my baby!"). Veale & Li (2011) futher show how n-gram evidence can be used to selectively project the salient norms of *M* onto *T*.

Once More With Feeling

Veale (2012b) shows that it is a simple matter to filter a set of stereotypes by affect, to reliably identify the metaphors that impart a mostly positive or negative "spin". But poems are emotion-stirring texts that exploit much more than a crude two-tone polarity. A system like *Stereotrope* should also model the emotions that a metaphorical conceit will stir in a reader. Yet before *Stereotrope* can appreciate the emotions stirred by the properties of a poetic conceit, it must model how properties reinforce and imply each other.

A stereotype is a simplified but coherent representation of a complex real-world phenomenon. So we cannot model stereotypes as simple sets of discrete properties – we must also model how these properties cohere with each other. For example, the property *lush* suggests the properties *green* and *fertile*, while *green* suggests *new* and *fresh*. Let *cohere*(p) denote the set of properties that suggest and reinforce *p*-ness in a stereotye-based description. Thus e.g. *cohere*(*lush*) = {*green, fertile, humid, dense, …*} while *cohere*(*hot*) = {*humid, spicy, sultry, arid, sweaty, …*}. The set of properties that coherently reinforce another property is easily acquired through corpus analysis – we need only look for similes where multiple properties are ascribed to a single topic, as in e.g. "as *hot and humid* as a jungle". To this end, an automatic harvester trawls the Web for instances of the pattern "*as X and Y as*", and assumes for each X and Y pair that Y ∈ *cohere*(X) and X ∈ *cohere*(Y).

Many properties have an emotional resonance, though some evoke more obvious feelings than others. The linguistic mapping from properties to feelings is also more transparent for some property / feeling pairs than others. Consider the property *appalling*, which is stereotypical of *tyrants*: the common linguistic usage "feel appalled by" suggests that an entity with this property is quite likely to make us "feel appalled". Corpus analysis allows a system to learn a mapping from properties to feelings for these obvious cases, by mining instances of the n-gram pattern "feel P+ed by" where P can be mapped to the property of a stereotype via a simple morphology rule. Let *feeling*(p) denote the set of feelings that is learnt in this way for the property *p*. Thus, *feeling*(*disgusting*) = {*feel_disgusted_by*} while *feeling*(*humid*} = {}. Indeed, because this approach can only find obvious mappings, *feeling*(p) = {} for most *p*.

However, *cohere*(p) can be used to interpolate a range of feelings for almost any property *p*. Let *evoke*(p) denote the set of feelings that are likely to be stirred by a property *p*. We can now interpolate *evoke*(p) as follows:

$$(6)\quad evoke(p) = feeling(p) \cup \bigcup_{c \in cohere(p)} feeling(c)$$

So a property *p* also evokes a feeling *f* if *p* suggests another property *c* that evokes *f*. We can predict the range of emotional responses to a stereotype *S* in the same way:

$$(7)\quad evoke(S) = \bigcup_{p \in typical(S)} evoke(p)$$

If *M* is chosen from *conceits*(T) to metaphorically describe *T*, the metaphor *M* is likely to evoke these feelings for *T*:

$$(8)\quad evoke(T, M) = \bigcup_{p \in salient(T, M)} evoke(p)$$

For purposes of gradation, *evoke*(p) and *evoke*(S) denote a *bag* of feelings rather than a *set* of feelings. Thus, the more properties of *S* that evoke *f*, the more times that *evoke*(S) will contain *f*, and the more likely it is that the use of *S* as a conceit will stir the feeling *f* in the reader. *Stereotrope* can thus predict that both *feel disgusted by* and *feel thrilled by* are two possible emotional responses to the property *bloody* (or to the stereotype *war*), and also know that the former is by far the more likely response of the two.

The set *evoke*(*T, M*) for the metaphorical conceit *T is M* can serve the goal of poetry generation in different ways. Most obviously, it is a rich source of feelings that can be explicitly mentioned in a poem about *T* (as viewed thru *M*). Alternately, these feelings can be used in a *meta*-text to motivate and explain the viewpoint of the poem. The act of crafting an explanatory text to showcase a poetry system's creative intent is dubbed *framing* in Colton *et al*. (2012). The current system puts the contents of *evoke*(*T, M*) to both of these uses: in the poem itself, it expresses feelings to show its reaction to certain metaphorical properties of *T*; and in an accompanying framing text, it cites these feelings as a rationale for choosing the conceit *T is M*. For example, in a poem based on the conceit *marriage is a prison*, the set *evoke*(*marriage, prison*) contains the feelings *bored_by*, *confined_in, oppressed_by, chilled_by* and *intimidated_by*. The meta-text that frames the resulting poem expresses the following feelings (using simple NL generation schema):

> "*Gruesome marriage and its depressing divorces appall me. I often feel disturbed and shocked by marriage and its twisted rings. Does marriage revolt you?"*

Atoms, Compounds and Conceptual Blends

If linguistic creativity is chemistry with words and ideas, then stereotypes and their typical properties constitute the periodic table of elements that novel reactions are made of. These are the descriptive atoms that poems combine into metaphorical mixtures, as modeled in (1) … (8) above. But poems can also fuse these atoms into nuanced compounds that may subtly suggest more than the sum of their parts.

Consider the poetry-friendly concept *moon*, for which Web similes provide the following descriptive atoms:

typical(moon) = {*lambent, white, round, pockmarked, shimmering, airless, silver, bulging, cratered, waning, waxing, spooky, eerie, pale, pallid, deserted, glowing, pretty, shining, expressionless, rising*}

Corpus analysis reveals that authors combine atoms such as these in a wide range of resonant compounds. Thus, the Google 2-grams contain such compounds as "pallid glow",

"lambent beauty", "silver shine" and "eerie brightness", all of which can be used to good effect in a poem about the moon. Each compound denotes a compound property, and each exhibits the same linguistic structure. So to harvest a very large number of compound properties, we simply scan the Google 2-grams for phrases of the form "ADJ NOUN", where ADJ and NOUN must each denote a property of the same stereotype. While ADJ maps directly to a property, a combination of morphological analysis and dictionary search is needed to map NOUN to its property (e.g. *beauty* → *beautiful*). What results is a large poetic lexicon, one that captures the diverse and sometimes unexpected ways in which the atomic properties of a stereotype can be fused into nuanced carriers of meaning. Compound descriptions denote compound properties, and those that are shared by different stereotypes reflect the poetic ways in which those concepts are alike. For example, *shining beauty* is shared by over 20 stereotypes in our poetic lexicon, describing such entries as *moon*, *star*, *pearl*, *smile, goddess* and *sky*.

A stereotype suggests behaviors as well as properties, and a fusion of both perspective can yield a more nuanced view. The patterns "VERB ADV" and "ADV VERB" are used to harvest all 2-grams where a property expressed as an adverb qualifies a related property expressed as a verb. For example, the Google 2-gram "glow palely" unites the properties *glowing* and *pale* of *moon*, which allows *moon* to be recognized as similar to *candle* and *ghost* because they too can be described by the compound *glow palely*. A *ghost*, in turn, can *noiselessly glide*, as can a *butterfly*, which may *sparkle radiantly* like a *candle* or a *star* or a *sunbeam*. Not every pairing of descriptive atoms will yield a meaningful compound, and it takes common-sense – or a poetic imagination – to sense which pairings will work in a poem. Though an automatic poet is endowed with neither, it can still harvest and re-use the many valid combinations that humans have added to the language trove of the Web.

Poetic allusions anchor a phrase in a vivid stereotype while shrouding its meaning in constructive ambiguity. Why talk of the *pale glow* of the moon when you can allude to its *ghostly glow* instead? The latter does more than evoke the moon's paleness – it attributes this paleness to a supernatural root, and suggests a halo of other qualities such as *haunting*, *spooky*, *chilling* and *sinister*. Stereotypes are dense descriptors, and the use of one to convey a single property like *pale* will subtly suggest other readings and resonances. The phrase "ghostly glow" may thus allude to any corpus-attested compound property that can be forged from the property *glowing* and any other element of the set *typical(ghost)*. Many stereotype nouns have adjectival forms – such as *ghostly* for *ghost*, *freakish* for *freak*, *inky* for *ink* – and these may be used in corpora to qualify the nominal form of a property of that very stereotype, such as *gloom* for *gloomy*, *silence* for *silent, or pallor for pale*. The 2-gram "inky gloom" can thus be understood as an allusion either to the *blackness* or *wetness* of *ink*, so any stereotype that combines the properties *dark* and *wet* (e.g. *oil*, *swamp*, *winter*) or *dark* and *black* (e.g. *crypt*, *cave*, *midnight*) can be poetically described as exhibiting an *inky gloom*.

Let *compounds*(…) denote a function that maps a set of atomic properties such as *shining* and *beautiful* to the set of compound descriptors – such as the compound property *shining beauty* or the compound allusion *ghostly glow* – that can be harvested from the Google 2-grams. It follows that *compounds(typical(S))* denotes the set of corpus-attested compounds that can describe a stereotype *S*, while *compounds(salient(T, M))* denotes the set of compound descriptors that might be used in a poem about *T* to suggest the poetic conceit *T is M*. Since these compounds will fuse atomic elements from the stereotypical representations of both *T* and *M*, *compounds(salient(T, M))* can be viewed as a blend of *T* and *M*. As described in Fauconnier & Turner (2002), and computationally modeled in various ways in Veale & O'Donoghue (2000), Pereira (2007) and Veale & Li (2011), a "blend" is a tight conceptual integration of two or more mental spaces. This integration yields more than a mixture of representational atoms: a conceptual blend often creates emergent elements – new molecules of meaning – that are present in neither of the input representations but which only arise from the fusion of these representations.

How might the representations discussed here give rise to emergent elements? We cannot expect new descriptive atoms to be created by a poetic blend, but we can expect new compounds to emerge from the re-combination of descriptive atoms in the compound descriptors of *T* and *M*. Just as we can expect *compounds(typical(T)* ∪ *typical(M))* to suggest a wider range of descriptive possibilities than *compounds(typical(T))* ∪ *compounds(typical(M))*, we say:

$$\text{(9)}\quad emergent(\mathrm{T},\mathrm{M}) = \{p \in compounds(salient(T,M)) \mid p \notin compounds(typical(T)) \wedge p \notin compounds(typical(M))\}$$

In other words, the compound descriptions that emerge from the blend of *T* and *M* are those that could not have emerged from the properties of *T* alone, or from *M* alone, but can only emerge from the fusion of *T* and *M* together.

Consider the poetic conceit *love is the grave*. The resulting blend – as captured by *compounds(salient(T, M))* – contains a wide variety of compound descriptors. Some of these compounds emerge solely from the concept *grave*, such as *sacred gloom*, *dreary chill* and *blessed stillness*. Many others emerge only from a fusion of *love* and *grave*, such as *romantic stillness*, *sweet silence*, *tender darkness, cold embrace, quiet passion* and *consecrated devotion*. So a poem that uses these phrases to construct an emotional worldview will not only demonstrate an understanding of its topic and its conceit, but will also demonstrate some measure of insight into how one can complement and resonate with the other (e.g., that darkness can be tender, passion can be quiet and silence can be sweet). While the system builds on second-hand insights, insofar as these are ultimately derived from Web corpora, such insights are fragmentary and low-level. It still falls to the system to stitch these into its own emotionally coherent patchwork of poetry. What use is poetry if we or our machines cannot learn from it the wild possibilities of language and life?

Generating Witty Insights from Banal Facts

Insight requires depth. To derive original insights about the topic of a poem, say, of a kind an unbiased audience might consider witty or clever, a system needs more than shallow corpus data; it needs deep knowledge of the real world. It is perhaps ironic then that the last place one is likely to find real insight is in the riches of a structured knowledge base. Common-sense knowledge-bases are especially lacking in insight, since these are designed to contain knowledge that is common to all and questioned by none. Even domain-specific knowledge-bases, rich in specialist knowledge, are designed as repositories of axiomatic truths that will appear self-evident to their intended audience of experts.

Insight is both a process and a product. While insight undoubtedly requires knowledge, it also takes work to craft surprising insights from the unsurprising generalizations that make up the bulk of our conventional knowledge. Though mathematicians occasionally derive surprising theorems from the application of deductive techniques to self-evident axioms, sound reasoning over unsurprising facts will rarely yield surprising conclusions. Yet witty insights are not typically the product of an entirely sound reasoning process. Rather, such insights amuse and provoke via a combination of over-statement, selective use of facts, a mixing of distinct knowledge types, and a clever packaging that makes maximal use of the Keats heuristic. Indeed, as has long been understood by humor theorists, the logic of humorous insight is deeply bound up with the act of framing. The *logical mechanism* of a joke – a kind of pseudological syllogism for producing humorous effects – is responsible for framing a situation in such a way that it gives rise to an unexpected but meaningful incongruity (Attardo & Raskin, 1992; Attardo *et al*., 2002). To craft witty insights from innocuous generalities, a system must draw on an arsenal of such logical mechanisms to frame its observations of the world in appeallingly discordant ways.

Attardo and Raskin view the role of a logical mechanism (LM) as the engine of a joke: each LM provides a different way of bringing together two overlapping scripts that are mutually opposed in some pivotal way. A joke narrative is fully compatible with one of these scripts and only partly compatible with the other, yet it is the partial match that we, as listeners, jump to first to understand the narrative. In a well-structured joke, we only recognize the inadequacy of this partially-apt script when we reach the punchline, at which point we switch our focus to its unlikely alternative. The realization that we can easily duped by appearances, combined with the sense of relief and understanding that this realization can bring, results in the AHA! feeling of insight that often accompanies the HA-HA of a good joke. LMs suited to narrative jokes tend to engineer oppositions between narrative scripts, but for purposes of crafting witty insights in one-line poetic forms, we will view a script as a stereotypical representation of an entity or event. Armed with an arsenal of stereotype "scripts", *Stereotrope* will seek to highlight the tacit opposition between different stereotypes as they typically relate to each other, while also engineering credible oppositions based on corpus evidence.

A sound logical system cannot not brook contradictions. Nonetheless, uncontroversial views can be cleverly framed in such a way that they appear sound *and* contradictory, as when the columnist David Brooks described the Olympics as a "peaceful celebration of our warlike nature". His form has symmetry and cadence, and pithily exploits the Keats heuristic to reconcile two polar opposites, *war* and *peace*. Poetic insights do not aim to create real contradictions, but aim to reveal (and reconcile) the unspoken tensions in familiar ideas and relationships. We have discussed two kinds of stereotypical knowledge in this paper: the property view of a stereotype *S*, as captured in *typical*(*S*), and the relational view, as captured by a set of question-derived generalizations of the form *Xs <relation> Ys*. A blend of both these sources of knowledge can yield emergent oppositions that are not apparent in either source alone.

Consider the normative relation *bows fire arrows*. Bows are stereotypically *curved*, while arrows are stereotypically *straight*, so lurking beneath the surface of this innocuous norm is a semantic opposition that can be foregrounded to poetic effect. The Keats heuristic can be used to package this opposition in a pithy and thought-provoking form: thus compare "*curved bows fire straight arrows*" (so what?) with "*straight arrows do curved bows fire*" (more poetic) and "*the most curved bows fire the straightest arrows*" (most poetic). While this last form is an overly strong claim that is not strictly supported by the stereotype model, it has the sweeping form of a penetrating insight that grabs one's attention. Its pragmatic effect – a key function of poetic insight – is to reconcile two opposites by suggesting that they fill complementary roles. In schematic terms, such insights can be derived from any single norm of the form *Xs <relation> Ys* where X and Y denote stereotypes with salient properties – such as *soft* and *tough, long* and *short* – that can be framed in striking opposition. For instance, the combination of the norm *cops eat donuts* with the clichéd views of cops as *tough* and donuts as *soft* yields the insight "*the toughest cops eat the softest donuts*". As the property *tough* is undermined by the property *soft*, this may be viewed as a playful subversion of the *tough cop* stereotype. The property *toughness* is can be further subverted, with an added suggestion of hypocrisy, by expressing the generalization as a rhetorical question: "*Why do the toughest cops eat the softest donuts*?"

A single norm represents a highly simplified script, so a framing of two norms together often allows for opposition via a conflict of overlapping scripts. Activists, for example, typically engage in tense struggles to achieve their goals. But activists are also known for the slogans they coin and the chants they sing. Most slogans, whether designed to change the law or sell detergent, are catchy and uplifting. These properties and norms can now be framed in poetic opposition: "*The activists that chant the most uplifting slogans suffer through the most depressing struggles*". While the number of insights derivable from single norms is a linear function of the size of the knowledge base, a combinatorial opportunity exists to craft insights from pairs of norms. Thus, "*angels who fight the foulest demons*

play the sweetest harps", "*surgeons who wield the most hardened blades wear the softest gloves*", and "*celebrities who promote the most reputable charities suffer the sleaziest scandals*" all achieve conflict through norm juxtaposition. Moreover, the order of a juxtaposition – positive before negative or vice versa – can also sway the reader toward a cynical or an optimistic interpretation.

Wit portrays opposition as an inherent part of reality, yet often creates the oppositions that it appears to reconcile. It does so by elevating specifics into generalities, to suggest that opposition is the norm rather than the exception. So rather than rely wholly on stereotypes and their expected properties, *Stereotrope* uses corpus evidence as a proxy imagination to concocts new classes of individuals with interesting and opposable qualities. Consider the Google 2-gram "*short celebrities*", whose frequency and plurality suggests that shortness is a noteworthy (though not typical) property of a significant class of celebrities. *Stereotrope* already possesses the norm that "*celebrities ride in limousines*", as well as a stereotypical expectation that *limousines* are *long*. This juxtaposition of conventions allows it to frame a provocatively sweeping generalization: "*Why do the shortest celebrities ride in the longest limousines*?" While *Stereotrope* has no evidence for this speculative claim, and no real insight into the status-anxiety of the rich but vertically-challenged, such an understanding may follow in time, as deeper and subtler knowledge-bases become available for poetry generation.

Poetic insight often takes the form of sweeping claims that elevate vivid cases into powerful exemplars. Consider how *Stereotrope* uses a mix of n-gram evidence and norms to generate these maxims: "*The most curious scientists achieve the most notable breakthroughs*" and "*The most impartial scientists use the most accurate instruments*". The causal seeds of these insights are mined from the Google n-grams in coordinations such as "*hardest and sharpest*" and "*most curious and most notable*". These n-gram relationships are then be projected onto banal norms – such as *scientists achieve breakthroughs* and *scientists use instruments* – for whose participants these properties are stereotypical (e.g. *scientists* are *curious* and *impartial*, *instruments* are *accurate*, *breakthroughs* are *notable*, etc.).

Such claims can be taken literally, or viewed as vivid allusions to important causal relationships. Indeed, when framed as explicit analogies, the juxtaposition of two such insights can yield unexpected resonances. For example, "*the most trusted celebrities ride in the longest limousines*" and "*the most trusted preachers give the longest sermons*" are both inspired by the 4-gram "*most trusted and longest.*" This common allusion suggests an analogy: "*Just as the most trusted celebrities ride in the longest limousines, the most trusted preachers give the longest sermons"*. Though such analogies are driven by superficial similarity, they can still evoke deep resonances for an audience. Perhaps a sermon is a vehicle for a preacher's ego, just as a limousine is an obvious vehicle for a celebrity's ego? Reversing the order of the analogy significantly alters its larger import, suggesting that ostentatious wealth bears a lesson for us all.

Tying it all together in *Stereotrope*

Having created the individual pieces of form and meaning from which a poem might be crafted, it now falls to us to put the pieces together in some coherent form. To recap, we have shown how affective metaphors may be generated for a given topic, by building on popular metaphors for that topic in the Google n-grams; shown how a tight conceptual blend, with emergent compound properties of its own, can be crafted from each of these metaphors; shown how the feelings evoked by these properties may be anticipated by a system; and shown how novel insights can be crafted from a fusion of stereotypical norms and corpus evidence.

We view a poem as a summarization and visualization device that samples the set of properties and feelings that are evoked when a topic *T* is viewed as *M*. Given *T*, an *M* is chosen randomly from *conceits*(T). Each line of the text renders one or more properties in poetic form, using tropes such as simile and hyperbolae. So if *salient(T, M)* contains *hot* and *compounds(salient(T, M))* contains *burn brightly* – for *T=love* and *M=fire*, say – this mix of elements may be rendered as "*No fire is hotter or burns more brightly*". It can also be rendered as an imperative, "*Burn brightly with your hot love*", or a request, "*Let your hot love burn brightly*". The range of tropes is best conveyed with examples, such as this poetic view of *marriage* as a *prison*:

The legalized regime of this marriage

My marriage is an emotional prison
Barred visitors do marriages allow
The most unitary collective scarcely organizes so much
Intimidate me with the official regulation of your prison
Let your sexual degradation charm me
Did ever an offender go to a more oppressive prison?
You confine me as securely as any locked prison cell
Does any prison punish more harshly than this marriage?
You punish me with your harsh security
The most isolated prisons inflict the most difficult hardships
O Marriage, you disgust me with your undesirable security

Each poem obeys a semantic grammar, which minimally indicates the trope that should be used for each line. Since the second-line of the grammar asks for an apt *<simile>*, *Stereotrope* constructs one by comparing *marriage* to a *collective*; as the second-last line asks for an apt *<insight>*, one is duly constructed around the Google 4-gram "*most isolated and most difficult*". The grammar may also dictate whether a line is rendered as an assertion, an imperative, a request or a question, and whether it is framed positively or negatively. This grammar need not be a limiting factor, as one can choose randomly from a pool of grammars, or even evolve a new grammar by soliciting user feedback. The key point is the pivotal role of a grammar of tropes in mapping from the properties and feelings of a metaphorical blend to a sequence of poetic renderings of these elements.

Consider this poem, from the metaphor *China is a rival*:

No Rival Is More Bitterly Determined

Inspire me with your determined battle
The most dogged defender scarcely struggles so much
Stir me with your spirited challenge
Let your competitive threat reward me
Was ever a treaty negotiated by a more competitive rival?
You compete with me like a competitively determined athlete
Does any rival test more competitively than this China?
You oppose me with your bitter battle
Can a bitter rival suffer from such sweet jealousies?
O China, you oppress me with your hated fighting

Stereotypes are most eye-catching when subverted, as in the second-last line above. The Google 2-gram "*sweet jealousies*" catches *Stereotrope*'s eye (and ours) because it up-ends the belief that *jealousy* is a *bitter* emotion. This subversion nicely complements the sterotype that *rivals* are *bitter*, allowing *Stereotrope* to impose a thought-provoking opposition onto the banal norm *rivals suffer from jealousy*.

Stereotype emphasises meaning and intent over sound and form, and does not (yet) choose lines for their rhyme or metre. However, given a choice of renderings, it does choose the form that makes best use of the Keats heuristic, by favoring lines with alliteration and internal symmetry

Evaluation

Stereotrope is a knowledge-based approach to poetry, one that crucially relies on three sources of inspiration: a large roster of stereotypes, which maps a slew of familiar ideas to their most salient properties; a large body of normative relationships which relate these stereotypes to each other; and the Google n-grams, a vast body of language snippets. The first two are derived from attested language use on the web, while the third is a reduced view of the linguistic web itself. *Stereotrope* represents approx. 10,000 stereotypes in terms of approx. 75,000 stereotype-to-property mappings, where each of these is supported by a real web simile that attests to the accepted salience of a given property. In addition, *Stereotrope* represents over 50,000 norms, each derived from a presupposition-laden question on the web.

The reliability of *Stereotrope*'s knowledge has been demonstrated in recent studies. Veale (2012a) shows that *Stereotrope*'s simile-derived representations are balanced and unbiased, as the positive/negative affect of a stereotype *T* can be reliably estimated as a function of the affect of the contents of *typical(T)*. Veale (2012b) further shows that *typical(T)* can be reliably partitioned into sets of positive or negative properties as needed, to reflect an affective "spin" imposed by any given metaphor *M*. Moreover, Veale (ibid) shows that copula metaphors of the form *T is an M* in the Google n-grams – the source of *srcTypical*(*T*) – are also broadly consistent with the properties and affective profile of each stereotype *T*. So in **87%** of cases, one can correctly assign the label *positive* or *negative* to a topic *T* using only the contents of *srcTypical*(*T*), provided it is not empty.

Stereotrope derives its appreciation of feelings from its understanding of how one property presupposes another. The intuition that two properties X and Y that are found in the pattern "*as X and Y as*" evoke similar feelings is supported by the strong correlation (**0.7)** observed between the positivity of X and of Y over the many X/Y pairs that are harvested from the web using this acquisition pattern.

The "fact" that *bats lay eggs* can be found over 40,000 times on the web via Google. On closer examination, most matches form part of a larger question, "*do bats lay eggs*?" The question "*why do bats lay eggs?*" has zero matches. So "*Why do*" questions provide an effective superstructure for acquiring normative facts from the web: they identify facts that are commonly presupposed, and thus stereotypical, and clearly mark the start and end of each presupposition. Such questions also yield *useful* facts: Veale & Li (2011) shows that when these facts are treated as features of the stereotypes for which they are presupposed, they provide an excellent basis for classifying different stereotypes into the same ontological categories, as would be predicted by an ontology such as WordNet (Fellbaum, 1998). Moreover, these features can be reliably distributed to close semantic neighbors to overcome the problem of knowledge *sparsity*. Veale & Li demonstrate that the likelihood that a feature of stereotype A can also be assumed of stereotype B is a clear function of the WordNet similarity of A and B. While this is an intuitive finding, it would not hold at all if not for the fact that these features are truly meaningful for A (and B).

The problem posed by "bats lay eggs" is one faced by any system that does not perceive the whole context of an utterance. As such, it is a problem that plagues the use of n-gram models of web content, such as Google's n-grams. *Stereotrope* uses n-grams to suggest insightful connections between two properties or ideas, but if these n-grams are mere noise, not even the Keats heuristic can disguise them as meaningful signals. Our focus is on relational n-grams, of a kind that suggests deep tacit relationships between two concepts. These n-grams obey the pattern "X <rel> Y", where X and Y are adjectives or nouns and *<rel>* is a linking phrase, such as a verb, a preposition, a coordinator, etc. To determine the quality of these n-grams, and to assess the likelihood of extracting genuine relational insights from them, we use this large subset of the Google n-grams as a corpus for estimating the relational similarity of the 353 word pairs in the Finklestein *et al*. (2002) WordSim-353 data set. We estimate the relatedness of two words X and Y as the PMI (pointwise mutual information score) of X and Y, using the relational n-grams as a corpus for occurrence and co-occurrence frequencies of X and Y. A correlation **of 0.61** is observed between these PMI scores and the human ratings reported by Finklestein *et al*. (2002). Though this is not the highest score achieved for this task, it is considerably higher than any than has been reported for approaches that use WordNet alone. The point here is that this relational subset of the Google n-grams offers a reasonably faithful mirror of human intuitions for purposes of recognizing the relatedness of different ideas. We thus believe these n-grams to be a valid source of real insights.

The final arbiters of *Stereotrope*'s poetic insights are the humans who use the system. We offer the various services of *Stereotrope* as a public web service, via this URL:

http://boundinanutshell.com/metaphor-magnet

We hope these services will also allow other researchers to reuse and extend *Stereotrope*'s approaches to metaphor, blending and poetry. Thus, for instance, poetry generators such as that described in Colton *et al*. (2012) – which creates topical poems from fragments of newspapers and tweets – can use *Stereotrope*'s rich inventories of similes, poetic compounds, feelings and allusions in its poetry.

Summary and Conclusions

Poets use the Keats heuristic to distil an amorphous space of feelings and ideas into a concise and memorable form. Poetry thus serves as an ideal tool for summarizing and visualizing the large space of possibilities that is explored whenever we view a familiar topic from a new perspective. In this paper we have modelled poetry as both a product and an expressive tool, one that harnesses the processes of *knowledge acquisition* (via web similes and questions), *ideation* (via metaphor and insight generation), *emotion* (via a mapping of properties to feelings), *integration* (via conceptual blending) and *rendering* (via tropes that map properties and feelings to poetic forms). Each of these processes has been made publicly available as part of a comprehensive web service called *Metaphor Magnet*.

We want our automated poets to be able to formulate real meanings that are worthy of poetic expression, but we also want them to evoke much more than they actually say. The pragmatic import of a creative formulation will always be larger than the system's ability to model it accurately. Yet the human reader has always been an essential part of the poetic process, one that should not be downplayed or overlooked in our desire to produce computational poets that fully understand their own outputs. So for now, though there is much scope, and indeed *need*, for improvement, it is enough to know that an automated poem is anchored in real meanings and intentional metaphors, and to leave certain aspects of creative interpretation to the audience.

Acknowledgements

This research was supported by the WCU (World Class University) program under the National Research Foundation of Korea (Ministry of Education, Science and Technology of Korea, Project no. R31-30007).

References

Attardo, S. and Raskin, V. 1991. Script theory revis(it)ed: joke similarity and joke representational model. *Humor: International Journal of Humor Research,* **4**(3):293-347.

Attardo, S., Hempelmann, C.F. & Di Maio, S. 2002. Script oppositions and logical mechanisms: Modeling incongruities and their resolutions. *Humor: Int. J. of Humor Research,* **15**(1):3-46.

Brants, T. & Franz, A. 2006. Web 1T 5-gram Version 1. *Linguistic Data Consortium*.

Colton, S., Goodwin, J. and Veale, T. 2012. Full-FACE Poetry Generation. *In Proc. of ICCC 2012, the 3rd International Conference on Computational Creativity*. Dublin, Ireland.

Fass, D. 1997. Processing Metonymy & Metaphor. *Contemporary Studies in Cognitive Science & Technology*. New York: Ablex.

Fauconnier, G. & Turner, M. 2002. *The Way We Think. Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.

Fellbaum, C. (ed.) 2008. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. and Ruppin, E. 2002. Placing Search in Context: The concept revisited. *ACM Transactions on Information Systems, 20(1):116-131*.

Lakoff, G. & Turner, M. 1989. *More than cool reason: a field guie to poetic metaphor*. University of Chicago Press.

McGlone, M.S. & Tofighbakhsh, J. 2000. Birds of a feather flock conjointly (?): rhyme as reason in aphorisms. *Psychological Science* **11** (5): 424–428.

Fishelov, D. 1992. Poetic and Non-Poetic Simile: Structure, Semantics, Rhetoric. *Poetics Today* 14(1):1–23.

McGlone, M.S. & Tofighbakhsh, J. 1999. The Keats heuristic: Rhyme as reason in aphorism interpretation. *Poetics* **26**(4):235-44.

Özbal, G. and C. Strapparava. 2012. A computational approach to automatize creative naming. In *Proc. of the 50th annual meeting of the Association of Computational Linguistics*, Jeju, South Korea.

Pereira, F. C. 2007. *Creativity and artificial intelligence: a conceptual blending approach*. Walter de Gruyter.

Shutova, E. 2010. Metaphor Identification Using Verb and Noun Clustering. *In Proceedings of the 23rd International Conference on Computational Linguistics*, 1001-10.

Veale, T. & D. O'Donoghue. 2000. Computation and Blending. *Cognitive Linguistics*, 11(3-4):253-281.

Veale, T. and Hao, Y. 2007a. Making Lexical Ontologies Functional and Context-Sensitive. In *Proceedings of the 46th Ann. Meeting of Assoc. of Computational Linguistics*.

Veale T. & Hao, Y. 2007b. Comprehending and generating apt metaphors: a web-driven, case-based approach to figurative language. In *Proceedings of AAAI'2007, the 22nd national conference on Artificial intelligence*, pp.1471-1476.

Veale, T. & Li, G. 2011. Creative Introspection and Knowledge Acquisition: Learning about the world thru introspective questions and exploratory metaphors. In Proc. of AAAI'2011, the 25th Conference of the Association for the Advancement of AI.

Veale, T. 2012a. *Exploding the Creativity Myth: Computational Foundations of Linguistic Creativity*. London: Bloomsbury.

Veale, T. 2012b. A Context-sensitive, Multi-faceted model of Lexico-Conceptual Affect. In *Proc. of the 50th annual meeting of the Association of Computational Linguistics*. Jeju, South Korea.

Harnessing Constraint Programming for Poetry Composition

Jukka M. Toivanen and **Matti Järvisalo** and **Hannu Toivonen**
HIIT and Department of Computer Science
University of Helsinki
Finland

Abstract

Constraints are a major factor shaping the conceptual space of many areas of creativity. We propose to use constraint programming techniques and off-the-shelf constraint solvers in the creative task of poetry writing. We show how many aspects essential in different poetical forms, and partially even in the level of language syntax and semantics can be represented as interacting constraints.

The proposed architecture has two main components. One takes input or inspiration from the user or the environment, and based on it generates a specification of the space and aesthetic of a poem as a set of declarative constraints. The other component explores the specified space using a constraint solver.

We provide an elementary set of constraints for composition of poetry, we illustrate their use, and we provide examples of poems generated with different sets of constraints.

Introduction

Rules and constraints can be seen as an essential ingredient of creativity. First, there typically are strong constraints on the creative artefacts. For instance, consider traditional western music. In order for a composition to be recognized as (western) music in the first place, it must meet a number of requirements concerning, e.g., timbre, scale, melody, harmony, and rhythm. For any specific genre of western music, the constraints usually become much tighter.

Similarly, the composition of many types of poetry is governed by numerous rules specifying such things as strict stress and syllable patterns, rhyming and alliteration structures, and selection of words with certain associations — in addition to the basic constraints of syntax and semantics that are needed to make the expressions understandable and meaningful.

However, constraints are not just a nuisance that creative agents need to cope with in order to produce plausible results. On the contrary, constraints are often considered to be an essential source of creativity for humans. For instance, composer Igor Stravinsky associated constraints with creating freedom, not containment:

> *"The more constraints one imposes, the more one frees one's self of the chains that shackle the spirit."*
>
> (Stravinsky 1947)

Constraints can also be used as computational tools for studies of creativity or creative artefacts. Artificial intelligence researcher Marvin Minsky suggested that a good way to learn about how music "worked" was to represent musical compositions as interacting constraints, then modify these constraints and study their effects on the musical structures (Roads 1980). This essential idea has been explored extensively in the field of computer music research afterwards.

Our domain of interest in this paper is composition of poetry. We envision a computational environment where formally expressed constraints and constraint programming methods are used to (1) specify a conceptual search space, (2) define an aesthetic of concepts in the space, (3) explore the space to find the most aesthetic concepts in it.

Any given set of (hard) constraints on poems specifies a space of possible poems. For instance, the number of lines and the number of syllables per line could be such constaints, contributing to the style of poetry. Soft constraints, in turn, can be used to indicate (aesthetical) preferences over poems and to rank poems that match the hard constraints. For instance, rhyme could be a soft constaint, giving preference to poems that follow a given rhyme structure but not absolutely requiring it.

In this paper we study and illustrate the power of constraint programming for creating poems. In our current setup, the creative system consists of two subcomponents. One takes input from user or from some other source of inspiration, and based on it *specifies* the space and poetical aesthetic (as a set of constraints). The other subcomponent *explores* the specified space using the aesthetic, i.e., produces optimally aesthetic poems in the space (using a constraint solver).

We show how poems can be generated by applying different kinds of constraints and constraint combinations using an off-the-shelf constraint programming tool. The elegance of this approach is that it is not based on specifying a step sequence to produce a certain kind of a poem, but rather on declaring the properties of a solution to be found using mathematical constraints. An empirical evaluation of the obtained poetry is left for future work.

We next briefly review some related work on constraint programming in creative applications, and on poetry generation. Then we provide a description of a constraint model for composing poems, illustrating the ideas with examples. We

discuss the results and conclude by outlining future work.

Related Work

Constraint-based methods have been applied in various fields such as configuration and verification, planning, and evolution of language, to name a few. In the area of computational creativity, constraints have been used mostly to describe the composition of various aspects of music. For example, Boenn et al. (2011) have developed an extensive music composition system called Anton which uses Answer Set Programming to represent the musical knowledge and the rules of the system. Anton describes a model of musical composition as a collection of interacting constraints. The system can be used to compose short pieces of music as well as to assist the composer by making suggestions, completions, and verifications to aid in the music composition process.

On the other hand, composition of poetry with constraint programming techniques has received little if any attention. Several different approaches have been used (Manurung, Ritchie, and Thompson 2000; Gervás 2001; Manurung 2003; Diaz-Agudo, Gervás, and González-Calero 2002; Wong and Chun 2008; Netzer et al. 2009; Colton, Goodwin, and Veale 2012; Toivanen et al. 2012), many involving constraints in one form or another, but we are not aware of any other work systematically based on constraints and implemented using a constraint solver.

The system developed by Manurung et al. (2003) uses a grammar-driven formulation to generate metrically constrained poetry out of a given topic. This approach performs stochastic hillclimbing search within an explicit state-space, moving from one solution to another. The explicit representation is based on a hand-crafted transition system. In contrast, we employ constraint-programming methodology based on searching for optimal solutions over an implicit representation of the conceptual space. Our approach should scale better to large numbers of constraints and a large input vocabulary than explicit state-space search.

The ASPERA poetry composition system (Gervás 2001), on the other hand, uses a case-based reasoning approach. This system generates poetry out of a given input text via composition of poetic fragments retrieved from a case-base of existing poetry. These fragments are then combined together by using additional metrical rules.

The Full-FACE poetry generation system (Colton, Goodwin, and Veale 2012) uses a corpus-based approach to generate poetry according to given constraints on, for instance, meter and stress. The system is also argued to invent its own aesthetics and framings of its work. In contrast to our system, this approach uses constraints to shape only some aspects of the poetry composition procedure whereas our approach is fully based on expressing various aspects of poetry as mutually interacting constraints and using a constraint-solver to efficiently search for solutions.

The approach of this paper extends and complements our previous work (Toivanen et al. 2012). We proposed a method where a template is extracted randomly from a given corpus, and words in the template are substituted by words related to a given topic. Here we show how such basic functionality can be expressed with constraints, and more interestingly, how constraint programming can be used to add control for rhyme, meter, and other effects.

Simpler poetry generation methods have been proposed, as well. In particular, Markov chains have been widely used to compose poetry. They provide a clear and simple way to model some syntactic and semantic characteristics of language (Langkilde and Knight 1998). However, the resulting poetry tends to have rather poor sentence and poem structures due to only local syntax and semantics.

Overview

The proposed poetry composition system has two subcomponents: a conceptual space *specifier* and a conceptual space *explorer*. The former one determines what poems can be like and what kind of poems are preferred, while the latter one assumes the task of producing such poems.

The modularity and the explicit specification of the conceptual search space have great potential benefits. Modularity allows one to (partially) separate the content and form of poetry from the computation needed to produce matching poetry. An explicit, declarative specification, in turn, gives the creative system a potential to introspect and modify its own goals and intermediate results (a topic to which we will return in the conclusion).

A high-level view to the internal structure of the poetry composition system considered in this work is shown in Figure 1. In this paper, our focus is on the explorer component and on the interface between the components. Our specifier component is built on the principles of Toivanen et al. (2012), but ideas from many other poetry generation systems (Gervás 2001; Manurung 2003; Colton, Goodwin, and Veale 2012) could be used in the specifier component as well.

The assumption in the model presented here is that the specifier can generate a large number of mutually dependent choices of words for different positions in the poem, as well as dependencies between them. The specifier uses input from the user and potentially other sources as its inspiration and parameters and automatically generates the input for the explorer component, shielding user from the details of constraint programming.

The automatically generated "data" or "facts" are conveyed to the explorer component that consists of a constraint solver and a static library of constraints. The library is provided by the system designers, i.e., by us, and any constraints that the specifier component wishes to use are triggered by the data it generates. The user of the system does not need to interact directly with the constraint library (but the specifier component may offer the user options for choosing which constraints to use).

Our focus in this paper is on the explorer component, and in the constraint specifications that it receives from the specifier component or from the static library:

- The number of lines, and the number of words on each line (we call this the *skeleton* of the poem).

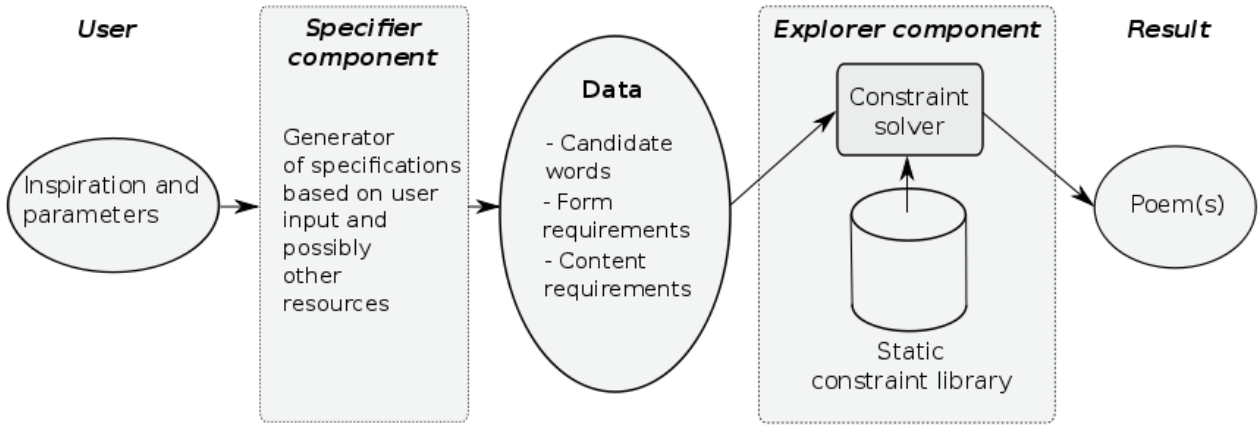


Figure 1: Overview of the poetry composition workflow. The user provides some inspiration and parameters, based on which the space specifier component generates a set of constraints, used as "data" by the constraint solver in the explorer component. The explorer component additionally contains a static library of constraints that are dynamically triggered by the data. Explorer component then outputs a poem that best fulfills wishes of the user.

- For each word position in the skeleton, a list of words that potentially can be used in the position (collectively called the *candidates*).
- Possible additional requirements on the desired form of the poem (e.g., rhyming structure).
- Possible additional requirements on the syntax and contents of the poem (e.g., interdependencies between words to make valid expressions).

We will next describe these in more detail.

Poetry Composition via Answer Set Programming

The explorer component takes as input specifications dynamically generated by the specifier, affecting both the search space and the aesthetic. In addition, it uses a static constraint library. Together, the dynamic specifications and the constraint library form a constraint satisfaction problem (or, by extension, an optimization problem; see end of the section). The constraint satisfaction problem is built so that the solutions to the problem are in one-to-one correspondence with the poems that satisfy the requirements imposed by the specifier component of the system (as potentially instructed by the user). Highly optimized off-the-shelf constraint satisfaction solvers can then be used to find the solutions, i.e., to produce poems.

In this work, we employ *answer set programming* (ASP) (Gelfond and Lifschitz 1988; Niemelä 1999; Simons, Niemelä, and Soininen 2002) as the constraint programming paradigm, since ASP allows for expressing the poem construction task in an intuitively appealing way. At the same time, state-of-the-art ASP solvers, such as Clasp (Gebser, Kaufmann, and Schaub 2012), provide an efficient way of finding solutions to the poem construction task. Furthermore, ASP offers in-built support for constraint optimization, which allows for searching for a poem of high quality with respect to different imposed quality measures.

We will not provide formal details on answer set programming and its underlying semantics; the interested reader is referred to other sources (Gelfond and Lifschitz 1988; Niemelä 1999; Simons, Niemelä, and Soininen 2002) for a detailed account. Instead, we will in the following provide a step-by-step intuitive explanation on how the task of poetry generation can be expressed in the language of ASP. For more hands-on examples on how to express different computational problems in ASP, we refer the interested reader to Gebser et al. (2008).

Answer set programming can be viewed as a data-centric constraint satisfaction paradigm, in which the input data, represented via *predicates*, expresses the problem instance. In our case, this dynamically generated data will express, for example, basic information on the poem skeleton (such as length of lines), and the candidate words within the input vocabulary that can be used in different positions within the poem. The actual computational problem (in our case poetry generation) is expressed via *rule-based constraints* which are used for inferring additional knowledge based on the input data, as well as for imposing constraints over the solutions of interest. The rule-based constraints constitute the static constraint library: once written, they can be reused in any instances of poem generators just by generating data that activates the constraints. Elementary constraints are an integral part of the system — comparable to program code. More rule-based constraints can be added by the specifier component if needed. The end-user does not need to write any constraints.

The Basic Model

We next describe a constraint library, starting with elementary constraints. We also illustrate dynamically generated specifications. While these are already sufficient to generated poetry comparable to that of Toivanen et al. (2012), we remind the reader that these constraints are examples illustrating the flexibility of constraint programming in compu-

Table 1: The predicates used in the basic ASP model

Predicate	Interpretation
`rows(X)`	the poem has `X` rows
`positions(X,Y)`	the poem contains `Y` words on row `X`
`candidate(W,I,J,S)`	the word `W`, containing `S` syllables, is a candidate for the `J`th word of the `I`th line
`word(W,I,J,S)`	the word `W`, containing `S` syllables, is at position `J` on row `I` in the generated poem

```
% Generator part
{ word(W,I,J,S) } :- candidate(W,I,J,S).                                    (G1)

% Testing part: the constraints
:- not 1 { word(W,I,J,S) } 1, rows(X), I = 1..X, positions(I,Y), J=1..Y. (T1)
```

Figure 2: Answer set program for generating poetry: the basic model

tational poetry composition, and different sets of constraints can be used for different effects.

We will first give a two-line basic model of the constraint library that takes the skeleton and candidates as input. This model simply states that exactly one of the given candidate words must be selected for each word position of the poem.

Predicates The predicates used in the basic answer set program are listed in Table 1, together with their intuitive interpretations.

The input predicates `rows/1` and `positions/2` characterize the number of rows and the number of words allowed on the individual rows of the generated poems. The input predicate `candidate/4` represents the input vocabulary, i.e., the words that may be considered as candidates for words at specific positions.

The output predicate `word/4` represents the solutions to the answer set program, i.e., the individual words and their positions in the generated poem.

Example. The following is an example of the basic structure of a data file representing a possible input to the basic ASP model

```
rows(6).
positions(1,6).
positions(2,8).
positions(3,8).
positions(4,5).
positions(5,6).
positions(6,6).

candidate("I",1,1,1).
candidate("melt",1,2,1).
candidate("weed",1,2,1).
candidate("teem",1,2,1).
candidate("kidnap",1,2,2).
candidate("perspire",1,2,2).
candidate("shut",1,2,1).
candidate("eclipse",1,2,1).
candidate("sea",1,2,1).
candidate("plan",1,2,1).
candidate("hang",1,2,1).
candidate("police",1,2,2).
candidate("revamp",1,2,2).
candidate("flip",1,2,1).
candidate("wring",1,2,1).
candidate("sting",2,2,2).

...
```

Rules The answer set program that serves as our basic model for generating poetry is shown in Figure 2. The program can be viewed in two parts: the *generator part* (Rule G1) and the *testing part* (Rule T1). The test part consists of rule-based constraints that filter out poems that do not satisfy the conditions for acceptable poems characterized by the program.

In the generator part, Rule G1 states that each candidate word for a specific position of the poem may be considered to be chosen as the word at that position in the generated poem (expressed using the so-called *choice* construct `{ word(W,I,J,S) }`).

In the testing part, Rule T1 imposes the most fundamental constraint that exactly one candidate word should be chosen for each word position in the poem: the empty left-hand-side of the rule is interpreted as *falsum*, a contradiction. The rule then states that, for each row and each position on the row, it is a contradiction if it is *not* the case that exactly one word is chosen for that position (expressed as the *cardinality* construct `1 { word(W,I,J,S) } 1`).

Example. Given the data presented above these basic rules are now grounded as follows. There are six lines in the poem as described by the *rows* predicate and each of these lines has a certain number of positions to be filled with words as described by the *positions* predicate. The *candidate* predicates specify which words are suitable choices for these positions. During grounding the solver tries to find a suitable candidate for each position, which is trivial in the basic model that lacks any constraints between the words. We consider more interesting models next.

Controlling the Form of Poems

We will now describe examples of how the form of the poems being generated can be further controlled in a modular fashion by introducing additional predicates and rules over these predicates to the basic ASP model. The additional predicates introduced for these examples are listed in

Table 2: Predicates used in extending the basic ASP model

Predicate	Interpretation
must_rhyme(I,J,K,L)	the word at position J on row I and the word at position L on row K are required to rhyme
rhymes(X,Y)	the words X and Y rhyme
nof_syllables(I,C)	the Ith row of the poem is required to contain C syllables
min_occ(W,L)	L is the lower bound on the number of occurrence of the word W
max_occ(W,U)	U is the upper bound on the number of occurrence of the word W

```
% Generator part

{ word(W,I,J,S) } :- candidate(W,I,J,S).                                    (G1)
rhymes(Y,X) :- rhymes(X,Y).                                                 (G2)
syllables(W,S) :- candidate(W,_,_,S).                                       (G3)

% Testing part: the constraints

:- not 1 { word(W,I,J,S) } 1, rows(X), I = 1..X, positions(I,Y), J=1..Y.   (T1)
:- word(W,I,J,S), word(V,K,L,Q), must_rhyme(I,J,K,L), not rhymes(W,V).     (T2)
:- Sum = #sum [ word(W,I,J,S) = S ], Sum != C, nof_syllables(I,C),         (T3)
   I = 1..X, rows(X).
:- not L { word(W,_,_,_) } U, min_occ(W,L), max_occ(W,U).                  (T4)
```

Figure 3: Answer set program for generating poetry: extending the basic model

Table 2. Using these predicates, rules that refine the basic model are shown in Figure 3 (Rules G2, G3, and T2–T4).

Rhyming The predicate `must_rhyme/4` is used for providing pairwise word positions that should rhyme. Knowledge on the pairwise relations of the candidate words, namely, which pairs of candidate words rhyme, is provided via the `rhymes/2` predicate. Rule G2 enforces that rhyming of two words is a symmetry relation. In the testing part Rule T2 imposes the constraint that, in case two words chosen for specific positions in a poem must rhyme, but the chosen two words do not rhyme, a contradiction is reached.

Numbers of Syllables The basic model can also be extended to generate poetical structures with more specific constraints. As an example, one can consider forms of poetry that have strict constraints on the numbers of syllables in every line, such as haikus, tankas, and sonnets.

We use the additional predicate `nof_syllables/2` for providing as input the required number of syllables on the individual rows. At the same time, Rule G3 projects the information on the number of syllables of each candidate word to the `syllables/2` predicate. Rule T3 can then be used to ensure that the number of syllables on each row (line) of the poem (computed and stored in the `Sum` variables using the counting construct `Sum = #sum [ word(W,I,J,S) = S ]`) matches the number of syllables specified for the row by the `nof_syllables/2` predicate.

Word Occurrences The simple model above does not control possible repetitions of words at all. Such control can be easily added by introducing input predicates `min_occ(W,L)` and `max_occ(W,U)`, which are then used to state for each word `W` the minimum `L` (respectively, maximum `U`) number of occurrences allowed for the word. Using these additional predicates, Rule T4 then constrains the number of occurrences to be within these lower and upper bounds (expressed by the cardinality constraint `L { word(W,_,_,_) } U`).

Further Possibilities for Controlling Form The possibilities of controlling poetical forms are not of course limited to simple requirements for fulfilment of certain syllable structures or rules for rhyming and alliteration. Besides strict constraints on numbers of syllables on verse, classical forms of poetry usually obey a specific stress pattern, as well. Stress can be handled with constraints similar to the ones governing syllables. Metric feet like iamb, anapest, and trochee can be used by specifying constraints that describe positions where the syllable stress must lie in every line of verse.

Controlling poetical form also provides interesting possibilities for using constraint optimization techniques (to be described below). As an example, consider different forms of lipograms i.e. poems that avoid a particular letter like *e* or univocal poems where the set of possible vowels in the poem is restricted to only one vowel. Similarly, more complex optimisations of the speech sound structure can be handled depending on whether the wished poetry is required to have soft or brutal sound, or to have characteristics of a tongue-twister.

Controlling the Contents and Syntax of Poems

While the example constraints presented above focus on controlling the form of poems, linguistic knowledge of phonology, morphology, and syntax (as examples) can similarly be controlled by introducing additional constraints in a modular fashion. This includes rules of syntax that specify

```
failed_rhyme(I,J,K,L) :- word(W,I,J,S), word(V,K,L,Q),
                         must_rhyme(I,J,K,L), not rhymes(W,V).              (T2')
failed_syllable_count(I) :- Sum = #sum [ word(W,I,J,S) = S ], Sum != C,
                            nof_syllables(I,C), I = 1..X, rows(X).          (T3')
failed_occount(W) :- not L { word(W,_,_,_) } U, min_occ(W,L), max_occ(W,U). (T4')

#minimize [ failed_rhyme(I,J,K,L) @3 ].                                     (O2)
#minimize [ failed_syllable_count(I) : I=1..X : rows(X) @2 ].               (O3)
#minimize [ failed_occount(W) @1 ].                                        (O4)
```

Figure 4: Handling inconsistencies by relaxing the constraints and introducing optimization criteria

how linguistic elements are sequenced to form valid statements and rules of semantics which specify how valid references are made to concepts.

Consider, for example, transitive and intransitive verbs, i.e., verbs that either require or do not require an object to be present in the same sentence. Here one can impose additional constraints for declaring which words can or cannot be used in the same sentence where a transitive verb requiring certain preposition and an object has been used. Similarly other constraints not directly related to the poetical forms but rather to linguistic structures like idioms, where several words are always bundled together, can be effectively declared as constraints. The same holds for syntactic aspects such as rules governing the constituent structure of sentences (Lierler and Schüller 2012).

As a simple, more concrete example, consider the following. In order to declare that the poems of interest start with the word "I", the fact `word("I",1,1,1).` can be added to the constraint model. In order to ensure that all verbs associated with the first person should be in past tense, the additional predicate `in_past_tense/1` can be introduced, and specified for each past-tense verb in the data. Combining the above, one can as an example declare that the word following any "I" is in a past tense, using the following two rules.

```
:- word("I",I,J,1), word(W,I,J+1,_),
  not in_past_tense(W).

:- word("I",I,J,1), positions(I,J),
  word(W,I+1,1,_), not in_past_tense(W).
```

Here the first rule handles the case that the occurrence of "I" is not the last word on a row. The second rule handles the case that "I" is the last word on a row, in which case the first word on the following row should be in past tense.

More generally, one can pose constraints that ensure that two (or more) words within a poem are compatible (in some specified sense), even if the words are not next to each other. For an example, consider the additional predicated `pronoun/1` and `verb/1` that hold for words that are pronouns and verbs, respectively, and the predicate `person/2` that specifies the grammatical person, expressed as an integer value, of a given word: `person(W,P)` is true if and only if the word `W` has person `P`. Using these predicates, one can enforce that, for the first verb following any pronoun (not necessarily immediately after the pronoun), the pronoun and the verb have to have the same person. For instance, after the pronoun "she" the first following verb has to be in the third person singular form. This can be expressed as the following rule:

```
:- word(W,I,J,_), pronoun(W), person(W,P),
  0{ word(U,I,L,_) : verb(U) : L>J : L<K }0,
  word(V,I,K,_), verb(V), person(V,Q),
  K>J, P!=Q.
```

Similarly, by specifying the additional predicate `verb/1` for each verb in the input data, one can require that the whole poem should be in past tense:

```
:- word(W,_,_,_), verb(W),
  not in_past_tense(W).
```

Specifying an Aesthetic via Optimization

Up to now, we have only considered hard constraints, and did not address how to assess the aesthetics of generated poems, or how to generate poems that are maximally aesthetic by some measures.

In the constraint programming framework, an aesthetic can be specified using *soft constraints*. The constraint solver then attempts to look for poems which maximally satisfy the soft constraints. In ASP, this is achieved by using *optimization* statements offered by the language.

As concrete examples, we will now explain how Rules T2–T4 can be turned into soft constraints. The soft variants, Rules T2'–T4', are shown in Fig. 4, together with the associated optimization statements O2–O4. Taking Rule T3 as an example, the idea is to introduce a new predicate `failed_syllable_count/1` with the following interpretation: Predicate `failed_syllable_count(I)` is true for row `I` if and only if the number of syllables on the row was not made to match the required number. In contrast to Rule T3, which rules out all solutions of the model immediately in such a case, Rule T3' simply results in assigning `failed_syllable_count(I)` to true. Thus the predicate `failed_syllable_count/1` acts as an indicator of failing to have the required number of syllables on a specific row.

The optimization statement associated with Rule T3' is Rule O3. This `minimize` statement declares that the number of rows `I` for which `failed_syllable_count(I)` is assigned to true should be minimized, or equivalently, that the numbers of syllables should conform to the required numbers of syllables for as many rows as possible. The optimization variants T2' and T4' and the associated optimization statements follow a similar scheme.

When multiple such optimization statements are introduced to the model, the relative importance of the statements is declared using the `@i` attached to each of the optimization statement. In the example of Figure 4, the primary objective is to minimize the number of rhyming failures (specified using `@3`). The secondary objective is then to find, among the set of poems that minimize this primary objective, a poem that has a minimal number of lines with a wrong number of syllables, (using `@2`), and so forth.

Examples

We will now illustrate the results and effects of some combinations of constraints.

In the data generation phase (the specifier component) we use the methodology by Toivanen et al. (2012), including the Stanford POS-tagger and morpha & morphg inflectional morphological analysis and generation tools (Toutanova et al. 2003; Minnen, Carroll, and Pearce 2001). The poem templates are extracted automatically from a corpus of human-written poetry. The only input by the user is a topic for the poem, and some other parameters as described below.

As a test case for our current system we study how the approach manages to produce different types of *quatrains*. It is a unit of four lines of poetry; it may either stand alone or be used as a single stanza within a larger poem. The quatrain is the most common type of stanza found in traditional English poetry, and as such is fertile ground on which to test theories of the rules governing poetry patterns.

The specifier component randomly picks a quatrain from a given corpus of existing poetry. It then automatically analyses its structure, to generate a skeleton for a new poem. The following poem skeleton is marked with the required part-of-speech for every word position (PR = pronoun, VB = verb, PR_PS = possessive pronoun, ADJ = adjective, N_SG = singular noun, N_PL = plural noun, C = conjunction, ADV = adverb, DT = determiner, PRE = preposition):

N_SG VB, N_SG VB, N_SG VB!
PR_PS ADJ N_PL ADJ PRE PR_PS N_SG:
– C ADV, ADV ADV DT N_SG PR VB!
DT N_SG PRE DT N_PL PRE N_SG!

The specifier component then generates a list of candidate words for each position. If we give "music" as the topic of the poem, the specifier specifically uses words related to music as candidates, where possible (Toivanen et al. 2012). A large number of poems are possible, in the absense of other constraints, and the constraint solver in the explorer component outputs this one (or any number of alternative ones, if required):

Music swells, accent practises, traditionalism marches!
Her devote narrations bent in her improvisation:
– And then, vivaciously directly a universe
she ventilates!
An anthem in the seasons of radio!

This example does not yet have any specific requirements for the prosodical form. Traditional poetry often has its prosodic structure advertised by one or more of several poetic devices, with rhyming and alliteration being best-known of these. Let the specifier component hence generate the additional constraints that the first and the third line must rhyme, as well as the second and fourth line. As a result of this more constrained specification we now get a very similar poem, but with some words changed to rhyme.

Music swells, accent practises, traditionalism hears!
Her devote narrations bent in her chord:
– And then, vivaciously directly a universe
she disappears!
An anthem in the seasons of record!

Addition of this simple constraint adds rhyme to the poem, which in turn draws attention to the prosodic structure of the poem. Use of prosodic techniques to advertise the poetical nature of a given text can also enhance coherence of the poetry as the elements are linked together more tightly. For example, a rhyme scheme of ABAB would give the listener a strong sense that the first and third as well as the second and fourth lines belong together as a group, heightening the saliency of the alternating structure that may be present in the content, as well.

The constraint on rhyming reflects the intuition that rhyme works by creating expectation and satisfaction of that expectation. Upon hearing one line of verse, the listener expects to hear another line that rhymes with it. Once the second rhyme is heard, the expectation is fulfilled, and a sense of closure is achieved. Similarly, adding constraints that specify a more sophisticated prosodic structure or content related aspects may lead to improved quality of the generated poetry.

Let us conclude this section with an example of an aesthetic, an optimization task concerning the prosodic structure of poetry. Consider composition of lipograms, i.e., poems avoiding a particular letter. (Also univocalism or more complex optimizations of the occurrence of certain speech sounds can be composed in a similar fashion.) The following poem is an example of a lipogram that avoids the letter *o*. As a result of this all words that contained that letter in the previous example are changed to match the strengthened constraints:

Music swells, accent practises, theatre hears!
Her delighted epiphanies bent in her universe:
– And then, singing directly a universe she disappears!
An anthem in the judgements after verse!

Empirical results of Toivanen et al. (2012) indicate that in Finnish, already the basic mechanism produces poems of surprisingly high quality. The sequence of poems above illustrates how their quality can be substantially improved by relatively simple addition of new, declarative constraints.

Discussion and Conclusions

We have proposed harnessing constraint programming for composing poetry automatically and flexibly in different styles and forms. We believe constraint programming has high potential in describing also other creative phenomena.

A key benefit is the declarativity of this approach: the conceptual space is explicitly specified, and so is the aesthetic, and both are decoupled from the algorithm for exploring the search space (an off-the-shelf constraint solver). Due to its modular nature, the presented approach can be an effective building block of more sophisticated poetry generation systems.

An interesting next step for this work is to build an interactive poetry composition system which makes use of constraint programming in an iterative way. In this approach the constraint model is refined and re-solved based on user feedback. This can be seen as an iterative abstract-refinement process, in which the first abstraction specifies a very large search-space that is iteratively pruned by refining the constraint model with more intricate rules that focus search to the most interesting parts of the conceptual space.

Another promising research direction is to consider a self-reflective creative system. Since the search space and aesthetic are expressed in an explicit manner as constraints, they can also be observed and manipulated. We can envision a creative system that controls its own constraints. For instance, after observing that a large amount of good results is obtained with the current constraints, it may decide to add new constraints to manipulate its own internal objectives. Modification of the set of constraints may lead to different conceptual spaces and eventually to transformational creativity (Boden 1992). Development of metaheuristics and learning mechanisms that enable such self-supported behavior is a great challenge indeed.

Acknowledgements

This work has been supported by the Academy of Finland under grants 118653 (JT,HT), and 132812 and 251170 (MJ).

References

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Boenn, G.; Brain, M.; vos, M. D.; and Ffitch, J. 2011. Automatic music composition using answer set programming. *Theory and Practice of Logic Programming* 11(2-3):397–427.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *International Conference on Computational Creativity*, 95–102.

Diaz-Agudo, B.; Gervás, P.; and González-Calero, P. A. 2002. Poetry generation in COLIBRI. In *ECCBR 2002, Advances in Case Based Reasoning*, 73–102.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Ostrowski, M.; Schaub, T.; and Thiele, S. 2008. A user's guide to gringo, clasp, clingo, and iclingo. `http://downloads.sourceforge.net/potassco/guide.pdf?use_mirror=`.

Gebser, M.; Kaufmann, B.; and Schaub, T. 2012. Conflict-driven answer set solving: From theory to practice. *Artificial Intelligence* 187:52–89.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium*, 1070–1080.

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. *Journal of Knowledge-Based Systems* 14(3–4):181–188.

Langkilde, I., and Knight, K. 1998. The practical value of n-grams in generation. In *Proceedings of the International Natural Language Generation Workshop*, 248–255.

Lierler, Y., and Schüller, P. 2012. Parsing combinatory categorial grammar via planning in answer set programming. In Erdem, E.; Lee, J.; Lierler, Y.; and Pearce, D., eds., *Correct Reasoning*, volume 7265 of *Lecture Notes in Computer Science*, 436–453. Springer.

Manurung, H. M.; Ritchie, G.; and Thompson, H. 2000. Towards a computational model of poetry generation. In *Proceedings of AISB Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, 79–86.

Manurung, H. 2003. *An evolutionary algorithm approach to poetry generation*. Ph.D. Dissertation, University of Edinburgh, Edinburgh, United Kingdom.

Minnen, G.; Carroll, J.; and Pearce, D. 2001. Applied morphological processing of English. *Natural Language Engineering* 7(3):207–223.

Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku : Generating haiku with word associations norms. In *Proceedings of NAACL Workshop on Computational Approaches to Linguistic Creativity*, 32–39.

Niemelä, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence* 25(3-4):241–273.

Roads, C. 1980. Interview with Marvin Minsky. *Computer Music Journal* 4.

Simons, P.; Niemelä, I.; and Soininen, T. 2002. Extending and implementing the stable model semantics. *Artificial Intelligence* 138(1-2):181–234.

Stravinsky, I. 1947. *Poetics of Music*. Cambridge, MA: Harvard University Press.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *International Conference on Computational Creativity*, 175–179.

Toutanova, K.; Klein, D.; Manning, C.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 252–259.

Wong, M. T., and Chun, A. H. W. 2008. Automatic haiku generation using VSM. In *Proceedings of ACACOS, The 7th WSEAS International Conference on Applied Computer and Applied Computational Science*, 318–323.

Slant: A Blackboard System to Generate Plot, Figuration, and Narrative Discourse Aspects of Stories

Nick Montfort
The Trope Tank, MIT
77 Mass Ave, 14N-233
Cambridge, MA 02139 USA
nickm@nickm.com

Rafael Pérez y Pérez
División de Ciencias de la Comunicación y Diseño
Universidad Autónoma Metropolitana, Cuajimalpa,
México D. F.
rperez@correo.cua.uam.mx

D. Fox Harrell
Imagination, Computation, & Expression Laboratory, MIT
77 Mass Ave, 14N-207
Cambridge, MA 02139 USA
fox.harrell@mit.edu

Andrew Campana
Department of East Asian Languages & Civilizations
Harvard University
Cambridge, MA 02138
USA
campana@fas.harvard.edu

Abstract

We introduce Slant, a system that integrates more than a decade of research into computational creativity, and specifically story generation, by connecting subsystems that deal with plot, figuration, and the narrative discourse using a blackboard. The process of integrating these systems highlights differences in the representation of story and has led to a better understanding of how story can be usefully abstracted. The plot generator MEXICA and a component of Curveship are used with little modification in Slant, while the figuration subsystem Fig-S and the template generator GRIOT-Gen, inspired by GRIOT, are also components. The development of the new subsystem Verso, which deals with genre, shows how different genres can be computationally modeled and applied to in-development stories to generate results that are surprising in terms of their connections and valuable in terms of their relationship to cultural questions. Example stories are discussed, as is the potential of the system to allow for broader collaboration, the empirical testing of how subsystems interrelate, and possible contributions in literary and artistic contexts.

Introduction

Slant is a system for creative story generation that integrates different types of expertise and creativity; the framework it provides also means that other systems, implementing other approaches to story generation, can be integrated into it in the future. The development of Slant has involved formalizing, reworking, and testing ideas about creative storytelling and what is important to writing stories—specifically, the poetics of figuration, the poetics of plot development, and the poetics of narrating. The system incorporates a new perspective on genre and integrates components from three existing systems: D. Fox Harrell's GRIOT, Rafael Pérez y Pérez's MEXICA, and Nick Montfort's Curveship.

Story generation systems have not yet used an architecture of this sort to encapsulate different expertise and different aspects of creativity; nor have they incorporated major components that are based on existing, proven systems by different researchers.

Slant is a blackboard system in which different subsystems, each of them informed by and modeling humanistic theories, collaborate together, working incrementally to fully specify a story. An alternative, simpler process involves making decisions in a "pipeline," in which one system offers, for instance, a plot and another system determines how the narrative discourse will be arranged. Although this system seems to be a poor model of human creativity, it is a reasonable first step toward a "blackboard" system. Two of the Slant collaborators previously developed such a pipelined system with two stages (Montfort and Pérez y Pérez 2008). The current project involves five major subsystems rather than two and uses a blackboard architecture, allowing any of the subsystems that work during the main phase of generation to augment the story representation at any point.

The generation of stories in Slant begins with minimal, partial proposals from a simple unit, the Seeder. In turn, the subsystems MEXICA, Verso, and Fig-S read and add to this set of proposals, each according to its focus. When the proposals are complete, the finished story specification is sent to GRIOT-Gen so conceptual blending can be applied to the relevant templates and then to the three-stage pipelined text generation component of Curveship. Curveship-Gen realizes a finished story in the form of a text file that can be read and considered by human readers.

This paper introduces the architecture of the system and describes the subsystems that build and realize stories together. It includes a discussion of what was learned by inte-

grating three different lines of research on story generation. Reflections are also offered on the first set of stories produced by the system, and some discussion of the potential of the system is included as well. Slant will undergo more refinement and development, but the work that has been done so far is of relevance to those working to implement large-scale computational creativity systems that integrate heterogeneous subsystems, to those developing representations of story and other creative representations, and to those working specifically in story generation.

Creativity and the Architecture of Slant

Boden holds that creativity involves the production of new, surprising, and valuable results (Boden 2004). In the case of story generation and other literary endeavors, being new involves not repeating what has been done before (by the system or in the wider culture); surprise often manifests itself in unusual juxtapositions that are effective, though one would not have guessed it; and value, rather than indicating that the story is of didactic or economic value, means that a story accomplishes some imaginative or poetic purpose—it connects in some way to cultural or psychological issues or questions and allows the reader to think about them in new ways. Stories that surprise readers by bringing unusual elements together and which provide for this sort of reflection, but which do so in the same way as existing stories, are not new. Stories that are innovative and could allow for reflection, but which do not involve unusual juxtapositions or connections, are not surprising. Stories that are fresh and involve unusual combinations of elements, but do not ultimately seem to have a point of any sort, are not of value.

Taking value to indicate relevance within culture means that the value of a story is similar to what has been called, with regard to conversational stories of the sort that are uttered all the time by people, its "point" (Polanyi 1989). While the point of a story is understood in the context of a specific conversation, the ability of a story to have a point at all can be understood within the context of culture. Valuable stories are those that have a point to at least some readers when they encounter them in some context.

Beyond Boden's three components of creativity, we also consider a higher level of creativity. Namely, the various cognitive processes for conceptualization that enable people to recognize and generate new, surprising, and valuable cultural content are forms of everyday creativity. Cognitive scientist Gilles Fauconnier has referred to these process of meaning construction as "backstage cognition" and asserts that backstage cognition includes specific phenomena such as "viewpoints and reference points, figure-ground/profile-bases/landmark-trajector organization, metaphorical, analogical, and other mappings, idealized models, framing, construal, mental spaces, counterpart connections, roles, prototypes, metonymy, polysemy, conceptual blending, fictive motion, [and] force dynamics" (Fauconnier 1999). These cognitive processes are especially important to note here because the notion of creativity informing Fig-S and GRIOT-Gen is based on a model of the creative backstage cognition phenomenon of metaphorical mapping, most prominently, but also mental spaces, counterpart connections, metaphor, analogy, and metonymy in the case of the GRIOT system that inspired them.

To succeed repeatedly and reliably at creativity, a storytelling system must have mechanisms relevant to each of these aspects of creativity. It must have some model of what has happened before to prompt novelty, somehow provide for stories that join aspects together in unusual and effective ways, and somehow provide for stories that relate to culture and have a point. The means of accomplishing these aspects of creativity do not have to be abstracted into separate components of a system, but they do need to be somehow realized by a creative system.

A simple way that systems can connect and to some extent collaborate involves organizing them in a pipeline. This can model a regimented assembly-line process or "waterfall" model in which each subsystem participates in one phase and interfaces only with the systems before and after it. For certain processes, this may be adequate, but for the nuanced process of creativity, which involves making interesting connections, the components of a system probably need to interact in a less constrained and unidirectional manner. This was the rationale for the blackboard architecture used in Slant.

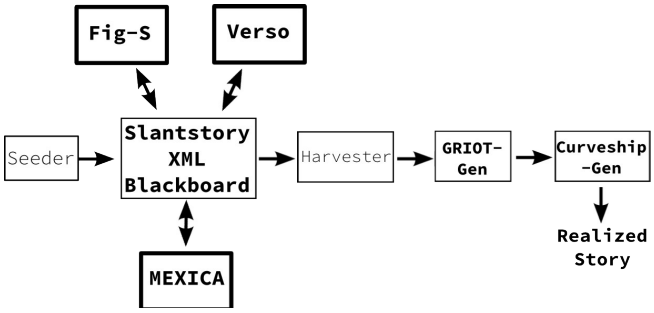


Figure 1: The architecture of Slant.

The Blackboard and Subsystems

In Slant, the three major story-building subsystems can write to and read from a blackboard representation of the story in progress. Currently, the systems function in practice much as a pipeline does, with each of the three subsystems augmenting the story representation once. The systems can influence each other "backwards" only via Verso examining the current plot and proposing a new action (not just a specification of narrative discourse, which is always proposed.) MEXICA can then incorporate that expanded plot into its next ER cycle that it uses to elaborate the plot. Although the interactions between subsystems are not intricate at this

point, the framework is in place for more elaborate blackboard interaction in future versions of Slant.

Currently, MEXICA contributes an initial, partial plot – a minimal, random one will eventually be provided at the first step by the Seeder. Then, Verso assigns a genre and a specification of the narrative discourse, and MEXCIA further elaborates the plot until it is complete. Verso may specify constraints on how the story is to be developed. For instance, it may specify that a particular character, who has been designated as the narrator of the story, should not die. MEXICA will respect these in elaborating the story. Finally, Fig-S determines what figuration will be used. Eventually, another system, the Harvester, will check to see if all aspects of the story are complete, allowing the subsystems to augment the story in several different orders. After the story representation is complete, it is realized. GRIOT-Gen determines how to realize figurative representations and Curveship-Gen does content selection, microplanning, and surface realization to produce the final text.

The MEXICA subsystem has the most explicit model of an aspect of creativity; it explicitly evaluates the novelty and interestingness of the component of story that it develops, the plot. Verso and Fig-S both aim to add surprise by combining conventional genres and metaphors in unusual ways. They do not currently measure how surprising their results are, but they embody techniques for choosing appropriate combinations that may be seen as creative by readers.

Foundational Systems

MEXICA. This system generates plots or frameworks for short stories about the Mexicas, the old inhabitants of what today is México city, also known as the Aztecs. MEXICA's process is based on the engagement/reflection cycle, a cognitive account of writing by Mike Sharples (Pérez y Pérez and Sharples 1999, 2001, 2004). During engagement the system focuses on generating sequences of actions driven by content and rhetorical constraints and avoids the use of explicit goals or predefined story-structures. During reflection MEXICA evaluates the novelty and interestingness of the material produced so far and verifies the coherence of the story (see also Pérez y Pérez et al. 2011).

The design of the system is based on structures known as Linguistic Representations of Actions (LIRAs), which are sets of actions that any character can perform in the story and whose consequences produce some change in the story-world context. There are two types of possible pre-conditions and postconditions in MEXICA: emotional links between characters and dramatic tensions in the story.

MEXICA is incorporated as the generator of plot. It generates plot in stages, allowing other systems to interact with the story representation as it does so. In the current system, it can be influenced by actions added to the story by Verso.

GRIOT. This is a system that is the basis for interactive and generative text and multimedia works using Harrell's Alloy algorithm for conceptual blending. These works include poetic, animated, and documentary systems that themselves produce different output each time they are run. While GRIOT allows authors to implement narrative and poetic structures (e.g., plots), a major contribution of GRIOT is its orientation toward the dynamic generation of content resulting from modeling aspects of figurative thought that can be described formally. That is, GRIOT allows authors to fix elements such as narrative structure while varying output in terms of theme, metaphor, emotional tone, and related types of what is here called "figuration" (results of figurative thought).

Rather than being based on a single knowledge base or ontology, as is the case with many classic AI systems, GRIOT creates blends between different ontologies (Harrell 2006, 2007). Indeed, a key feature of GRIOT is the ability of authors to construct subjective ontologies based in specific authorial worldviews, elements of which are then blended in a manner that maintains coherence based on several formal optimality principles inspired by a subset of those proposed by Gilles Fauconnier and Mark Turner (1999). This approach allows for novel, surprising, and valuable content to be generated that retains conceptual coherence. GRIOT, like MEXICA, has also been used to implement cultural forms of narrative that are not often privileged in computer science, in this case oral traditions of narrative from the African diaspora (Harrell 2007a). This is important because some forms of oral narrative have more in common with narratives in virtual worlds than the graphocentric (text-biased) forms of narrative privileged in most research in the field of narratology in literary studies.

The implemented GRIOT system, and experience with it, have informed the development of Fig-S, a component of Slant that proposes what types of figuration, mainly metaphor, will be used in telling the story. GRIOT also inspires GRIOT-Gen, the component that generates natural language representations for figuratively enriched versions of particular actions after the story representation is completely developed (see also Goguen and Harrell 2008).

Curveship. This is an interactive fiction system that provides a world model (of characters, objects, locations, and things that happen) while also modeling the narrative discourse, so that the narration and description of the simulated world can change (Montfort 2009, 2011). Curveship can tell events out of order, using flashback and other techniques, and can tell the story from the standpoint of particular characters and their perceptions and understandings. It is based on Genette's theories (Genette 1983) and incorporates other ideas from narratology. The architecture of Curveship draws on well-established techniques for simulating an IF world, separating these from the subsystem for narrating, which in-

cludes a standard three-stage natural language generation pipeline. To make use of the system, either for interactive fiction authoring or story generation, one specifies high-level narrative aspects; the system does appropriate content selection, works out grammatical specifics, and realizes the text with, for instance, proper verb formation.

Some world simulation abilities and the narrative text generation capabilities of Curveship are used directly in Slant in Curveship-Gen, the component that outputs the finished, realized story.

The Slantstory XML Format

Connecting different systems so that they can work together means establishing shared representations. For Slant, that representation is an XML format called Slantstory. It contain all of the information that is needed in the final steps to represent each action and realize the story, meaning that it must contain sufficiently granular information about the plot, the narrative discourse, and the types of conceptual blending that are to be done. This information is not only needed at the last stage, where the generation of text is done. It can also be read by the different subsystems during story generation, when the story is not yet complete, and can influence the next stage of story augmentation. Because of this, Slantstory is a format not only for representing entire, complete stories but also for representing partial stories, the composition of which is in progress. In the current implementation, subsystems can augment a story and declare it complete, but cannot revise or remove what has already been contributed.

To declare a common representation for (both partial and complete) stories, an agreement had to be reached between different perspectives on what the elements of a story are, what is to be represented about each, and how granular the representation of each element is. The Slantstory DTD specifies five elements that occur within the root:

```
<!ELEMENT slantstory
(existents, actions, spin?, genre?, figuration?)>
```

A story cannot be complete without all five of these present, but only existents and actions are required at every stage of story development. The existents are of three types: locations, characters, and things. Actions each have a verb (which might be a phrase such as "try to flee") and may have any or all of agent, direct object, and indirect object specified. The "instantaneous" tension level, or change in the tension associated with an action, is also represented there. The actions also have a unique ID number which indicates their chronological order in the story world, as in:

```
<action verb="cure" agent="virgin" direct="enemy"
indirect="curative plant" location="Texcoco Lake"
tension="0" id="42" />
```

One challenge in developing and using this blackboard representation involves the different models of existents and actions that the three foundational systems use. Characters and locations, but nothing like props or "things," are represented in MEXICA, while Curveship represents all three sorts of existents to provide the type of simulation that is typical in interactive fiction, where objects can typically be acquired, given to other characters, placed on surfaces and in containers, and so on. MEXICA was modified for use in Slant to produce appropriate representations of whatever things were mentioned in actions.

The representation of action was also not consistent between the foundational systems. Curveship has a typology of four actions: Configure (move some existent into, onto, out of, off, or to a different location), Modify (change the state of some existent), Sense (gain information about the world from sensing), and Behave (any other action, not resulting in any change of state in the world). Although they may be quite different, all actions are meant to correspond to a sentence with a single verb phrase when realized. MEXICA's actions, on the other hand, are not categorized in this way and include many different sorts of representations. There are, for instance, complex actions such as FAKED_STAB_INSTEAD_HURT_HIMSELF, indications that an action was not taken such as NOT_CURE, and indications that a state is to be described at a certain point such as WAS_BROTHER_OF.

The first of these issues, the granularity of action, was handled by developing a mapping between MEXICA actions and Slantstory actions. A limitation of this approach is that actions cannot be inserted into the middle of a series of Slantstory actions that correspond to a single MEXICA action; this is enforced by giving the actions consecutive IDs, so that there is no room to add further actions. Ideally, however, other subsystems would be able to modify the Slantstory representation of actions in any way. The second of these issues bring up the interesting issue of disnarration (Prince 1988), that it is possible in a story to not only tell what has happened but to also tell what what did not happen, and that doing so can have an interesting effect on the reader. Disnarration is not the representation of action, however, so it cannot be represented in a straightforward way in a list of actions, and should be handled elsewhere—in the spin element, for instance. Resolving the final issue related to stative information also requires further work, since the system should both represent facts about the story world (probably in the existents element) and when to mention them (probably in the spin element).

GRIOT transforms, for instance, the "agent" and "direct" attributes of an action into conceptual categories. While Slantstory uses a grammatical-sounding model of actions, with direct and indirect objects, Curveship can in fact realize sentences out of these where the agent is the direct object and the "direct" existent is the subject—when it realizes a sentence in the passive, for instance. So, both GRIOT and Curveship treat the seemingly grammatical attributes of ac-

tion in slightly different ways.

Furthermore, the templates that are used to represent sentences in Curveship, which is designed for narrative variation, are not well-designed for the generation of figurative text. Curveship's templates are set up to allow a slot for an agent, for example, which might eventually be filled with "the jaguar knight" "I" "he" or "you" depending upon how narrator and narratee are set and whether the noun phrase is pronominalized. Fig-S, however, may determine that the adjective "enflamed" should be used with this noun phrase because it will participate in the conventional metaphor LOVE IS FIRE. In this case, Curveship-Gen should generate either "the enflamed jaguar knight" "I, enflamed," "he, enflamed," or "you, enflamed." All the possibilities for combinations of figuration (not just the use of an adjective) and all the existing ways that Curveship can generate noun phrases need to be implemented in the next version of Slant.

Verso: Augmenting a Story Based on Genre

Verso, like MEXICA and Fig-S, reads a Slantstory XML file from the blackboard and outputs an updated one. While MEXICA is focused on plot and Fig-S selects an appropriate domain for blending particular representations of action, Verso's operation is based on a model genre. This subsystem operates by:

1. Detecting particular aspects of the in-progress story (typically actions with particular verbs, although possibly series of actions or sets of characters) that indicate the story's suitability to a particular genre, for all known genres.
2. Selecting the genre that is most appropriate.
3. Updating the story using rules specific to that genre. The narrative discourse is always updated by specifying attributes of and elements within "spin." This determines elements such as the focalizer, narrator, time of narrating, rhetorical style, and beginning and/or ending phrases to frame the story. The update can also contribute new actions to the story, which can influence the way that MEXICA continues to develop the plot.

This procedure brings a model of genre awareness into Slant, but it is an unusual process from the standpoint of conventional human creativity. More often than not, an author chooses a genre and then writes or tells something within it, rather than begin with a partial story and finding a genre that suits it. The overall effect, however, is to introduce sensitivity to an important aspect of human creativity.

Verso's model does not seem completely aligned with the direction of genre studies in recent decades. This field has moved from a formalist definitional framework of genre to one that is semiotic, focusing in particular on the "rhetorical study of the generic actions of everyday readers and writers" (Devitt 2008). Recently, genre studies has deemphasized and argued against the idea of genres as distinct categories with characteristic elements that identify them. Scholars now dispute the idea that characteristics can be identified and summed up to indicate the likelihood that a text is part of a certain genre. They note that few genres have true fundamental elements. Particularly in the case of literary genres (e.g. detective fiction, science fiction, horror, fantasy), even when there seem to be some core characteristics that all works within a category share, almost any "defining" characteristic could be countered by an example work which lacks that element but is still undeniably of that genre. Furthermore, a fundamental dilemma arises in the act of classification itself, the problem of "whether these units exist independently of the taxonomical scheme, or arise as a result of the attempt to classify" (Ryan 1981).

However, these recent concerns pertain most directly to scholarly and critical work; they do not bear upon the way genre is used in literary creativity. Sharp definitions of genre that are developed through writing practice have served many authors well, including Raymond Queneau, who used 99 different genres, modes, or styles to retell the same simple story in *Exercises in Style.* The problem of whether classification compels texts into categories is a problem for analysis, but it is a productive idea for literary creativity. Additionally, as Steve Neale has pointed out, "genres are instances of repetition and difference;" it is precisely through the differentiation from the established norms of a genre that a work can become part of it (Neale 1984). Verso, while making use of those "instances of repetition," also aims to effectively model the production of this necessary difference.

The genres that have been implemented so far are not literary, either in the sense of broad differentiations such as "prose" and "poetry," or in the sense of categories such as "romance," "cyberpunk," "noir," and so on. Instead, Verso uses a broader definition of what constitutes genre, one which includes categories that may very well be alternatively thought of as styles, modes, or even distinct media, and which relate to both fiction and non-fiction as well as to oral and written communication. In the introduction to *Writing Genres,* Devitt provides many examples of the influence of genre in our daily lives, including such wide-ranging categories as the joke, lecture, mystery novel, travel brochure, small talk, sales letter, and, most appropriately, the research paper (Devitt 2008). It is this broader conception of genre, rather than a strictly literary one, that Verso aims to model.

The genres implemented in Verso tend towards the stylistic rather than the thematic. In part due to the pre-existing capabilities of Curveship, and in part because of the domain in which MEXICA operates, the genres used are those that can be identified and produced through changes in the narrative discourse (focalization, time of narrating, order of events in the telling, etc.) rather than the story world domain (which could incorporate dragons, spaceships, magic, etc.).

A concrete example is provided by the "confession"

genre, which casts a story so that it sounds like it is being told to a priest at confession. To determine if this genre is applicable, the system checks to see if one or more actions are likely "sins" (robbing, killing, etc.) based on a list of these. Each "sin" raises the suitability of this genre. If "confession" is selected as the genre to use, the Slantstory XML representation is updated. A "sinner" is located—the agent of the last sinful action. This sinner is specified as the narrator (the "I" of the story). There is no narratee (or "you"), since we presume that the priest was not part of the events that were being told. The time of narrating is set to "after," which results in past-tense narration, and the "hesitant" style is used, injecting "um" and "er" into the story as if the speaker were nervous and reticent. Finally, a conventional opening ("Forgive me, Father, for I have sinned. It has been a month since my last confession.") and a conventional conclusion ("Ten Hail Marys? Thank you, Father.") are added.

The "confession" genre produces plausible and amusing results. Some of this has to do with the formulaic nature of the genre. As one reads additional confessions, the rigid, repetitive opening and conclusion can be amusing, because they model the ritualized interaction of confession. Read in this light, it is only more amusing that ten Hail Marys are always given for penance, whether the penitent tried to swipe something or committed a murder. Finally, because Spanish conquerors came to the Americas and imposed Catholicism on the natives, MEXICA-generated plots that are told in this genre can be read as a comment upon, or at least a provocation about, the colonial history of Mexico. Importantly, these two subsystems did not invent this juxtaposition of the MEXICA and Catholic ritual; rather, humans decided many years go to develop a story generator about the Mexica and decided recently to develop a "confession" genre template. However, the subsystems' collaboration as part of Slant involves automatically finding occasions when the juxtaposition of these two is particularly effective. Verso's work and MEXICA's work combine in Slant to provide more cultural resonance, to be more surprising and also to be more valuable by virtue of being thought-provoking.

In the current system 10 genres have been implemented: confession, diary, dream, fragments, hangover, joke, letter, memento, memoir, play-by-play, prophecy, and the default "standard" story. These take advantage of only a limited range of Curveship's narrative variation capabilities. For instance, the focalization of a story can be varied, but we have not yet implemented genres that focalize stories based on particular characters; similarly, Curveship is already capable of narrating with flashbacks and making other more elaborate changes in order. There are now only two prose styles that are used, "excited" for play-by-play and "hesitant" for confession. It would also be straightforward to elaborate the Slantstory representation and to modify Curveship-Gen to allow for expression that better relates to a wider variety of genres. In discussions so far we have already listed more than 100 genres, most of which we believe will be to some extent recognizable and applicable to the short stories produced by Slant.

Fig-S and GRIOT-Gen for Figuration

Fig-S reads a Slantstory XML file from the blackboard and updates it to include metaphorical content. Metaphor here can be understood as an asymmetrical conceptual blend in which all content from one domain called the "target space" is integrated with a subset of content from another called the "source space" (Grady, Oakley, and Coulson 1999). Fig-S currently implements ontologies representing several domains empirically identified as important in poetry such as "death" and "love" (Lakoff and Turner 1989) that can be used to generate metaphors such as REJECTION IS DEATH or ADMIRATION IS LOVE.

Fig-S begins by processing each of the actions from the Slantstory XML file to assess whether they will be replaced by metaphorical versions of the same action. Currently, there are two modes in which this processing can be done. If ONE-METAPHOR is set to true, then the Slantstory is analyzed to find which single source domain is appropriate to map onto the greatest number of actions in order to produce metaphors. Otherwise, each action will be analyzed individually in order to find an appropriate source domain to map onto it. The first mode typically results in more coherent output, the second mode typically results in a greater degree and variety of metaphorical output. As an example of an action that has been mapped onto by the source domain LOVE in order to produce a metaphorical action, the Slantstory action:

```
<action agent="virgin" direct="princess" id="61"
location="Texcoco Lake" tension="40" verb="get
jealous of" />
```

could be processed by Fig-S and added to the Slantstory as:

```
<figuration domain="fire">
<blend id="61" verb="get jealous of/burn for"
agent="virgin/burning one*agent"
direct="princess/hot*direct">
</figuration>
```

While Fig-S currently has implemented simple, metaphorical form of blending as a first step, it could be extended to use a more robust blending algorithm such as Alloy, or even to extend Alloy to result in even more novel, surprising, and/or culturally valued blends using an extended set of optimality principles.

GRIOT-Gen is used to produce specific output template from metaphorical actions in a Curveship-Gen format. For example, the metaphorical action above could be realized in a number of ways. The default produced by GRIOT-Gen, for a story in which neither virgin nor princess are narrator or narratee, would be structured as:

```
'61': 'the burning virgin [become/v] jealous-of
the incendiary princess',
```

however, it can alternatively be structured as:

```
'61': '[@virgin/s] like burning [get/v] jealous
of the incendiary [princess/o]',
```

if there is a preference for a simile-oriented style for the subject. It is also possible to use a "source-element/target-element" structure as in:

```
'61': 'the burning/virgin [get/v] jealous of and
[burn/v] for the incendiary/princess'
```

to be very explicit about every element that has been integrated. GRIOT-Gen currently has multiple such exposition forms implemented and is easily extensible.

Slant's First Stories

In the current system some spin (narrative discourse specification) is necessary, although it may simply involve the default settings, while figurative action representations are optional. To begin with, this amusing but flawed story was generated without figuration, but with contributions from MEXICA and Verso:

> Forgive me, Father, for I have sinned. It has been a month since my last confession. An enemy slid. The enemy fell. The enemy injured himself. I located a curative plant. I cured the enemy with the curative plant. The tlatoani kidnapped me. The enemy sought the tlatoani. The enemy travelled. The enemy, um, looked. The enemy found the tlatoani. The enemy observed, uh, the tlatoani. The enemy drew a weapon. The enemy attacked the tlatoani. The enemy killed the tlatoani with a dagger. The enemy rescued me. The enemy entranced, uh, me. I became jealous of the enemy. I killed the enemy with the dagger. I killed myself, uh, with the dagger. Ten Hail Marys? Thank you, Father.

The "sinner" who narrates the story dies, a problem which can also crop up when the "diary" genre issued. Since Verso can assign the genre of the story before the plot is complete, there was initially no way that Verso be sure that the character it selects as narrator will not die. This requires an interaction between the genre-selecting system, Verso, and the plot-generating system, MEXIA. We implemented an additional set of constraints on how the plotting could be done which either require or prohibit that a certain tension, as defined in MEXICA, arise. One of these tensions is "actor dead," letting Verso prohibit a narrator's death.

A story with figuration follows. This one is generated without the constraint for a single conventional metaphor to be used (ONE-METAPHOR is false), so there is a colorful diversity of less consistent metaphors. The genre chosen is "play-by-play," based on sports commentary, which may be a suitable one for the range of metaphor that is used:

> This is Ehecatl, live from the scene. The cold-wind eagle knight is despising the icy jaguar knight! The cold-wind jaguar knight is despising the chilling eagle knight! Yes, an eagle knight is fighting a jaguar knight! Look at this, the eagle knight is drawing a weapon! Look at this, the eagle knight is closing on the jaguar knight! The gardener eagle knight is wounding the weed jaguar knight! And now, the jaguar knight is bleeding! Yes, the consumed eagle-knight is panicking! And, eagle knight is hiding! Holy -- the snowflake slave is despising the chilling jaguar knight! The freezing-wind jaguar knight is despising the cold slave! And, yes, the cold-wind slave is detesting the chilling jaguar knight! A slave is curing the jaguar knight! And, the slave is returning to the city! And, the jaguar knight is suffering! The frozen jaguar knight is dying! Back to you!

MEXICA's stative descriptions of characters could probably be mentioned more rapidly, or perhaps not at all, to keep the action going. This could be done with an existing facility in Slantstory for omitting actions when narrating. This story would also benefit from pronominalization, which Curveship-Gen is capable of but which would need to be either turned on for all stories or specified at an earlier stage.

Slant's Research Potential

We plan to further develop the system we have initiated to explore new ways that computational creativity researchers can collaborate, new models of storytelling that abstract different sorts of expertise and emphasis, and new ways to compare the importance of and interaction between different aspects of story. We intend that the system will be used for empirical studies of how people receive generated stories and will also be brought into literary and artistic contexts.

Using the Slantstory XML blackboard, many different subsystems can be developed for Slant, which will allow Slant to be run with any subset of them. For instance, if Verso is turned off so that the specification of the narrative discourse is not done by that subsystem, either a default narrative discourse specification could be used (as would be the case now, since Verso is the only subsystem that updates this aspect) or that specification can be built up by one or more other subsystems. This allows the effect of each subsystem, in the context of Slant overall, to be carefully examined. Readers of stories generated under different conditions could be asked not only to rank the outputs in terms of quality, but also to comment on what they thought about particular elements (such as characters) and high-level qualities (whether the story was funny, for instance, or whether it seemed plausible).

The project can also facilitate a broader collaboration between researchers of story generation. As long as researchers find the Slantstory XML representation adequate for their purpose, they can develop new subsystems that help to build stories based on other theories or concerns. For instance, a researcher interested in how creativity occurs in social contexts could model the process in a unit that reads from and writes to the blackboard and models social influ-

ence and awareness. As just discussed, this new system could be tried in many combinations with existing systems and the outputs could be compared. This would help to show not only the importance of social creativity as modeled in this particular subsystem, but also how creativity of this sort interacts with plot generation using the engagement-reflection cycle, figuration based on conventional metaphors, and awareness of genre.

We also anticipate that Slant will supply stories for exhibition and publication in arts contexts, and the functional system itself could be part of a digital media, electronic literature, or e-poetry exhibit. In this way, Slant can contribute to creative practice, and reactions and discussion in this context can help us further develop a system that relates to contemporary literary concerns.

Acknowledgements

Thanks to Clara Fernandez-Vara and Ayse Gursoy for their discussions of genre and of early ideas about Slant.

References

Boden, M.A. 2004. *The Creative Mind: Myths and Mechanisms.* 2nd Ed. London and New York: Routledge.

Devitt, A.J. 2008. *Writing Genres*. Carbondale: Southern Illinois University Press.

Fauconnier, G. 1999. "Methods and Generalizations." In *Cognitive Linguistics, Foundations, Scope, and Methodology*, ed. T. Janssen and G. Redeker, 95–127. The Hague: Mouton De Gruyter: 96.

Fauconnier, Gilles, and Turner, M. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books.

Goguen, J., and Harrell, D.F. 2008. Style, computation, and conceptual blending. In Argamon, S., and Dubnov, S., eds., *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*. Berlin: Springer-Verlag. 291–316.

Grady, J. E., Oakley, T., and Coulson, S. 1999. Blending and Metaphor. In *Metaphor in Cognitive Linguistics*, ed. Gerard Steen and Ray Gibbs, 101–124. Amsterdam: John Benjamins.

Genette, G. 1983. *Narrative discourse: An essay in method.* Cornell University Press.

Harrell, D.F. 2006. Walking blues changes undersea: Imaginative narrative in interactive poetry generation with the GRIOT system. In *Proceeding of the AAAI 2006 Workshop in Computational Aesthetics: Artificial Intelligence Approaches to Happiness and Beauty*, 61–69. AAAI Press.

Harrell, D.F. 2007. GRIOT's tales of haints and seraphs: A computational narrative generation system. In Wardrip-Fruin, N., and P. Harrigan, eds., *Second Person: Role-Playing and Story in Games and Playable Media*. Cambridge, MA: MIT Press, 2007. 177–182.

Harrell, D. F. 2007a. "Cultural Roots for Computing: The Case of African Diasporic Orature and Computational Narrative in the GRIOT System," *Fibreculture Journal*, Vol. 11, http://journal.fibreculture.org/issue11/issue11_harrell.html

Lakoff, G. and Turner, M. 1989. *More than Cool Reason—A Field Guide to Poetic Metaphor*. Chicago: University of Chicago Press.

Montfort, N. 2009. Curveship: An interactive fiction system for interactive narrating. In *Proceedings of the NAACL HLT Workshop on Computational Approaches to Linguistic Creativity*, 55–62.

Montfort, N. 2011. Curveship: Adding control of narrative style. In *Proceedings of the Second International Conference on Computational Creativity*, 163.

Montfort, N., and Pérez y Pérez, R. 2008. Integrating a plot generator and an automatic narrator to create and tell stories. In *Proceedings of the 5th International Joint Workshop on Computational Creativity.* http://nickm.com/if/mexica-nn_ijwcc08.pdf

Neale, S. 1980. *Genre*. London: British Film Institute.

Pérez y Pérez, R., Ortiz, O., Luna, W. A., Negrete, S., Peñaloza, E., Castellanos, V., and Ávila, R. 2011. A system for evaluating novelty in computer generated narratives. In *Proceedings of the Second International Conference on Computational Creativity*, 63–68.

Pérez y Pérez, R., and Sharples, M. 1999. MEXICA: A computational model of the process of creative writing. In *Proceedings of the AISB Symposium on Creative Language: Humour and Stories*, 46–51.

Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*. 13(2): 119–139.

Pérez y Pérez, R., and Sharples, M. 2004. Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge Based Systems Journal*. 17(1): 15–29.

Polanyi, L. 1989. *Telling the American Story: A Structural and Cultural Analysis of Conversational Storytelling*. Cambridge, MA: The MIT Press.

Prince, G. 1988. The disnarrated. *Style* 22(1): 1–8.

Ryan, M-L. 1981. The why, what and how of generic taxonomy. *Poetics* 10: 109–126.

Ryan, M-L. 1991. *Possible worlds, artificial intelligence, and narrative theory.* Bloominton: Indiana University Press.

Using Theory Formation Techniques for the Invention of Fictional Concepts

Flaminia Cavallo, Alison Pease, Jeremy Gow, Simon Colton
Computational Creativity Group
Department of Computing
Imperial College, London

ccg.doc.ic.ac.uk

Abstract

We introduce a novel method for the formation of fictional concepts based on the non-existence conjectures made by the HR automated theory formation system. We further introduce the notion of the typicality of an example with respect to a concept into HR, which leads to methods for ordering fictional concepts with respect to novelty, vagueness and stimulation. To test whether these measures are correlated with the way in which people similarly assess the value of fictional concepts, we ran an experiment to produce thousands of definitions of fictional animals. We then compared the software's evaluations of the non-fictional concepts with those obtained through a survey consulting sixty people. The results show that two of the three measures have a correlation with human notions. We report on the experiment, and we compare our system with the well established method of conceptual blending, which leads to a discussion of automated ideation in future Computational Creativity projects.

Introduction

Research in Artificial Intelligence has always been largely focused on reasoning about data and concepts which have a basis in reality. As a consequence, concepts and conjectures are *generated* and *evaluated* primarily in terms of their truth with respect to a given a knowledge base. For instance, in machine learning, learned concepts are tested for predictive accuracy against a test set of real world examples. In Computational Creativity research, much progress has been made towards the automated generation of artefacts (painting, poems, stories, music and so on). When this task is performed by people, it might start with the conception of an idea, upon which the artefact is then based. Often these ideas consist of concepts which have no evidence in reality. For example, a novelist could write a book centered on the question 'What if horses could fly?' (e.g., Pegasus), or a singer could write a song starting from the question 'What if there were no countries?' (e.g., John Lennon's Imagine). However, in Computational Creativity, the automated generation and evaluation of such fictional concepts for a creativity purposes is still largely unexplored.

The importance of evaluating concepts independently of their truth value has been highlighted by some cognitive science research. Some of the notions that often appear in the cognitive science and psychology literature are those of *novelty*, *actionability*, *unexpectedness* and *vagueness*. Novelty is used to calculate the distance between a concept and a knowledge base. In (Saunders 2002), interestingness is evaluated through the use of the Wundt Curve (Berlyne 1960), a function that plots hedonistic values with respect to novelty. The maximum value of the Wundt curve is located in a region close to the y-axis, meaning, as Saunders points out, that the most interesting concepts are those that are "similar-yet-different" to the ones that have already been explored (Saunders 2002). The notions of actionability and unexpectedness were first introduced in (Silberschatz and Tuzhilin 1996) as measurements of subjective interestingness. Actionability evaluates the number of actions or thoughts that an agent could undertake as a consequence of a discovery. Unexpectedness is a measurement inversely proportional to the predictability of a result or event. Finally, vagueness is referred to as the difficulty of making a precise decision. Several measurements have been proposed in the literature for the calculation of this value, particularly using fuzzy sets (Klir 1987).

The importance of *generating* concepts which describe contexts outside of reality was underlined by Boden when she proposed her classification of creative activity. In particular, Boden identifies 'three ways of creativity' (Boden 2003): *combinational creativity*, *exploratory creativity* and *transformational creativity*. Transformational creativity involves the modification of a search space by breaking its boundaries. One reading of this could therefore be the creation of concepts that are not supported by a given knowledge base; we refer to these as fictional concepts herein. Conceptual blending (Fauconnier and Turner 2002) offers clear methods for generating fictional concepts, and we return to this later, specifically with reference to the Divago system which implemented aspects of conceptual blending theory (Pereira 2007).

We propose a new approach to the formation and evaluation of fictional concepts. Our method is based on the use of the HR automated theory formation system (Colton 2002b) (reviewed below), and on cognitive science notions of concept representation. In particular, we explore how the notion of *typicality* can improve and extend HR's concept formation techniques. In the field of cognitive psychology, typicality is thought of as one of the key notions behind concept representation. Its importance was one of the main factors that led to the first criticisms of the classical view (Rosch

1973), which argues that concepts can be represented by a set of necessary and sufficient conditions. Current cognitive theories therefore take into account the fact that exemplars can belong to a concept with a different degree of membership, and the typicality of an exemplar with respect to a concept can be assessed.

In the following sections, we discuss the methods and results obtained by introducing typicality values into HR. We argue that such typicality measures can be used to evaluate and understand fictional concepts. In particular, we propose calculations for three measures which might sensibly be linked to the level of *novelty*, *vagueness* and *stimulation* associated with a fictional concept. We generated definitions of fictional animals by applying our method to a knowledge base of animals and we report the results. We then compare the software's estimate of novelty, vagueness and stimulation with data obtained through a questionnaire asking sixty people to evaluate some concepts with the same measures in mind. The results were then used to test whether there is a correlation between our measurements and the usual (human) understanding of the terms novelty, vagueness and stimulation. We then compare our approach and the well established methods of conceptual blending. Finally, we draw some conclusions and discuss some future work.

Automated Theory Formation

Automated theory formation concerns the formation of interesting theories, starting with some initial knowledge then enriching it by performing inventive, inductive and deductive reasoning. For our purposes, we have employed the HR theory formation system, which has had some success inventing and investigating novel mathematical concepts, as described in (Colton and Muggleton 2006). HR performs concept formation and conjecture making by applying a concise set of production rules and empirical pattern matching techniques respectively. The production rules take as input the definition of one or two concepts and manipulates them in order to output the definition of the new concept. For example, the *compose* production rule can be used to merge the clauses of the definitions of two concepts into a new definition. It could, therefore, be given the concept of the number of divisors of an integer and the concept of even numbers and be used to invent the concept of integers with an even number of divisors. The *success set* – the collection of all the tuples of objects which satisfy the definition – of the new defined concept is then calculated. Once this is obtained, it is compared with all the previously generated success sets and used to formulate conjectures about the new concept. These conjectures take the form of equivalence conjectures (when two success sets match), implication conjectures (when one success set is a subset of another), or non-existence conjectures (when a success set is empty).

In domains where the user can supply some axioms, HR appeals to third party theorem provers and model generators to check whether a conjecture follows from the axioms or not. HR follows a best-first non-goal-oriented search, dictated by an ordered agenda and a set of heuristic rules used to evaluate the interestingness of each concept. Each item in the agenda represents a *theory formation step*, which is an instruction about what production rule to apply to which existing concept(s) and with which parameters. The agenda is ordered with respect to the interestingness of the concepts in the theory, and the most interesting concepts are developed first. Overall interestingness is calculated as a weighted sum (where the weights are provided by the user) of a set of measurements, described in (Colton 2002b) and (Colton, Bundy, and Walsh 2000). These were developed to evaluate non-fictional concepts, but some of them could be modified to evaluate fictional concepts for our system, and we plan to do this in future work. HR was developed to work in mathematical domains, but different projects have demonstrated the suitability of this system to work in other domains such as games (Baumgarten et al. 2009), puzzles (Colton 2002a), HR's own theories (Colton 2001) and visual art (Colton 2008).

Using HR to Generate Fictional Concepts

We are interested in the generation and evaluation of concepts for which it is not possible to find an exemplar in the knowledge base that completely meets the concept's definition. Throughout this paper we use the term *fictional concepts* to refer to this kind of concept. We use the HR system for the generation of such fictional concepts. To do so, after it has formed a theory of concepts and conjectures in a domain, we look at all the non-existence conjectures that it has generated. These are based on the concepts that HR constructs which have an empty success set. Hence, the concepts that lie at the base of these conjectures are fictional with respect to the knowledge base given to HR as background information. For example, from the non-existence conjecture:

$$\nexists(x)(Reptile(x) \,\&\, HasWings(x))$$

we extract the fictional concept:

$$C_0(x) = Reptile(x) \,\&\, HasWings(x)$$

To see whether typicality values can be used for the evaluation of these fictional concepts, we have introduced this notion into HR. Typicality values are obtained by calculating the degree of membership of each user-given constant (i.e., animals in the above example) with respect to every fictional concept which specialises the concept of the type of object under investigation (which is the concept of being an animal in this case). This is done by looking at the proportion of predicates in a concept definition that are satisfied by each constant. Hence, for each constant a_j and for each fictional concept C_i in the theory, we will have $Typicality(a_j, C_i) = t$, where $0 \leq t < 1$. For example, for the concept definition:

$$C_1(x) = Mammal(x) \,\&\, HasWings(x) \\ \,\&\, LivesIn(x, Water)$$

the typicality values for the constants in the set $\{Lizard, Dog, Dolphin, Bat\}$ are as follows:

$$Typicality(Lizard, C_1) = 0;$$
$$Typicality(Dog, C_1) = 0.\overline{3};$$
$$Typicality(Dolphin, C_1) = 0.\overline{6};$$
$$Typicality(Bat, C_1) = 0.\overline{6};$$

We see that the constant 'Dolphin' has typicality of $0.\overline{6}$ with respect to C_1 because a dolphin is a mammal which lives in water but which doesn't have wings – hence it satisfies two of the three predicates ($\approx 66.6\%$) in the definition of C_1.

It is important to note that for each fictional concept C there are at least n constants $a_1, ..., a_n$ such that $\forall j,\ 0 < Typicality(a_j, C) < 1$, where n is the number of predicates in the concept definition. We refer to these as the *atypical exemplars* of fictional concept C, and we denote this set of constants as $atyp(C)$. The atypical exemplars of C have typicality bigger than zero because they partly belong to C, and less than one because the concept is fictional, and hence by definition it doesn't have any real life examples. The number of atypical exemplars of a fictional concept is always more than or equal to the number of predicates in the concept definition because fictional concepts originate from the manipulation of non-fictional concepts, and hence, – given a well formed knowledge base – each predicate in a fictional concept definition will correspond to a non-fictional concept with at least one element in its success set.

Evaluating Concepts Based on Typicality

We explain here how typicality can be used to evaluate fictional concepts along three axes which we claim can be sensibly used to estimate how people will assess such concepts in terms of vagueness, novelty and stimulation respectively. This claim is tested experimentally in the next section. To define the measures for a fictional concept C produced as above, we use E to represent the set of constants (examples) in the theory, e.g., animals, and we use NF to denote the set of non-fictional concepts produced alongside the fictional ones. We use $|C|$ to denote the number of conjunct predicates in the clausal definition of concept C. We further re-use $atyp(C)$ to denote the set of atypical exemplars of C and the $Typicality$ measure we introduced above. It should be noted that the proposed methods of evaluation of fictional concepts have not been included into the HR program to guide concept formation.It is, however, our ambition to turn these measurements into measures of interest for ordering HR's agenda.

Using Atypical Exemplars

Our first measure, M_V, of fictional concept C, is suggested as an estimate of the *vagueness* of C. It calculates the proportion of constants which are atypical exemplars of C, factored by the size of the clausal definition of C, as follows:

$$M_V(C) = \frac{|atyp(C)|}{|E| * |C|}$$

As previously discussed, vagueness is a measurement that has been widely studied in the context of fuzzy sets. Klir (1987) emphasises the difference between this measurement and the one of *ambiguity*, and underlines how vagueness should be used to refer to the difficulty of making a precise decision. While several more sophisticated measurements have been proposed in the literature, as explained in (Klir 1987), we chose the above straightforward counting method, as this is consistent with the requirement that if concept C_a is intuitively perceived as more vague than concept C_b, then $M_V(C_a) > M_V(C_b)$. To see this, suppose we have the following two concepts:

$$C_1(x) = Animal(x)\ \&\ has(x, Wings)$$
$$C_2(x) = Reptile(x)\ \&\ has(x, Wings)$$

In this case, we can intuitively say that an animal with wings is more vague than a reptile with wings, because for the first concept, we have a larger choice of animals than for the second. In terms of typicality, this can be interpreted as the fact that C_1 has a larger number of atypical exemplars than C_2, and it follows that $M_V(C_1) > M_V(C_2)$.

Using Average Typicality

Our second measure, M_N, of fictional concept C, is suggested as an estimate of the *novelty* of C. It calculates the complement of the average typicality of the atypical exemplars of C, as follows:

$$M_N(C) = 1 - \frac{1}{|atyp(C)|}\left(\sum_{a \in E} Typicality(a, C)\right)$$

Novelty is a term largely discussed in the literature, and can be attached to several meanings and perspectives. In our case, we interpret novelty as a measurement of distance to the real world, as inferred in previous work in computational creativity research, such as (Saunders 2002). As an example of this measure, given the concepts:

$$C_1(x) = Bear(x)\ \&\ Furniture(x)\ \&\ Has(x, Wings)$$
$$C_2(x) = Bear(x)\ \&\ Furniture(x)\ \&\ Brown(x)$$

then, in a domain where all the constants are either exclusively bears or furniture (but not both), and assuming that all the bears and all the furniture are brown, we calculate:

$$M_N(C_1) = 0.\overline{6}$$
$$M_N(C_2) = 0.\overline{3}$$

This is because for C_1, all exemplars will satisfy just one of the three clauses ($\frac{1}{3}$) in the definition, hence this will be their average typicality, and C_1 will score $1 - \frac{1}{3} = 0.\overline{6}$ for M_N. In contrast, all exemplars will satisfy two out of the three clauses in C_2, and hence it scores $0.\overline{3}$ for M_N. Hence we can say that C_1 is more distant from reality, and hence more novel, than C_2. Consistent with the literature, and in particular with the Wundt Curve (which compares novelty with the hedonic value), we assume that the most interesting concepts have an average typicality close to 0.5. Note that this implies that fictional concepts whose definition contains two conjuncts are always moderately interesting in terms of novelty, as their average typicality is always equal to 0.5.

Using Non-Fictional Concepts

Our final measure, M_S, of fictional concept C is suggested as an estimate of the *stimulation* that C might elicit when audiences are exposed to it (i.e., the amount of thought it provokes). It is calculated as the weighted sum of all the non-fictional concepts, r, in NF that HR formulates for which their success set, denoted $ss(r)$, has a non-empty intersection with $atyp(C)$. The weights are calculated as the sum of the typicalities over $atyp(C)$ with respect to C. $M_S(C)$ is calculated as follows:

$$M_S(C) = \sum_{r \in NF} \left(\sum_{a \in atyp(C) \cap ss(r)} Typicality(a, C) \right)$$

This calculation is motivated by Ward's path-of-least-resistance model (Ward 2004). This states that when people approach the task of developing a new idea for a particular domain, they tend to retrieve basic level exemplars from that domain and select one or more of those retrieved instances as a starting point for their own creation. Having done so, they project most of the stored properties of those retrieved instances onto the novel ideas they are developing. As an example, the fictional concept:

$$C_1(x) = Horse(x)\ \&\ Has(x, Wings)$$

could lead to the following questions: Is it a mammal? Can humans ride it? Does it live in a farm? Does it fly? Does it lay eggs? Each of these questions can be derived from the corresponding HR generated concepts which have in their success set a large number of the atypical exemplars of C_1.

Experimental Results

To evaluate our approach, we started with a knowledge base of animals, based on similar inputs to those used for the conceptual blending system Divago (Pereira 2007), which is described in the next section. The concept map for a horse was taken from (Pereira and Cardoso 2003) and reapplied to each animal from a list of 69 animals reported in the National Geographic Kids website[1]. The relations were maintained when relevant, and extended when necessary according to the Generalized Upper Model hierarchy, as instructed in (Pereira 2007). Figure 1 illustrates a small part of the information we provided as background knowledge for HR to form a theory with.

To generate fictional concepts with HR, we used a random-search setup and ran the system for 100,000 steps, which took several hours. We limited the HR system to use only the *compose*, *exists* and *split* production rules, as described in (Colton 2002b). Extracting them from non-existence conjectures, the system produced 4623 fictional concepts, which were then automatically ranked in terms of their M_V, M_N and M_S values, as described above. From each of the ranked lists, a sub-list of 14 fictional concepts was created. The fictional concepts were taken at regular intervals so that they were evenly distributed numerically over the sub-lists, from highest scoring to lowest scoring. For

[1] kids.nationalgeographic.co.uk/kids/animals/creaturefeature

Figure 1: Details from the knowledge base for animals.

the M_N sub-list, all the fictional concepts with two clauses in the definition were first filtered out. For the M_V and M_S sub-lists, all the fictional concepts with more than two clauses in the definition were filtered out instead. The resulting sub-lists are given in tables 2, 3 and 4 of the appendix respectively.

We performed a survey of sixty people who were shown these lists and asked to rank them from 1 to 14 with respect to their own interpretations of the fictional concepts and their values. The aim of the survey was to verify how measurements M_V, M_N and M_S described above correlate with respect to common (human) understanding of vagueness, novelty and stimulation respectively. The survey was composed of four parts. The first three parts asked people to rank the three sets of 14 concepts in terms of vagueness, novelty and stimulation. We didn't include an explanation of our interpretation of these words in the questions, to encourage participants to use their own understanding of the three terms. The fourth part of the survey asked for a qualitative written definition of each of the three criteria of evaluation: vagueness, novelty and stimulation. Tables 2, 3 and 4 in the appendix report the three sub-lists of fictional concepts and the ranking (1 to 14) that our software assigned to them, along with the rankings obtained from the survey.

In order to establish whether our ranking and the survey rankings are correlated, we calculated Pearson's correlation, r, between the system's ranking and an aggregated ranking. The aggregated ranking was calculated by ordering the fictional concepts 1 to 14, according to the mean rank from the participants. We then calculated the respective 95% Confidence Intervals (CI) and p-values, using the alternative hypothesis that the correlations are greater than zero. We obtained the following results (quoted to 3 decimal places):

M_V/vagueness: r = 0.552, p = 0.020, 95% CI = [0.124, 1]

M_N/novelty: r = 0.697, p = 0.003, 95% CI = [0.350, 1]

M_S/stimulation: r = -0.029, p = 0.059, 95% CI = [-0.481, 1]

We can therefore conclude that there is strong and highly statistically significant correlation between the software rankings given by M_N and the survey rankings for novelty. We have similarly found a significant and moderate correlation

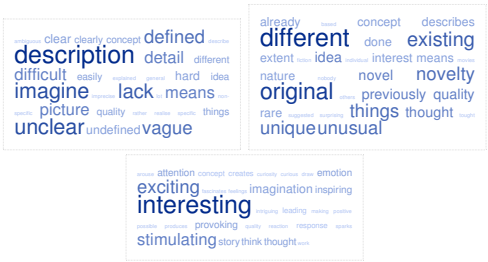


Figure 2: Word clouds: vagueness, novelty and stimulation.

with the survey rankings for M_V. Hence it appears that the novelty and vagueness measurements we suggested offer sensible calculations for the general understanding of these two terms for fictional concepts.

We found no correlation between the survey rankings for the stimulation value and the software measure M_S. This could be due to two reasons. Firstly, looking at the general descriptions of the word 'stimulating' given by people in the last section of the survey, they present a broader range of meanings than the word 'novel' or 'vague'. Moreover, these meanings are often very distant from the interpretation of the term 'stimulation' that we used in deriving the M_S measure. In figure 2, we present word clouds obtained from the definitions that people in the survey gave of the words vagueness, novelty and stimulation respectively. We can see that the the word cloud for vagueness includes words such as 'description', 'unclear' and 'difficult' as might be expected, and the word cloud for novelty includes words such as 'different', 'unusual' and 'original', also as expected. However, the word-cloud for 'stimulation' includes words such as 'emotion', 'exciting' and 'imagination'. This suggests a second reason that could explain the lack of correlation: our measure M_S lacks factors to estimate emotions and surprisingness elements, which will be studied in future work.

To explore the question of stimulation further, we looked at another measure of fictional concepts which might give us a handle on this property. Table 1 portrays the non-fiction concepts found (during the experimental session with HR described above) to have examples overlapping with the atypical exemplars of this fictional concept: $C_p(A) = isa(A, equine), pw(A, wings)$ [noting that $pw(A, X)$ means that animal A has a body (p)art (w)ith aspect X]. These non-fiction concepts comprised the subset of NF that was used to calculate $M_S(C_p)$. The non-fiction concepts overlapping with C_p are given along with a calculation which was intended to capture an essence of C_p as the *likelihood* of additional features being true of the fictional animals described by C_p. The calculation takes the sum of the typicalities of the atypical exemplars of the fictional concept which are also true of the non-fiction concept. We see that it is more likely for the winged horse to have feathers than to have claws, as pw(A,feathers) scores 10, while pw(A,claws) scores just 1. In future, we plan to use these likelihood scores at the heart of new measures. For instance, we can hypothesise that the inverse of average likelihood over all the associated non-fiction concepts might give an indication of how thinking about C_p could lead to less likely, more imaginative and possibly more stimulating real world concepts.

CONCEPT: isanimal(A,horse), pw(A,wing)	
Non-fictional concept	**Likelihood**
isa(A,bird)	6.5
isa(A,bug)	3.0
isa(A,mammal)	1.0
pw(A,lung)	8.5
pw(A,mane)	0.5
pw(A,tail)	7.0
pw(A,claws)	1.0
pw(A,teeth)	1.0
pw(A,eye)	10.5
pw(A,legs)	10.5
pw(A,fur)	1.0
pw(A,feathers)	10.0
pw(A,beak)	10.0
pw(A,hoof)	0.5
pw(A,claw)	5.5
existence(A,mountain)	2.5
isa(A,bug)	3.0
isa(A,bird)	6.5
isa(A,mammal)	1.0
hasAbility(A,carry)	1.0
hasAbility(A,hunt)	1.5
hasAbility(A,flying)	8.0

Table 1: Non-fiction concepts with success sets overlapping with atypical exemplars of the given concept, along with their actionability.

A Comparison with Conceptual Blending

We compare our system to the well-established conceptual blending technique, as this technique performs fictional concept formation and evaluation, as defined above. We therefore present a comparison of our system with Divago (Pereira 2007), which is a conceptual blending system implemented on the basis of the theory presented in (Fauconnier and Turner 2002). It applies the notions suggested by this theory in order to combine two concepts into a stable solution called a *blend*. Blends are novel concepts that derive from the knowledge introduced via the inputs, but which also acquire an emerging structure of their own (Pereira 2007).

Divago has been successfully tested in both visual and linguistic domains (Pereira 2007). It is comprised of six different modules: the *knowledge base*, the *mapper*, the *blender*, the *factory*, the *constraints module* and the *elaboration module*. The knowledge base contains the following elements: *concept maps* that are used to define concepts through a net of relations; *rules* that are used to explain inherent causalities; *frames* that provide a language for abstract or composite concepts; *integrity constraints* that are used to assess the consistency of a concept; and *instances* that are optional sets

of examples of the concepts. The mapper takes two random or user selected concepts and builds a structural alignment between the two respective concepts maps. It then passes the resulting mapping to the blender, which produces a set of projections. Each element is projected either to itself, to nothing, to its *counterpart* (the elements it was aligned with by the mapper), or to a compound of itself and its counterpart. The blender therefore implicitly defines all possible blends that constitute the search space for the factory.

The factory consists of a genetic algorithm used to search for the blend that is evaluated as the most satisfactory by the constraints module. The algorithm uses three reproduction rules: asexual-reproduction, where the blend is copied; crossover, where two blends exchange part of their lists of projections; and mutation, where a random change in one of the projections in a blend is applied. The factory interacts both with the elaboration module and the constraints module. The elaboration module is used to complete each blend by applying context-dependent knowledge provided by the rules in the knowledge base. The constraints module is used for the evaluation of each blend. It does this by measuring its compatibility with the frames, integrity constraints, and a user-specified goal (Pereira 2007).

The first high-level difference between Divago and our system derives from the motivations behind their implementations. Divago was constructed to test the cognitive plausibility of a computational theory of conceptual blending, and hence their aims were to construct complete and stable concepts, i.e., the blends. Details of the system's reasoning process, used for the formation and elaboration of such concepts, are therefore presented in the final output. Our system was instead constructed to generate fictional ideas of value. These are concise concepts which are purposely left in a simple and ambiguous form. The aim is in fact to find the concepts that stimulate the highest amount of thought and interest in an audience. The system's reasoning process is hence hidden from the outputs, and used only for evaluation purposes.

In the following paragraphs, we describe the parallels between Divago's modules and the different components of our system. In doing so, we identify the consequences of using each methodology. The first comparison that can be made is between the structures of the user-provided knowledge bases. In HR, the knowledge base is used only to define a set of concepts. It is hence equivalent in functionality to Divago's concept maps. The rules, frames and integrity constraints that need to be user-specified in Divago, are instead automatically learned in HR. They take the form of conjectures, non-fictional concepts and function specifications respectively. On one hand, this implies that HR has a greater degree of autonomy. On the other hand, HR is more prone to errors, as the constructed conjectures, non-fictional concepts and functions may not be relevant for the construction of fictional concepts.

For example, given an appropriate knowledge base, HR could construct the concept of an animal being amphibious, which is defined as an animal that lives in water and lives on earth. The same frame can be manually defined and used in Divago. However, HR will simultaneously construct other similar concepts. For example, the concept of animals that live in water and are red; or the concept of animals that live on earth and have four legs. If we assume that these concepts could be used for the evaluation of fictional concepts (as we plan to do in the future), then there is currently no way to differentiate between them in terms of the relevance they might have on the definition of a fictional concept (i.e., the system couldn't itself determine that an amphibian is more relevant than a water-living red animal). Moreover, HR is not capable of constructing all the rules, frames and constraints that Divago uses, but we believe that a similar functionality could be achieved through the use of typicality-based exemplar membership, and we plan to explore this possibility.

Despite the evident differences between their internal mechanisms, we can make a comparison between the blends produced by Divago's mapper and blender modules, and HR's non-existence conjectures. The first observation regards the range of the potential outputs. For HR, we only consider the concepts that are empirically known to be fictional. Divago's blends could instead be fictional, non-fictional, or exact copies of the two initial inputs. Moreover, Divago focuses only on one of the possible bijections between the elements in the concept maps. Pereira recognises that this restriction narrows the creative potential of the system (Pereira 2007, p. 117). HR is instead able to consider all possible structural alignments. Furthermore, Divago works on the blend of two randomly selected or user specified concepts, while HR can consider multiple concepts at once.

A component to develop and elaborate on HR's fictional concepts is still missing from our system, which we are planning to implement soon. In order to do so, we will take inspiration from Divago's factory and elaboration modules, while also taking into consideration the typicality values discussed above. However, as explained before, in our case this reasoning module will be used to calculate the *potential reasoning* that can originate from a fictional concept. In Divago, the factory and elaboration modules are instead used for the completion of a blend. Finally, Divago's constraints module can be compared with measures M_V, M_N and M_S introduced above. Divago's constraints module aims to evaluate a completed blend, while our system rates fictional concepts. Nevertheless, a correspondence between the evaluation methods can be noted. For example, the *topology* constraint used in Divago measures the novelty of a blend, like the M_N measure for fictional concepts investigated above, and the *integration* constraint used in Divago measures how well-defined a blend is, which is similar to the M_V measurement we have found is correlated with vagueness.

Conclusions and Further Work

We have proposed a method for generating and evaluating fictional concepts, using the HR theory formation system enhanced with typicality values. With the experiments above, we have shown that it is possible to create fictional concepts by using this process and that it is possible to meaningfully order the fictional concepts in terms of interestingness-oriented measurements. We have compared the automatically achieved evaluations with a ranking obtained through the analysis of a survey consulting sixty people. This

showed that our M_V and M_N measures are correlated positively with common understandings of vagueness and novelty respectively. Finally, we compared our approach to the one based on conceptual blending in the Divago system, which placed our work in context and highlighted comparisons which will inform future implementations.

Our system is still at the developmental stage. The experiment above, however, indicates that it is capable of creating fictional concepts that could be of interest to an audience. Moreover, this ideation process could be used at the heart of more sophisticated artefact generation systems, e.g., for poems or stories.

As previously discussed, the methods used to rank such fictional concepts have been shown to be useful, but also present some issues. Our next steps will therefore be to refine our current approach and implement new measures to estimate the interestingness of fictional concepts. To start this process, we will take inspiration from the notions analysed in (Colton, Bundy, and Walsh 2000) and used in the HR system, and modify them as appropriate. We will also look at other measurements suggested and used in Computational Creativity literature, such as Ritchie's criteria (Ritchie 2007). These, for example, could be used to assess the novelty of a fictional concept with respect to other fictional concepts.

We will then refine our measurement of typicality. To do so we hope to take inspiration from the theories proposed in cognitive science on the evaluation of the prototype theory and the weighting of category features. Each feature will be given a value called *salience*, used to indicate how important it is for the concept's definition. The salience values will then be used to calculate the typicality values with more accuracy.

Ultimately, we aim to introduce the notion of the *distortion of reality*. This measurement will serve to calculate how many real world constraints a fictional concept breaks. We will start by studying two methods for the calculation of values related to this. The first method is inspired from (Pease 2007) and will be based on the number of conjectures that each atypical exemplar of a fictional concept breaks. The second method is based on the scale of the distortion that an ontology would be subject to in order to include a fictional concept. We will also implement further methods for reasoning with fictional concepts. These methods will be used to estimate actionability; for the elaboration of fictional concepts; and for potential renderings of ideas in cultural artefacts such as poems and stories. We also plan to study how the different methods of measurement could be related to a rendering choice and vice versa. For example, non-vague concepts could be suitable for paintings, while actionable concepts might be more suitable for storytelling. We hope that such studies will help usher in a new era of idea-centric approaches in Computational Creativity as we hand over the creative responsibility for ideation to our software and address high level issues such as imagination in software.

Acknowledgments

We would like to thank the anonymous reviewers for the comments and suggestions we received. This research was funded by EPSRC grant EP/J004049.

References

Baumgarten, R.; Nika, M.; Gow, J.; and Colton, S. 2009. Towards the automatic invention of simple mixed reality games. In *Proc. of the AISB'09 Symp. on AI and Games*.

Berlyne, D. 1960. *Conflict, arousal, and curiosity.* McGraw-Hill Book Company.

Boden, M. 2003. *The Creative Mind: Myths and Mechanisms (second edition)*. Routledge.

Colton, S., and Muggleton, S. 2006. Mathematical applications of Inductive Logic Programming. *Machine Learning* 64(1):25–64.

Colton, S.; Bundy, A.; and Walsh, T. 2000. On the notion of interestingness in automated mathematical discovery. *Int. Journal of Human-Computer Studies* 53(3):351–375.

Colton, S. 2001. Experiments in meta-theory formation. In *Proc. of the AISB'01 Symp. on AI and Creativity in Arts and Science*.

Colton, S. 2002a. Automated puzzle generation. In *Proc. of the AISB'02 Symp. on AI and Creativity in Arts and Science*.

Colton, S. 2002b. *Automated theory formation in pure mathematics*. Springer.

Colton, S. 2008. Automatic invention of fitness functions, with application to scene generation. In *Proceedings of the EvoMusArt Workshop*.

Fauconnier, G., and Turner, M. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.

Klir, G. 1987. Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like? *Fuzzy Sets and Systems* 24(2):141–160.

Pease, A. 2007. *A Computational Model of Lakatos-style Reasoning*. Ph.D. Dissertation, School of Informatics, University of Edinburgh.

Pereira, F., and Cardoso, A. 2003. The horse-bird creature generation experiment. *AISB Journal* 1(3):257.

Pereira, F. 2007. *Creativity and artificial intelligence: a conceptual blending approach*. Walter de Gruyter.

Ritchie, G. 2007. *Some Empirical Criteria for Attributing Creativity to a Computer Program*. *Minds and Machines*, Springer, 17:76–99.

Rosch, E. 1973. Natural categories. *Cognitive Psychology* 4(3):328 – 350.

Saunders, R. 2002. *Curious Design Agents and Artificial Creativity: A Synthetic Approach to the Study of Creative Behaviour*. Ph.D. Dissertation, Department of Architectural and Design Science, University of Sydney.

Silberschatz, A., and Tuzhilin, A. 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. Knowledge and Data Engineering* 8(6):970–974.

Ward, T. B. 2004. Cognition, creativity, and entrepreneurship. *Journal of Business Venturing* 19(2):173 – 188.

Appendix

Concept Definition	Software Ranking	Survey Global Ranking	Survey Mean Ranking
An animal that has a body-part with which it can both see and eat	1	1	4.88
A mammal with feathers	2	4	7.11
A dolphin that lives on grass	3	11	7.89
A bird with tentacles	4	3	6.89
A bird with a trunk	5	10	7.58
A pig which is a bug	6	2	5.85
A fish with a trunk	7	7	7.37
An animal that lives both under freshwater and in the arctic	8	8	7.52
A fox which is an amphibian	9	9	7.54
A cow with tentacles	10	12	8.43
A fish which is also an otter	11	6	7.14
A salmon with feathers	12	13	9.82
A bat which is also a zebra	13	5	7.12
A gecko with spines	14	14	9.88

Table 2: Fictional concepts sorted from highest scoring to lowest scoring with respect to the software ranking for measure M_V, compared with the survey values for vagueness.

Concept Definition	Software Ranking	Survey Global Ranking	Survey Mean Ranking
A mammal that lives in the ocean that can fly	1	1	3.93
A mammal that lives in the ocean with wings	2	3	6.18
A mammal with wings that can be ridden by humans	3	2	3.94
A bird that lives in a forest that can swim under water	4	4	6.81
An invertebrate with legs that can swim under water	5	5	7.39
A mammal with wings that can hunt	6	7	8.11
A mammal that lives under freshwater and with fins	7	13	9.36
A mammal that lives both under freshwater and under the ocean	8	14	9.5
A mammal with fins that can hunt	9	12	9.24
An animal that lives both under freshwater and in a forest and that has wings	10	6	8.09
An animal that lives both under freshwater and in a forest and that has a fur	11	8	8.13
A bird that lives under freshwater and that can swim underwater	12	9	8.35
A bug that lives in a forest and has claws	13	11	9.14
A mammal with a tail that can fly	14	10	8.36

Table 3: Fictional concepts sorted from the highest scoring to the lowest scoring with respect to the software ranking for measure M_N, compared with the survey values for novelty.

Concept Definition	Software Ranking	Survey Global Ranking	Survey Mean Ranking
A fish with lungs	1	13	9.98
An animal that has eyes with which it can defend itself	2	3	5.88
A fish that can walk	3	7	7.22
An arachnid which is a mammal	4	11	8.85
A tiger with wings	5	2	5.85
An animal that lives under the ocean and that humans can ride	6	5	6.22
A wolf that can fly	7	4	5.97
A horse that lives under freshwater	8	10	8.27
A predatory bird with fins	9	12	9.19
A chicken that lives in the arctic	10	14	10.27
A dolphin which is also an arachnid	11	8	7.33
A chicken which is also a shark	12	1	5.3
An animal that has a body-part with which it can both see and eat	13	9	8.02
An animal with trunk with which it can fly	14	6	6.68

Table 4: Fictional concepts sorted from the highest scoring to the lowest scoring with respect to the software ranking for measure M_S, compared with the survey values for stimulation.

e-Motion: a system for the development of creative animatics

Santiago Negrete-Yankelevich and Nora Morales-Zaragoza
División de Ciencias de la Comunicación y Diseño
Universidad Autónoma Metropolitana (Cuajimalpa)
Av. Constituyentes 1054, México D.F. 11950, México
{snegrete/nmorales}@correo.cua.uam.mx

Abstract

This paper introduces *e-Motion*, a software system for the creation of *animatics*[1], which are important tools within the process of creation of animated graphics for TV. This type of animation, generated by the system from plots in plain text, allows production teams to envision how a final motion graphics piece can be developed. We argue that our system plays a creative role within the generative process. Specifically, our work is linked to a real production team, involved in the creation of animated shorts, called *Imaginantes,* for Mexican television.

Introduction

Computer systems intervene more and more in creative practices. They play different roles in teams of people working on projects that produce innovative work and whose overall process can be deemed creative by a suitably selected group of human experts. Just as in the case of creative teams formed strictly by human members, the blame for creativity can be distributed amongst the team members including computer systems (Maher,M.L. 2012).

In this paper we describe *e-Motion*, a computer system that builds *animatics* for a pre-production process to create motion graphics. In order to test the system we embed it in the process of *Imaginantes,* a TV production of a series of animated shorts (one minute long) based on texts of different authors aimed at encouraging viewers to get involved in Music, Literature, Fine Arts and Film. The first season (12 shorts), launched in October 2006, captivated young audiences who shared and published them through different social media and the web; some people even created their own shorts. Since then, *Imaginantes* has won numerous awards, and it's on the 4th season with a total of 46 shorts produced and delivered in several media (*Imaginantes*, 2006).

[1]An *animatic* is a visualization tool used in the pre-production process of an animation that informs about movement, narrative structure, framing aspects and visual effects to the production team before the animation is actually done.

e-Motion is part of a research project on computational creativity where we take a proven, creative process that produces a recognized, valuable product and use it as an environment to test our systems. We are interested in studying how the overall creativity is affected if computer systems take over different roles within the pre-production stage. The *Imaginantes* team and process are well defined as well as the work products that must be produced. We hypothesize that all stages and work products contribute to the overall creativity but we test our system's creativity by inserting it in the human process, to see how it affects the outcome and ask human members of the team to assess the system's performance.

There are two main advantages in this approach:

1. The system is assessed within a recognized real-world creative process, so we can avoid the toy-world generalization problem.
2. Our system plays a role within a human process, so it is easier for the human members of the team to assess the system's performance. They are experts in the area, they know very well what to expect.

The following are the main motivations for our work:

- Study computational creativity within real-world creative practices
- Understand the creative process of multi-sensorial content, sound and movement of visual narratives.
- Develop a computational system that works collaboratively in the creative production process of visual narratives.
- Develop sound user criteria to evaluate the system as a valuable tool.
- Experiment with automatically-created motion graphics to study how different frame, color and graphic element combinations transmit emotional content to an audience. We want to maximize narrative appeal while preserving the logical structure suggested by the original plot. (Malamed, 2009).

Animatics constitute an intermediate step towards a full motion graphics piece and, although they depict simple

representations, they have a complex structure and most of the high level architecture and elements of the final product. They are built from storyboards,[2] which in turn, are assembled from scripts and constitute an important tool within the process to develop motion graphics. They convey decisions about editing, camera framing and special effects. A production team can discuss several of these options using various *animatics* before embarking on the production stage, saving resources on this costly process. Hart actually describes *animatics* as the "future of motion control" to stress their importance (Hart, J. 2008).

As a starting point of our research into the nature of the creation of animated stories, we use the output of *Mexica* (Pérez y Pérez and Sharples, 2001), a computer system that generates story-plots about characters, places and themes of pre-Hispanic folklore; in particular, that of the Mexicas (most commonly known as Aztecs). These stories were originally represented in codices: pictographic documents where cultures from Mesoamerica used to write their history and other important aspects of their lives (Galarza, J.1997).

Mexica plots are useful for our purpose because they have very well defined and simple syntactic and narrative structures, yet they have an immense potential for expression. In fact most of the themes of classical literature can be represented by *Mexica* plots: betrayal, sacrifice, courtly love, deceit, loyalty conflict, etc.

The basic visual elements to assemble the animatic are provided by another system: *Visual Narrator (VN)* (Pérez y Pérez at al. 2012). This program illustrates story-plots from *Mexica* by producing sequences of still images composed of characters and scenes that literally represent the input plot by following a set of rules used in some pre-hispanic codices in a pictographic fashion. The rules specify how characters are presented according to their rank in society, activity, gender, and tension (emotional links represented by facial expression). They also tell how locations must be represented as well as action conventions. For instance, the rules describe how to represent a person that has a high social rank, who is talking to the people and who is angry. All characters used by *e-Motion* are built by VN, within the context of the process to produce a full motion graphics piece, the sequence produced by VN can be considered as a rough storyboard.

e-Motion generates *animatics* that follow a set of conventions for the representation of characters and locations, but also depict the dynamics of the action and emotion found in the original plot.

[2] Sequential drawings adapted from the script, depicted as concept drawings that illuminate and augment the script narrative. (Hart, J. 2008)

This paper is divided into four sections besides this introduction. In the first section we describe the *Imaginantes* project and why we think the use of computer systems can improve it. In the next section we explain how the system works. Then we propose a set of criteria to evaluate the system. Finally we present some conclusions and the current state of our project.

Building Motion Graphics for *Imaginantes*

Motion graphics are already present in everyday life. They are used in a variety of media: TV identities, film titles and credits, DVD's, videogames, smartphones interfaces, advertising displays and multiple media.

The creation of motion graphics is considered a special skill, usually handled by artists or graphic designers focused on the combination of design and television broadcast or film (Frantz, M. 2003). The term is an abbreviation of "Motion Graphic Design". Kook refers to it as the use of graphics, video footage and animation technology to create the illusion of motion or rotation, usually combined with audio (Kook, E. 2011).

The *Imaginante's* team consists of 8 to 10 people including: an executive producer, an art director, a design and animation coordinator, animators, illustrators, a musician or audio designer. The total time spent on the creation of a short ranges from 10 to 12 weeks.
The team starts with an original script that provides the general structure of the story. In some cases, this script has some extra indications describing shots, special effects, sound, etc.

Concept creation. The team collects all kinds of reference material related to the theme and author. It's a collaborative and exploratory work.

Pre-visualization. At this stage, the team develops two main tools: first, the storyborad, whose purpose is to show the key moments of the story in a sequence, suggest framing of the scenes and inform other specifics, like lighting, camera movements and special effects. It gives the entire pre-production team, a visual sequential breakdown of the main scenes in the narrative.
The other tool is the *animatic,* which brings the storyboard alive with motion, visual effects and a visual style for the animation. It is very effective tool to pace the narrative and timing and later add music and dialog (Hart, J. 2008).

Production. After the *animatic* is developed, illustrations are created, digitalized, and rendered; sound and music are also added to produce the final piece.

In a process like the one described above, a system like *e-Motion,* that suggests a variety of *animatics* with some camera-direction decisions based on the dramatic content the director wants to pursue, would be of great value for the production team. In the regular process there is a lim-

ited feedback the team receives from just one animatic per motion graphics project. It would also open new communication channels between the team members by expanding the discussion to new options and save time and work resources.

e-Motion

Plots, in *Mexica,* are built by selecting characters and structure from a repository of previous plots, combining them in a way that makes sense, story-wise, and trying to preserve well-known, successful emotional tensions. Emotional tensions are collected during the process in an emotional-tension profile for the story. This can be viewed as a chart where overall emotion varies against time. Emotional tension preservation in *Mexica* is a key factor in the guidance towards the selection and combination of elements for a successful plot.

A plot is a sequence of events in the order they occur in the story. It is the skeleton, the structure that tells the main events that occur in the story in a sequence of short action descriptions. Before the story is complete and ready for a final reader, it would have to be further developed to include all aspects that fulfill a creative piece of literary work. Yet, for our purposes it constitutes a good starting point because in the *Imaginantes* process the starting point is a plot from a text script (similar to a plot) with a few very structured actions or events in sequential order.

An example story plot can be seen below. Emotional tensions are inserted between brackets as they occur (Lc = love conflict); (Lr = life at risk), (Hr = health at risk), (Ad = actor dead):

Jaguar Knight was in Texcoco Lake
Enemy was in Texcoco Lake
Enemy got intensely jealous of Jaguar Knight (Lc)
Enemy Attacked Jaguar Knight (Lr)
Jaguar Knight fought Enemy
Enemy wounded Jaguar Knight (Hr)
Enemy ran away
Enemy went back to Texcoco Lake
Enemy did not cure Jaguar Knight (Lr)
Farmer prepared to sacrifice enemy
Enemy ran away
Jaguar Knight died by injuries (Ad)

Figure 1. A plot from *Mexica* and its tensions.

In *e-Motion*, a story plot with its emotional profile is taken as input as well as a set of characters generated by *VN*. An example character from *VN* can be seen in (Figure 2). It depicts the 'enemy' character from the story being angry as an emotional response to the fight it held with jaguar knight (see plot above in Figure 1).

Each line in the plot is an event and these, in turn, are incorporated into scenes by *e-Motion*. A scene has a set of performers. A performer is a character in action. That is, a character associated to an action to be performed. Characters in the animation include anything that appears on the screen and can be animated: humans, locations, emotional tokens, etc. They are all images and can be modified by 'moods'.

Figure 2. Enemy in angry mood

A character can have several moods depending on the representational variations available to it. A human can be looking right or left (there are only two dimensions, so far); he/she can be normal, angry or sad, etc. A location can have rain, sunshine, etc. Actions encode the movements of the characters on screen, they have a name and are a combination of the following basic animation operations: translation, scale and rotation. Performers can realize an action from the plot, like 'fight' or enact the manifestation of an emotional-tension like 'got intensely jealous of'.

Emotions in animation may be expressed in different ways: character moods, textures flying as clouds across the scene, icons depicting specific feelings —similar to the ones presented in codices—, scene elements or characters appearing as text, etc. In the latter case, the font, size and color of the text are used to manifest different emotions too. We call these: emotional tokens.

e-Motion builds scenes by following cinematic rules of composition: transition, character distribution, motion trajectories, framing and color. There are several options for each and the system builds the scenes of the animatic by selecting combinations of them that reflect the emotional profile of the original plot.

Emotional tensions may be of different kinds: love, hate, danger, anger, etc. As a story progresses, each event may bring new emotional tensions into consideration. Some of them may reinforce others previously introduced or may counteract them. Each new tension introduced in a plot manifest itself in the composition of a scene by affecting, in a certain amount, the dramatic quality of the scene. For every character there is an emotional profile consisting of three parameters (Table 1): affect (the level of acceptance /rejection the character feels), health (a level of well-being) and excitement (a measurement of the degree of arousal of the character). Each occurrence of a tension in the original plot contributes, by a certain, predefined, amount to some of the emotional profile parameters just mentioned. They are ranked from -3 to 3. Hence, whenever a character is to be integrated into a scene, its emotional profile defines

Dimension	-3	-2	-1	0	1	2	3
affection	hate	contempt	envy	neutral	sympathy	affection	love
health	death	illness	hurt	neutral	sane	welfare	happiness
excitement	horror	dread	cautious	neutral	surprise	joy	bliss

Table 1. Dimensions of emotional profile and their discrete values.

how it appears in it: a character's affect, health and excitement values determine its performance parameters in the scene: mood, speed, and emotional tokens (Table 1).
There is also a set of global rules that determine framing transition and trajectory for the characters and scenes.

Framing refers to decisions based on how close to frame an action of the story and how far to pull it back so the audience can see where the action is taking place. *e-Motion* choses its framing from a range of 4 types, based on camera angles of photography and film (McCloud, S. 2006): First plane, Middle-shot, Middle close-up and Extreme close-up. A story will always start with a first plane view (a), making a *zoom in* camera movement, followed by a middle-shot (b). As the story continues with the events carrying on, the characters change their affect and health levels. *e-Motion* will always look for the highest tension value to change the framing. E.g. (+2 or -2) or (+3 or -3) will change to frame (c) or (d) respectively. Every time we have a middle close-up the program sets back to a middle shot view until the levels of tension of the character are increased again to +3 or -3 values. The system will always end the story with a first plane view unless the overall level of excitement shows -3 or +3. In that case the system selects and extreme close-up and tilts the object in the frame.

Trajectory refers to the path that a character follows to enter and exit a scene. Each character has an entrance to the scene and a position to move to in the plane. It can also continue it towards the edges of the plane.
Emotional tokens have two main trajectory paths: *clouds* follow a curve; *stains* and *pictograms* stay in their initial position of appearance while varying their scale according to the character's level of excitement.

Transition refers to the sequence of movements from one key scene of the story to another, thus establishing the flow of the story (McCloud, S. 1993).

The example plot shown in Figure 1 produces the *animatic* presented in Figure 3 as a sequence of images that show some of its scenes. The third frame shows character Enemy in angry mood because he is jealous of Jaguar Knight. There is a green cloud traveling across the frame from left to right showing that feeling in the scene. The eagle and the cactus compose the name of "Texcoco Lake", the place where the scene takes place. The angry mood is due to a low level of affect (-1) and excitement (-1). The cloud is one of many possible manifestations of jealousy; this one in particular was selected randomly. The framing of that scene is solved in a middle shot angle. In the last scene Jaguar Knight is dead. The tension refers to Actor death, health (-3), excitement (0) and affect (0). There is an emotional token (red stain) which depicts the "being hurt"action and an up-scaling black cloud covers the body, dissolving into the final scene.

Assesing Animatics

To evaluate an animatic we have designed a questionnaire to measure its efficiency as a valuable tool and how new and surprising are the results to the team. (Boden, M. 1992). The questions were designed following interviews with actual members of the team where they set the parameters for effectiveness. The questionnaire is aimed at members of the *Imaginantes* team and will be rated higher according to their experience. Each questions offers 5 levels of agreement (1 means "totally disagree", while 5

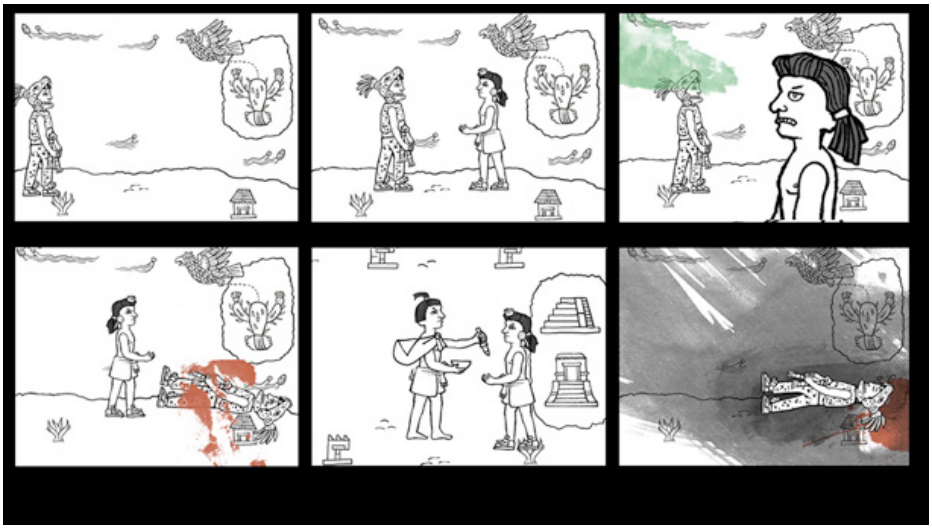


Figure 3. Six still frames extracted from an animatic as a means of illustration.

means “totally agree”):

1. Does the animatic show a logical selection of key moments from the text script?
2. Does the animatic give you general information on how many characters/objects/locations need to be drawn. Does it help you visualize a particular graphic style?
3. Does the animatic give you a general direction on time, special effects, movements and transitions you need to consider for the animation?
4. Does the animatic show an appropriate selection of camera angles according to the dramatic content of the script?
5. Is it likely that anyone on your team would have come up with a similar solution?

We end the questionnaire with a question that asks the participant to locate in a chart the balance of intensity/clarity and creativity of the animatic they have just seen (McCloud, S. 2006). We expect that *e-Motion* should ideally be ranked within the upper-right quadrant. (Figure 4). All team members have access to all work products, so they can relate them to the initial plot.

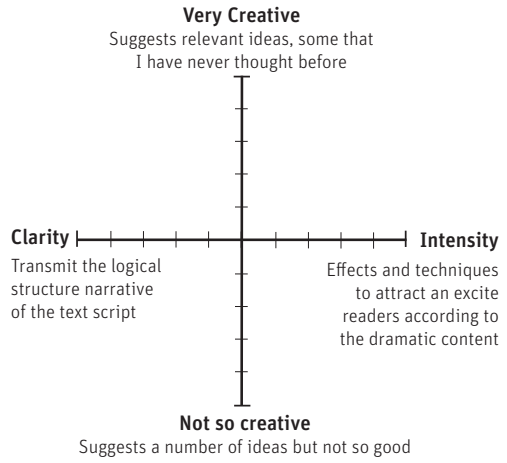


Figure 4. Efficiency and creativity chart

Conclusions and Future Work

e-Motion is a system that contributes to the development of motion graphics by creating *animatics.* By doing so, it plays an important role in the creative process since it determines a great deal of the structure and action of the final motion graphics piece. We consider our system important because it allows us to experiment with computational creativity in a proven creative process in the real world. This setting is particularly appropriate, we find, to evaluate the system's performance since the human part of the team can do so, with well defined parameters. As far as the authors of this paper are aware, there is no other work involving computational creativity and animation.

At the time of writing *e-Motion* is in β-test for its first version. We have run it with a few plots but the evaluation process, although it has already been designed, it still hasn't been applied. It will as soon as the system is ready.

In the first stage of the project we use *Mexica* plots as a starting point. This allows us to use a standard for plots that also contain information about their emotional tensions. In the *Imaginantes* project, the starting point is a script derived from an art piece. In subsequent versions we will use the experience with *Mexica* plots to standardize scripts taken from other sources and provide them with emotional descriptions. We are also planning to develop other systems that take over other aspects of the process and study their effects.

The system currently works with a set of tension values; these contribute cumulatively to character's emotional profiles, which determine how characters are animated. The general rules about framing, transition and trajectory follow basic cinematic standards but we would also like them to be determined by emotional parameters, an aspect that needs to be further investigated

References

Frantz, M. 2003. *Changing Over Time: The Future of Motion Graphics*. Thesis and Research work. In http://www.mattfrantz.com/thesisandresearch/motiongraphics.html.

Galarza, J. 1997. Los codices mexicanos. Arqueología Mexicana. 4:6-24.

Imaginates 2006. In http://www.fundaciontelevisa.org/imaginantes-2/page/3/.

Malamed, C. 2009. Visual Language for Designers: Principles for creating graphics that People Understand. Massachusetts: Rockport Publishers. 2252-3018.

Maher, M.L. 2012. Computational and Collective Creativity: Who’s Being Creative? In Proceedings of International Conference of Computational Creativity 2012.

Pérez y Pérez, R.; Sharples, M. 2001. MEXICA: a computer model of a cognitive account of creative writing. Journal of Experimental and Theoretical Artificial Intelligence. Volume 13, number 2, pp. 119-139.

Pérez y Pérez, R.; Morales, N. and Rodríguez L. 2012. In *Proceedings of International Conference of Computational Creativity 2012*.

Hart, J. 2008. *The Art of the Storyboard. A Filmmaker’s introduction.* Oxford, U.K: Focal Press. Elsevier 10:175.

McCloud, S. 1993. *Understanding Comics The Invisible Art.* N.Y.: Harper Collins Publishers. 70-89

McCloud S; 2006. *Making Comics: Storytelling Secrets of Comics, Manga and Graphic Novels.* Harper Collins Publishers. 19-25, 336-2900.

An Emerging Computational Model of Flow Spaces to Support Social Creativity

Shiona Webster, Konstantinos Zachos & Neil Maiden
Centre for Creativity in Professional Practice
City University London
Northampton Square, London, EC1V0HB, UK
{shiona.webster.2, k.zachos, N.A.M.Maiden}@city.ac.uk

Abstract

This position paper reports an emerging computational model of flow spaces in social creativity and learning that can be applied to guide human-centered creative cognition in social groups. In particular we are planning for the model to be applied to inform creative goal setting, creativity technique selection and adaptation, and guided social interaction during creative problem solving and learning.

Social Creativity and Learning

Social creativity and learning are increasingly important and related phenomena. Indeed, fostering creativity in learning is seen as a key direction with which to transform promising ideas into new processes, products or services (Retalis and Sloep, 2010). The explosion of information made available through the advancement of Web 2.0 has resulted in publicly available content that is continuously (re)created over the social media universe at an ever-increasing speed (Kaplan and Haenlein, 2010). Such rich content resources can provide a wealth of useful information that can support creativity and learning in both informal and formal social groups. Technologies are available to support such social creativity and learning and which support many different techniques that can be applied to solve problems creatively.

However, one outstanding challenge is which techniques to use to support different forms of social creativity and learning. The techniques can be categorized by the creative outcome that each can deliver when applied effectively, for example, the distinction between transformational, exploratory and combinatorial creativity (Boden, 1990), yet these categories offer few insights into effective processes that lead to social creativity and learning.

We argue that the success of social creative processes can depend on the extent to which people in the process are able to collect and relate information as well as create ideas collaboratively (Shneiderman, 2002), and whether these people experience flow and can create and learn, as opposed to becoming bored or anxious during it (Csikszentmihalyi, 1974).

For example, consider the following three different creativity techniques that could be deployed in a social creative process: (i) creativity triggers for business services, an exploratory creativity technique which directs the problem solver to solutions associated with creative ideas with qualities such as *convenience* and *trust*; (ii) constraint removal, a transformational creativity technique that removes or reduces perceived constraints to increase the possible search space e.g. (Onarheim, 2012), and; (iii) analogical problem solving, an exploratory creativity technique that transfers a network of interrelated facts from a mapped source domain to the target domain e.g. (Gick and Holyoak, 1983). Each of the techniques has different strengths and weaknesses. Analogical reasoning from a source domain necessitates information about the domain to be collected and related before ideas can be generated. Analogical knowledge transfer can then trigger the problem solver to generate multiple and more radical new ideas and concepts, but is cognitively difficult to do (Gick and Holyoak, 1983), and can lead to anxiety rather than flow and learning through the formation of new problem schemata. Constraint removal also necessitates information to be collected beforehand, and can lead to the generating of more ideas than with analogical problem solving (Jones et al., 2008).

We argue that criteria and mechanisms for selecting the most effective creativity technique at the right time in a social creative process are currently lacking. Whilst some experienced human consultants demonstrate an ability to select and adapt techniques to changing situations in social processes, such work is best categorized as craft, with little externalization of the knowledge and mechanisms applied. Moreover, if we are to embed such knowledge and mechanisms in computational environments that will guide and support people in the use of Web 2.0 creativity support tools during such processes, then new research is needed to discover and describe this knowledge and mechanisms – new research that we are undertaking in the COLLAGE consortium.

COLLAGE is a EU-funded Integrated Project, to inform and enable the design of effective Web 2.0 social creativity and learning technologies and services. The focus is to design, develop and validate an innovative cloud-enabled social creativity service-set that will support the

interlinking of learning processes and systems with (i) social computational services for inspiring learners, (ii) social affinity spaces for leveraging expression and exploration, and (iii) social game mechanics for supporting social evaluation and appreciation of creative behaviour. The new computational environment that we are developing to invoke different services in this set will need new capabilities to select between and recommend services, then adapt guidance to the social group during the social process. To deliver these capabilities, the approach adopted in COLLAGE is to develop a descriptive model of the desirable creative processes that is derived from existing theories and models of creativity and learning.

In this paper we report a first version of the model that describes how creativity and learning might be associated within a social process. The focus of the model is on descriptions of conceptual spaces in which flow, creativity and learning can be achieved. This model will, we anticipate, enable the design of effective social creativity and learning technologies and computational services with which to inform the selection and use of different creativity techniques and support tools.

Initial version of the COLLAGE Social Creativity and Learning Model

The COLLAGE Social Creativity and Learning (SCL) model is being developed to inform the principled selection and use of different techniques and computational services that support creative idea generation based on inspiration and recommendation engines, game mechanics and affinity spaces. To develop the model, we have drawn on Shneiderman's GENEX framework and Boden's concept of conceptual space to support social creativity and collaborative learning in workplaces. The use of each is reported in turn.

GENEX Framework

The SCL Model is based on the GENEX framework (Shneiderman, 2002) – an established situationalist model of social creative processes. The GENEX framework identifies four key processes during social creativity: (i) collecting information from public domain and available digital sources; (ii) relating, interacting, consulting and collaborating with colleagues and teams; (iii) creating, exploring, composing, and evaluating solutions; and (iv) disseminating and communicating solutions in a team and storing them in digital sources. These phases may occur in any order and may repeat and cycle iteratively.

Boden's Theory of Search Spaces

In COLLAGE we use Boden's model of creativity (Boden, 1990) to support the creative work by exposing novel information spaces to problem solvers and in turn, recommend creativity techniques that can be used to discover novel ideas for problem solving. Creativity is seen as a search of solution possibilities in a space based on measures of dissimilarity between possibilities as proxies for solution novelty (Ritchie, 2007). The search task is to find a complete solution among a set of partial and complete solutions that make up the search space. Hence, we assert that the problem at hand can be mapped to a problem of searching a space of solution possibilities.

The SCL model extends both the GENEX framework and Boden's concept of conceptual space to incorporate three capabilities that are critical to support social creativity and learning: (i) to reason about a new solution in order to discover the spaces in which novel and useful ideas are most possible; (ii) to guide the use of creativity techniques to search these spaces in order to discover novel and useful ideas; (iii) to engage the problem solver in such a way that he is fully immersed, feeling involved and successful in exploring the space of possible ideas.

To deliver these capabilities the SCL model includes: (a) a theory of goal-driven creative search spaces that computes novel search spaces and recommends creativity techniques to discover novel ideas; (b) a collaborative learning model for creativity that exploits a problem solver's real learning capacity in a collaborative and creative setting. The next section describes our use of the theory of goal driven creative search and new collaborative learning model that combines Csíkszentmihályi's notion of 'flow' with Vygotsky's Zone of Proximal Development.

Theory of Goal-driven Creative Search Spaces

Since search spaces have an implicit modularity in their structure (Johnson, 2005) and are often too large to search in a single search activity, the SCL model supports the discovery and exploitation of modular building blocks in the space. In COLLAGE we see the SCL model as a search-based creative process, i.e. a process of breaking down an initial, bigger problem into sub-problems, working out how those sub-problems fit together, and then tackling those sub-problems.

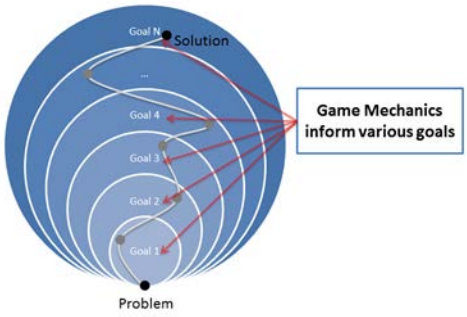


Figure 1. The overall search space divided into sub-spaces

Figure 1 shows a representation of two types of search space that we are seeking to describe and enable the search of, and discovery of ideas within. The first one is the larger overall search space that includes all of the ideas in the space. Since the space is too large to search in a single creative search activity, the space is searched through a series of creative search activities, each of which searches

the local part of the space expressed by the current goal, related to the ideas already discovered in the space. We can express a creative search activity in terms of a current subspace in a wider design space, and apply search-based techniques and theories to it.

One characteristic of creative search processes is that the criteria for evaluation of where to make the moves in the search space are not easy to capture in rule-bound form. Therefore, in COLLAGE we will employ game mechanics as a means to set intermediate goals in the overall search space that will both guide and engage problem solvers in further creative activities. Just as a game has levels that one tries to achieve, so should each creative search activity be informed by specific goals; game mechanics are used to provide these goals, which can be in the form of awards, credits and acknowledgements, in order to motivate and engage learners further in the creative problem solving process. Each subspace reveals a new goal that compels the problem solver to continue their creative search activity.

Collaborative Learning Model

The fundamental idea of how a subspace is traversed can be illustrated through an approach that combines Csíkszentmihályi's notion of 'flow' (Csíkszentmihályi, 1996) with Vygotsky's notion of the Zone of Proximal Development (Vygotsky, 1978). By combining both ideas, we introduce the concept of the collaborative learning model.

Csíkszentmihályi suggests that a person (or group) can experience 'flow' when fully immersed in an activity, feeling full involvement, an energized focus and success. Creativity is more likely to result from flow states (Csikszentmihalyi, 1996). Csíkszentmihályi identified three things that must be present to enter a state of flow:

- **Goals** – Goals add motivation and structure to the task; therefore, the person must be working towards a goal to experience flow.
- **Balance** – There must be a good balance between a person's perceived skill and the perceived challenge of the task. If one weighs more heavily than the other, flow probably won't occur.
- **Feedback** – A person must have clear, immediate feedback, so that he can make changes and improve his performance. This can be feedback from other people, or the awareness that progress is being made.

Vygotsky's conceptualisation of the zone of proximal development (ZPD) is designed to capture that continuum between the things that a learner can do without help, and the things that a learner can do when given guidance, or in collaboration with more knowledgeable others. According to Vygotsky, learning occurs in this zone.

Therefore, for learning to occur, people in a creative social process must be presented with tasks that are just out of reach of our present abilities. Tasks that are in the ZPD are tasks we can almost do ourselves, but need help from others to accomplish. After receiving help from others we will eventually be able to do the tasks on our own, thus shifting them out of our ZPD, in other words we have learned something.

In COLLAGE we combine flow and the zone of proximal development in the collaborative learning model depicted graphically in figure 2. The concentric circles represent the subspaces and goals that make up the larger overall search space. The horizontal axis represents a problem solver's domain-specific knowledge of the task at hand and the vertical axis represents the level of the task challenge.

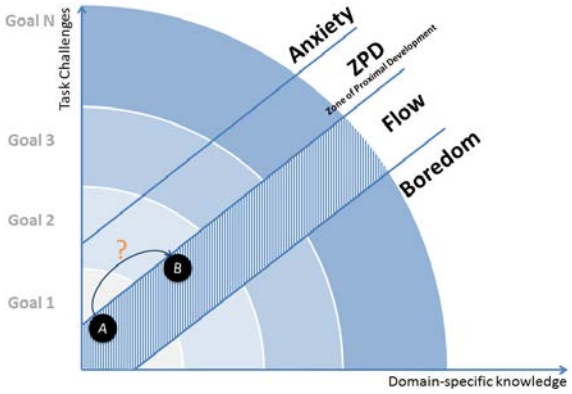


Figure 2. The collaborative learning model

As the problem solver's acquisition of knowledge advances in response to the challenges, an ideal path in the flow region would progress from the origin towards the upper right. The transition from starting point (A) to destination point (B) indicates the increase of knowledge and challenge that naturally traverses the ZPD, but under control and with the expectation that the problem solver will return to the flow zone again. We can see how a problem solver can move from bored (when their domain-specific knowledge exceeds their challenges) into the flow zone (where everything is in balance), but can easily move into a space where he needs some help. Most importantly, if we move upwards and out of the ZPD by increasing the challenge too soon, we reach the point where a problem solver starts to realize that he is well beyond his comfort zone. In COLLAGE, we seek to characterize each path connecting a knowledge/challenge space by the goal, balance and feedback needed to encourage flow:

- Game mechanics can provide achievable goals;
- Balance between a problem solver's domain-specific knowledge and skills and the perceived challenge of the task will be sought;
- Specific COLLAGE creativity-supported feedback services will provide clear and immediate feedback.

The next section describes how we are developing computational guidance for social creative processes.

Providing Guidance for Creative Processes

Our vision in COLLAGE is to utilize the emerging model with its concepts of information search for idea discovery, individual and social flow, and zones of proximal

development to recommend and adapt the use of different computational services and affinity spaces during a social creative process. The ambition is deliberately ambitious, with the aim to develop a computational environment to propose and adapt different services and spaces to maximize search, and achieve flow and learning. Indeed, according to Amabile, one of the single most important factors that induces creativity is a sense of making progress on a meaningful task (Amabile and Kramer, 2011), therefore the guidance will provide catalysts that induce progress, for example by setting achievable goals, providing resources, offering help and enabling users to learn from knowledge gained during previous creative activities.

The guidance is being developed to direct users along paths that connect a knowledge/challenge starting point (A) with destination point (B) in the collaborative learning model depicted in figure 2. We see the role of the creative process guidance to direct the problem solvers to effectively use the different creativity techniques, dependent on the situation, to bring balance to the knowledge/challenge. The creativity-supported feedback component incorporates all four processes from the GENEX framework.

The first version of the model identifies at least the following characteristics of social creativity and learning:

1. Defining and searching conceptual spaces of possible ideas
2. The setting of goals that render effective periods of individual and group flow achievable, within risking boredom and/or anxiety;
3. The maintenance of group flow in groups of distributed individuals who are often collaborating asynchronously;
4. Guiding individual learners into zones proximal development to encourage then support learning about creativity techniques and/or the problem domain as part of the flow process.

COLLAGE creativity services and affinity spaces need to support people to undertake creativity and learning activities with these characteristics. Moreover, we argue that each of these characteristics indicates one or more affordances of creativity services and affinity spaces for these characteristics of social creativity and learning. Consider each of the characteristics in turn.

Defining and Searching Conceptual Spaces of Possible Ideas

Any creativity service and affinity space should afford:

- One or members of the social group to undertake explicit information search and idea discovery in a conceptual space of possible ideas;
- These members to explicitly implement creativity services and affinity spaces that support different forms of transformational, exploratory and combinational creativity in a conceptual space.

An example of an established creativity service that affords exploratory information search and idea discovery is a creativity trigger. A creativity trigger is a generic desirable quality of a future solution that the social group is directed to discover new ideas to deliver – in software-based solutions, these qualities can include convenience, choice and trust. For example, use of the creativity trigger convenience guides one or members of the social group to undertake explicit information search and idea discovery in a space of ideas that can deliver the convenience of quality – and the search can be supported through the retrieval of information related to the quality of convenience.

Setting of Goals that Render Effective Periods of Individual and Group Flow Achievable

Any creativity service and affinity space should have assigned to it:

- A rating of the prototypical distance between the current set of ideas and the set goal that can be achieved through effective application of the creativity service or affinity space – the creative potential of the service or space;
- A rating of the prototypical distance between the content of the current set of ideas and the set goal content that can be achieved through effective application of the creativity service or affinity space – the creative potential of the service's or space's content;
- A difficulty rating indicating the potential level of difficulty that one person or a social group might encounter when learning and/or applying the service or space.

An example of a creativity service that demonstrates goal setting for individual and group flow is analogical reasoning. Analogical reasoning is the systematic transfer of a network of related information from a source domain to a target domain in order to generate new ideas in the target domain based on the transferred information (Gentner, 1983). Analogical reasoning has considerable potential to reconceptualise problem and solution spaces, hence the service's creative potential is high. Key to its success is the selection of source domain(s) from which to transfer knowledge for idea generation. Source domains semantically close to the target domain are easier for people to map to, but can lead to less new idea generation, and can risk boredom. In contrast, source domains semantically further from the target domain can lead to greater idea generation, are more difficult for people to map to and risk anxiety. Moreover, empirical evidence has revealed that people find analogical reasoning difficult (Gick & Holyoak 1983), hence they are likely to encounter difficulties during its use compared with creativity services that are easier to use such as creativity triggers.

The maintenance of group flow in groups of distributed individuals

Any creativity service and affinity space should afford:

- Collaborative creativity and learning by the members of the social group;
- The externalization of new ideas and knowledge that can be shared effectively with the members of the social group as part of a creative process;
- Explicit support for turn taking by members of the social group during the collaborative creative process.

An example of an affinity space that can afford the maintenance of group flow is design storyboarding. A storyboard is a graphic organizer in the form of illustrations or images displayed in sequence for the purpose of pre-visualizing a motion picture, animation, motion graphic, interactive media sequence or, for COLLAGE, a business or service design. Developing a storyboard from a set of existing concepts and ideas can afford collaborative creativity and learning by members of a social group through focused work on individual storyboard frames – the new ideas and knowledge generated from this creative work are shared with other members of the social group through the emerging storyboard, which acts as common ground in the collaborative creative process. Moreover, the development of discrete storyboard frames by individual members of the social group can afford turn taking based on game mechanics.

Guiding Individual Learners into Zones of Proximal Development

Any creativity service and affinity space should afford:

- The acquisition and learning of new knowledge in order to achieve flow as part of the individual and collaborative creative processes;
- The adaptation of any creativity service and affinity space in real-time to guide one or members of the social group into the zone of proximal development to support learning during creative flow.

An example of a creativity service that guides learners into zones of proximal development to encourage learning is the constraint removal service reported earlier. During the create activity, one or more members of the social group are required to envision a future version of the domain in which a constraint no longer applies or has been significantly relaxed. For example, during the exploration of new, more environmentally friendly operational concepts for an airport management system, one constraint that was removed was the variability of the weather. To generate new ideas, each member of the social group was required to envision an alternative reality of the domain in which weather was predictable. This required learning by the social group.

Future Work

Clearly we have only reported preliminary research in this paper, and much work remains to be done to develop, implement and validate the concepts proposed. The next stages of the research are to complete a first description of the model and build a first computational model of creative search spaces that the model will be applied to. We have a set of available computational creativity services that can be applied to search the space, as a basis for prototypical development of first versions of the computational model. We will look forward to reporting these advances in the near future.

Acknowledgements

The research reported in this paper is supported by the EU-funded COLLAGE integrated project 318536, 2012-15.

References

Amabile, T., Kramer, S., 2011. T*he progress principle: using small wins to ignite joy, engagement and creativity at work*. Harvard Business School Press.

Boden, M.A., 1990. *The creative mind: myths and mechanisms*. Routledge, London; New York.

Csikszentmihalyi, M., 1974. *Beyond boredom and anxiety*. Jossey-Bass Publishers.

Csikszentmihalyi, M., 1996. *Creativity: Flow and the psychology of discovery and invention*. HarperCollins, New York, N.Y.

Gick, M.L., Holyoak, K.J., 1983. Schema Induction and Analogical Transfer. *Cognitive Psychology* 1–38.

Johnson, C.G., 2005. Search and notions of creativity, in: *Proceedings of the IJCA 2005, Workshop on Computational Creativity*.

Jones, S., Lynch, P., Maiden, N., Lindstaedt, S., 2008. Use and influence of creative ideas and requirements for a work-integrated learning system, in: *International Requirements Engineering*, 2008. 16th IEEE. pp. 289–294.

Kaplan, A.M., Haenlein, M., 2010. Users of the world, unite! The challenges and opportunities of Social Media. *Business horizons* 53, 59–68.

Onarheim, B., 2012. Creativity from Constraints in Engineering Design: Lessons Learned at Coloplast. *Journal of Engineering Design*. 23, 323–336.

Retalis, S., Sloep, P., 2010. idSpace: A groupware system for supporting collaborative creativity.

Ritchie, G., 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17, 67–99.

Shneiderman, B., 2002. Creativity support tools. *Communications of the ACM* 45, 116–120.

Vygotsky, L.S., 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, Massachusetts.

Warr, A., O'Neill, E., 2005. Understanding design as a social creative process, in: *Proceedings of the 5th Conference on Creativity & Cognition*. pp. 118–127.

Idea in a bottle – A new method for creativity in Open Innovation

Matthias R. Guertler, Christopher Muenzberg, Udo Lindemann
Institute of Product Development
Technische Universität München
Boltzmannstr. 15, 85748 Garching
guertler@pe.mw.tum.de

Abstract

This paper presents an approach to increase the creativity of ideas/solutions in an idea contest. Analog to a *letter in a bottle* tasks are distributed in a randomized way to potential problem solvers. The idea contest is a method from Open Innovation which opens a company's innovation process to its environment (e.g. customers, suppliers). By using idea contests the creative potential of a large crowd of people can be used for developing innovative solutions for a specific task. Nevertheless, based on experience from industry projects we found that creativity often is limited. This paper presents an approach for increasing the creative potential of participants. The new integrated method combines idea contest with lead user's methods and aspects from synectics and communication.

Introduction

Open Innovation integrates a company's environment into its innovation process, e.g. in terms of customers or suppliers, and enables new innovations (Chesbrough et al. 2006). A popular Open Innovation method is the, usually web-based, idea contest which allows companies to publish a specific issue/task to a large crowd of people. These develop and post potential solutions for the issue. The idea behind is using the diversity of the crowd to generate creative and innovative solutions (Keinz et al. 2012). By giving participants/users the possibility to review other posts they can evaluate them as well as advance them. However, in industry projects we found that submitted solutions often are relatively homogeneous, of small number and of low degree of creativity.

In order to improve participants' creativity and the quality of posts, we developed the approach "Idea in a Bottle" based on the creativity method Synectics and Shannon's model of communication. The idea is to break up entrenched processes within an idea contest where users/problem solvers choose tasks to contribute. This is done by allocating the four phases of synectics (see next chapter) to different persons or groups and instrumentalize the primarily negative "noise source" of Shannon's model in a positive manner. We propose, by randomly allocating issues from idea-seekers to other users, their creativity is stimulated. The confrontation with an unexpected, non-self-chosen task helps overcoming our assumption that users usually choose issues they are familiar with. To direct the randomized process into efficient channels the Pyramiding method from the lead user concept is utilized. Thus, the first recipients of the issue do not solve it but act as agents and forward it to users they consider to be suitable and experienced on the specific field. These users submit suggested solutions to the idea seeker who evaluates the usefulness.

The proposed approach is applicable for issues/tasks of low and medium complexity. This means the improvement or new development of everyday products or the solution of medium complex problems. All issues should be processable without the need of highly specialized expertise or know-how.

The paper starts with a rough overview of the state of the art of Open Innovation, different user integration concepts, synectics, and Shannon's communication model. Based on this we present our I aB approach. We close the paper with a discussion about the planned evaluation of our approach by integrating I aB into a web-based idea contest platform.

State of the art

This chapter shortly explains the underlying concepts of the proposed approach "Idea in a Bottle". The basic elements are Open Innovation, synectics, Shannon's communication model, analysis-of-stimulating-word and pyramiding.

Open Innovation and Crowdsourcing

Open Innovation opens a company's innovation process to its environment (Chesbrough et al. 2006). The interaction with the environment enables innovations inside and outside the company. A concept focusing on the innovative potential of a large group of people is Crowdsourcing (Sloane 2011). The crowd can help elaborating and solving

specific issues and tasks by using the diversity of persons with their individual backgrounds, mindsets, abilities and knowledge (Keinz et al. 2012). A popular Crowdsourcing method is the usually web-based idea contest. Companies or individuals can publish issues on a web-platform. Users of the platform look at the issues and post ideas for solutions. Other users review these posts, advance them or get inspiration for new ideas. The goal is obtaining a large number of advanced ideas.

Lead User

According to von Hippel et al. (2006) lead users are characterized by (1) their capability for innovation as they are ahead of the market, and (2) their motivation for contribution. Several methods were developed to identify these innovative users. One method based on the snowball effect is the method Pyramiding. It is based on the assumption that people who are interested in a topic know other people who are more expert than themselves. Thus, Pyramiding starts with an initial group of people who name other people they consider to be more expert. These persons again name persons considered to be more expert. After some iterations potential lead users are gained (von Hippel et al. 2006).

Synectics

Synectic is a creativity technique based on brainstorming and was developed by W.J.J. Gordon in 1960 (Daenzer and Huber 2002). By postulating analogies from different fields, e.g. literature, nature, or symbols, users of this method are supported to find new solutions spaces for a stated problem. Synectic is a group technique with a proposed maximum of 10 participants who are instructed by a skilled moderator (Daenzer and Huber 2002). Synectics is structured into four phases which are passed through sequentially. The four phases are:

1. In the **Analysis** phase the group exposes the problem and states a problem definition. Also first solutions will be gathered and documented. Finally the problem should be restated.
2. The second phase, **Incubation**, is characterized by taking one step back with the help of building analogies. For example the group tries to build personal analogies by thinking how the object of interest feels. The outcomes of this phase are abstract solutions of the problem.
3. In the third step the stated analogies get analyzed and it is tried to transfer the solutions on the original problem. This can also be done with the help of force fit, i.e. oppressive reforming of the analogies. The results of the **Illumination** are new solutions approaches.
4. In the **Verification** phase the proposed approaches are used to elaborate solution concepts.

Presentation of Communication by Shannon

The communication process within an idea contest or synectics, e.g. the problem description formulated by the idea-seeker and interpreted by the problem solver, is one of the success factors for developing appropriate solutions. In 1963 Shannon and Weaver proposed a schematic diagram of a communication system (Shannon 1998). The proposed diagram consists of five essentially parts. These are information source, transmitter, channel, receiver, and the destination depict in Figure 1.

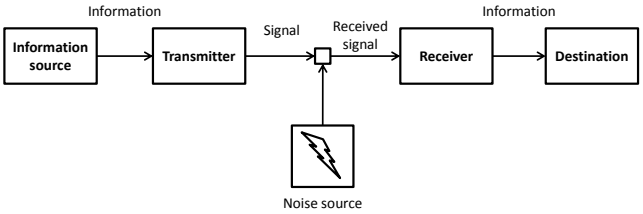


Figure 1: Schematic diagram of Communication by Shannon and Weaver (Shannon 1998)

The information source produces messages or sequences of messages which should be communicated. These messages can be of various kinds, e.g. letters or functions (Shannon 1998). The operator produces a suitable signal for transportation. The channel is the medium which transmits the signal to the receiver. The receiver reconstructions the signal and transports it to the destination, i.e. the person for whom the message is intended.

An important factor in Shannon and Weaver's diagram is the noise source introduced in the channel. This source leads to impacts on the communication. These impacts can change the original message by new interpretations, extension, reduction, or adaption (Lindemann 2009).

Analysis of stimulus words

This is a creativity method for developing new ideas by confronting participants with words not related to the actual topic. Participants analysis these words spontaneously by relevant criteria and build links to the original topic (Lindemann 2009).

A new method for creativity in idea contests: Idea in a bottle (I^2aB)

In order to increase the creativity and quality of ideas developed during an idea contest, we suggest redesigning the present communication process on an idea contest platform. So far, in analogy to Figure 1, an idea-seeker describes his issue (information source) by a problem description/task (transmitter) and publishes it on the platform. Here other users (receivers) can select this task, read it and derive their understanding of the task (destination). The following posting of solution ideas proceeds in an analogous way.

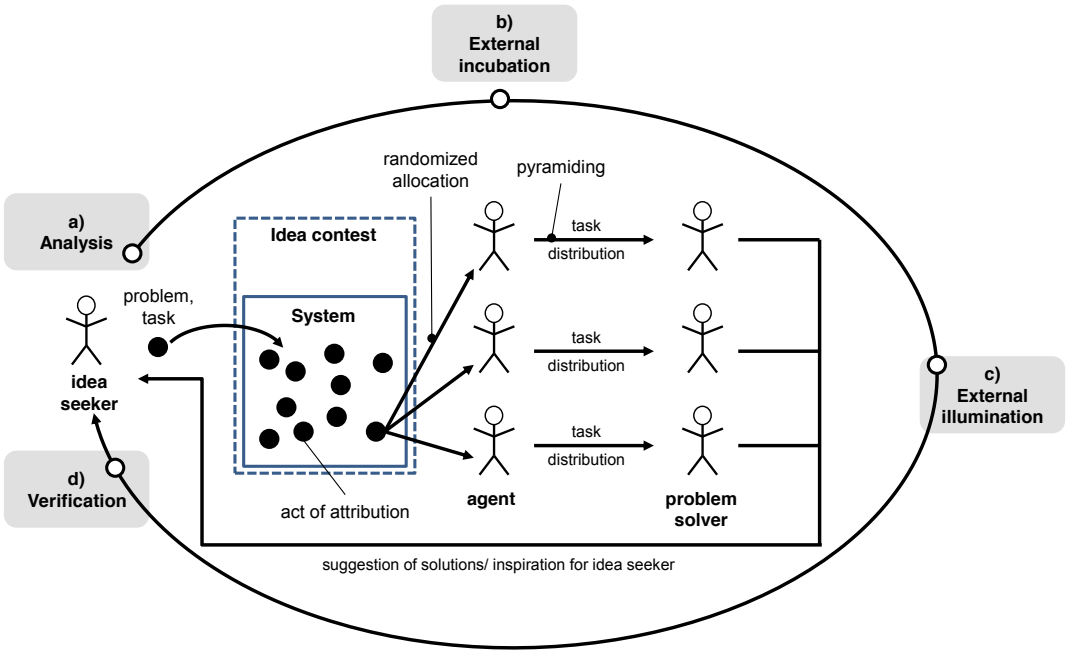


Figure 2: Model of Idea in a Bottle

Our approach splits up the four steps of synectics and distributes each to another group in order to increase efficiency and creativity. The analysis (1) is performed by the idea-seeker as "owner" of the problem. His analysis and statement of the issue affect the entire following I aB process. The incubation (2) is located by users of the platform who read and interpret the problem statement. Based on their understatement they link the issue to other users they consider suitable for the issue. The illumination (3) is conducted by the recommended users. They develop solution ideas for the given task based on their own interpretation of the issue and their personal background. The final verification (4) of the created solution ideas is performed by the idea-seeker himself again. Due to the incubation and illumination stage are not executed by the idea-seeker but by other users we term them "external".

The I aB approach instrumentalizes the "noise source" in terms of a randomized distribution of tasks to users. Instead of selecting familiar issues users get new tasks. Receiving unfamiliar topics shall support out-of-the-box thinking by providing an external perspective on a topic. To prevent demotivating users by receiving to many unfamiliar topics the distribution and solving step are separated by an intermediary Pyramiding step. The primary receivers of a task forward it to other users they consider to contribute a value gain to solving the problem. The process of the idea-seeker putting his issue into the platform without knowing who is receiving the issue is comparable to a letter in a bottle thrown into the sea. Hence, the approach was named "Idea in a Bottle" in analogy.

Figure 2 illustrates the concept of Idea in a Bottle (I aB). It consists of four stages analog the synectics approach, as mentioned previously:

In Stage 1 "**analysis**" idea-seekers phrase their problem/issue in a written task statement. It can also be enhanced by a picture or sketch. However, it is the intension to gain a compact description of the issue which focuses on relevant aspects. This increases the comprehensibility and thereby the user's motivation to deal with the issue. Thus, the number of words will be limited to abstracts' length with ca. 250 words in the beginning. Adding characterizing keywords supports the later forwarding process by the so-called agents. All issues are stored on the web-based idea contest platform.

In Stage 2 "**external incubation**" the Idea in a Bottle (I aB) system distributes the issue in a randomized way to three registered users on the platform. These users act as agents: they examine and, due to its shortness, interpret the issue. They are allowed to reply a potential solution idea. However, primarily their function is forwarding the issue to another user they consider able to contribute an add value for solving the issue, e.g. due to their experience/behavior in other idea contests on the platform. This forwarding process is based on the pyramiding method of the lead user concept. The optimal number of agents and problem solvers needs to be evaluated in practical tests. The randomized distribution and interpretation of the issue by the agents equate the noise source of Shannon's model. Summarized, the randomization stimulates the creativity of problem solvers in terms of analysis-of-stimulus-words (Lindemann 2009). By receiving forwarded issues, we as-

sume an increased motivation of problem solvers due to the honor of being recommended by other users.

Stage 3 is called "**external illumination**" due to the interpretation of the issue by other users. As described before, the potential problem solvers receive a random issue with the request for solving it. Since the problem solver does not know the real problem, only the problem statement, he builds new analogies of the given problem by interpreting the issue. These new analogies combined with the randomized distribution should lead to creative solutions which were not considered by the idea-seeker. Similar to Stage 1 also the solution ideas can be consist of text, photos or sketches. The size is limited, too. The problem solver is considered to contribute with solution ideas. Otherwise it is also possible to submit advices/hints which might indirectly draw the idea-seekers attention toward alternative potential sources and directions for a solution. Both the solution ideas and the hints are submitted electronically via the I aB system.

In Stage 4, "**verification**", the idea-seeker receives potential solution ideas and evaluates them regarding their applicability to his problem. In comparison to "classical" idea contest with a high effort in evaluating the gained ideas (Kain et al. 2012), we assume the verification effort for ideas created by I aB being lower since the solutions were elaborated by qualified system user. In the case of no appropriate idea the idea-seeker can submit his issue for a second loop.

Conclusion and next steps

The presented approach supports increasing the creativity and quality of solution ideas posted in an idea contest. This is realized by a combination of crowdsourcing, synectics, creativity techniques and pyramiding. Issues/tasks published by idea-seekers cannot be chosen by other users as in "classical" idea contests but are distributed in a randomized way to users who forward it to potential problem solvers. This randomized distribution combined with both the interpretation by the agent and the potential problem solver supports "out-of-the-box" ideas which might lead to innovative solutions. At this, the confrontation with unfamiliar topics acts as an analysis-by-stimulating-words and affects the problem solver's creativity. Additionally by being considered as a kind of expert by other users the motivation should tend to be high to contribute a solution.
I aB is enhancing, transferring and implementing classical creativity methods for new media and distributed product development activities. However, synectics was developed in the 1960s and is a classical creativity method which can be used in teams. We try to adapt this method for today's multi-media society.
To evaluate and proof these advantages we plan to implement I aB in a web-based way. The basis will be an idea contest platform at the institute which is being implemented at the moment and is specifically designed for testing new methods in the field of Open Innovation. This platform allows Open Innovation contest with students as well as industry as evaluation partners.
Here, we have the possibility to assess I aB in direct comparison to a "classical" idea contest. At this, the user pool of the platform can be used, as a sufficient community is seen as crucial success factor.
Besides others, the following questions need to be addressed:

1. Does the satisfaction and motivation of problem solvers increase?
2. Are differences regarding the number of replies to an issue; the quality and usefulness of ideas; the creativeness and the evaluation effort by the idea-seeker?
3. Is the choice of limitations of the issue description useful?
4. Are there any specific patterns within the forwarding process with frequently involved users?

Summarized, the expected key contributions of I aB are (1) a higher creativity, (2) a higher motivation of problem solvers and (3) a higher resulting quality of solution ideas.

References

Chesbrough, H., Vanhaverbeke, W. and West, J. 2006. *Open Innovation: Researching a New Paradigm,* New York, Oxford University Press Inc.

Daenzer, W. F. and Huber, F. 2002. *Systems Engineering: Methodik und Praxis,* Zürich, Verlag Industrielle Organisation.

Kain, A., Kirschner, R. and Lindemann, U. 2012. Utilization of Outside-In Innovation Input for Product Development. *International Design Conference DESIGN 2012*. Dubrovnik, Croatia.

Keinz, P., Hienerth, C. and Lettl, C. 2012. *Designing the Organization for User Innovation*.

Lindemann, U. 2009. *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden,* Berlin, Springer.

Shannon, C. E. 1998. Communication in the presence of noise. *Proceedings of the IEEE,* 86**,** 447-457.

Sloane, P. 2011. *A guide to open innovation and crowdsourcing : expert tips and advice,* London ; Philadelphia, Kogan Page.

Von Hippel, E., Franke, N. and Prugl, R. Efficient Identification of Leading-Edge Expertise: Screening vs. Pyramiding. Technology Management for the Global Future, 2006. PICMET 2006, 8-13 July 2006 2006. 884-897.

Multilevel Computational Creativity

Ricardo Sosa
Singapore University of Technology and Design
Singapore 138682
ricardo_sosa@sutd.edu.sg

John S. Gero
Krasnow Institute for Advanced Study
George Mason University
Fairfax, VA, 22030
john@johngero.com

Abstract

Creativity can hardly be understood in isolation from a context where values such as novelty and usefulness are ascribed. This paper presents a multi-level perspective for the study of creativity and formulates a framework for computational creativity that consists of 1) Culture; 2) Society; 3) Groups; 4) Products; 5) Personality; 6) Cognition, 7) Neural processes; and 8) CC processes. This model enables the definition of functional relationships among these levels. As an initial step to illustrate its usefulness, an analysis is made of the ICCC'12 proceedings in view of this model.

Introduction

The assessment of creativity is increasingly being recognized as an important direction in the research program of computational creativity (Jordanus 2011; Indurkhya 2012; Maher 2010, 2012). One of the main arguments is that creativity is in fact defined via the evaluation or ascription of values such as novelty and utility by third-parties beyond the creator(s). In other words, a creative product, person or process can hardly be understood in isolation from a context where such values are ascribed. Rather than a binary property, we consider that the composite value of creativeness is easier to define as a relative value ascribed by weak-to-strong levels of agreement or consensus to a range of products, persons or processes ranging from non-creative or routine to transformative or disruptive creativity (Gero 1990; Kaufman and Beghetto 2009). Because creativity is defined through the ascription of values (novelty, utility, expectation) in a system where creators and evaluators interact, this paper regards creativity as an eminently psycho-socio-cultural phenomenon; its aim is to frame computational creativity from such perspective.

Computational creativity (CC) has inherited an emphasis on individual processes, performance and products from the mainstream Artificial Intelligence worldview. In that paradigm, the agent architecture consists of autonomous individuals interacting with an external environment (Russell and Norvig 2005). CC has assumed that understanding individual behavior is a sufficient way of modeling creativity. A social-psychology approach to creativity began to illustrate the interaction between individual and external factors (Hennessey 2003). More recently, cultural-psychology creativity seeks to extend that work by shifting the architecture from a view of individual behavior "conditioned" by social factors and towards a more integrated view where interdependent relationships co-constitute a complex creative system (Glăveanu 2010)

This paper presents a multi-level perspective for the study of creativity and formulates a framework for computational creativity (CC). The aims of this work include: to enable new ways of thinking about CC from different disciplines, to support communication between research traditions, and to start mapping the units of analysis, variables and interactions between levels. The paper is organized as follows: Section 2 introduces key concepts and draws from the theoretical bases of this approach; Section 3 presents our framework and explains structural and functional aspects of our model. Section 4 evaluates this model using the 34 papers presented at the previous International Conference of Computational Creativity (ICCC'12). Section 5 closes the paper presenting modeling strategies and guidelines as well as discussing potential approaches to CC.

Background

Integrating scientific disciplines goes back to Comte's hierarchy of sciences according to the scale and complexity of theoretical tools (Mayer and Lang 2011). The role of cultural mediation in the development of cognitive functions has its origins in the tradition of cultural psychology since Vygotsky (Moran and John-Steiner 2003). Ecological models of creative problem solving integrate cognitive, personality, and situational factors (Isaksen et al 1993). Views of creativity as a social construct have been formulated elsewhere (Sawyer 2010; Westmeyer 2009).

Multilevel models that capture the interactions between psychological, social and cultural factors enable two complementary research directions. On the one hand, holistic explanations are possible by going up in the hierarchy drawing upon higher levels that moderate lower effects. On the other hand, reductionistic explanations go down in the hierarchy to inspect lower-level factors that account for high-level phenomena (Koestler and Smythies 1969). For example, accounting for cultural constructs can be essential to understand individual attitudes to altruism (Sheldon et

al). Likewise, the characterization of individual cognitive styles helps explain and manage group conflict (Kim et al 2012). Despite the disciplinary divides between psychology, anthropology and sociology, a phenomenon such as creativity may require a cross-disciplinary perspective that includes the interplay between levels of causality (Sternberg and Grigorenko 2001). Computational creativity has the potential to embark on cross-disciplinary modeling.

Contemporary personality research is a relevant example as it provides empirical support for the *irreducibility* postulate: i.e., "no scientific discipline is likely to subsume the others, all are needed" (Sheldon 2004). In the field of personality and well-being, multilevel approaches show the complex interactions and effects among factors located within and between levels of organization -from cultural to social, personality, cognition and neural processes (West et al 2010). Such integrated and interdisciplinary models account for moderator relationships between levels of organization.

The Multilevel Personality in Context (MPIC) (Sheldon et al 2011) and the Cognitive-Affect Personality System (CAPS) (Mischel and Shoda 1995) are two examples of how multiple levels of analysis can be integrated for a more reliable and complete understanding of complex human behavior –such as creativity. The MPIC model specifies the following levels: Culture, Social relations, and four levels of Personality: Self-Narratives, Goals/Motives, Traits/Dispositions, and Needs/Universals (Sheldon et al 2011). Reviewers of the MPIC model further suggest the addition of Situations to account for contextual factors beyond the bio-psychosocial (Mayer and Lang 2011).

In computational creativity, Indurkhya (2012) identifies the interplay between system levels by framing the following dilemma: when non-conscious or unintentional processes generate artifacts deemed as creative by an audience (i.e., works of art by a schizophrenic but also the ubiquitous cases of unexpected successful products), "where is the creativity?". A similar point can be made when considering the attribution of creativity to designs by Nature (McGrew 2012). Understanding the interplay between generative and evaluative processes of creativity has the potential to transcend such apparent paradox where at a given level it may seem like "there is nothing distinctive […] that we can label as creative" (Indurkhya 2012).

Maher (2012) frames the need for evaluation criteria that are independent of the generative process. Jordanus (2011) suggests a standardized approach to evaluation where key components are identified, clear metrics are defined and tests are implemented. The work presented in this paper is aligned to these aims and puts forward a structural and functional framework for an integrated cross-disciplinary study of computational creativity.

Multi-level Computational Creativity

The Multi-level Computational Creativity (MLCC) model builds upon the Ideas-Agent-Society (IAS) framework which maps three dimensions of creative systems: epistemological, individual and social dynamics (Sosa et al 2009). That structural framework synthesizes constructs from five influential theories related to creativity and innovation, i.e.: exemplars, proponents, and communities (Kuhn); innovations, entrepreneurs and markets (Schumpeter); noosphere, strong spirit and culture (Morin); domain, individual and field (Csikszentmihalyi); and logic, genius and zeitgeist (Simonton).

MLCC specifies eight separate levels of analysis: 1) Culture; 2) Society; 3) Groups; 4) Products; 5) Personality; 6) Cognition, 7) Neural processes; 8) CC processes. In addition, MLCC goes beyond the mapping of systemic dimensions and enables the definition of functional relationships among these levels. These relationships can be defined as independent or interdependent, i.e., the former represent processes that occur only within a single level in isolation, whilst the latter represent processes that are connected between levels. Namely, a range of cognitive functions can be studied in a CC system, some of which can be assumed to emerge from explicit lower-level neural processes, others that are defined only within the cognitive level, and a third type that lead to higher-level personality or group processes.

MLCC level	Sample models
1: Culture	Cultural dimensions in creativity (Lubart 2010); Peer-reviewed epositories (Duflou and Verhaegen 2011); IP law (Lessig 2008); Built environment (McCoy and Evans 2002).
2: Society	Gatekeeping (Sosa and Gero 2005a); Creative class (Florida); Migration (Hansen and Niedomysl 2009); Social capital (Fischer et al 2004).
3: Groups	Group conformity (Kaplan et al 2009); Team diversity (Bassett-Jones 2005); Group brainstorming (Sosa and Gero 2012).
4: Products	Rogers (1995) five factors (relative advantage, compatibility, complexity, trialability, observability).
5: Personality	Extraversion and dominance (Anderson and Kilduff 2009); Openness (Dollinger 2004).
6: Cognition	Creative cognition (Finke et al 1996); Bilingüalism (Adesope et al 2010).
7: Neural processes	Neuroanatomy (Jung et al 2010); NN models (Iyer et al 2009).
8: CC processes	Machine creativity (Cohen 1999; Maher et al 2012); Computational models of innovation (Young 2009; Sosa and Gero 2005b); tools and support systems (Liu et al 2004).

Table 1. The eight levels of our multi-level model of computational creativity (MLCC) and exemplary creativity models

MLCC level 1, Culture, refers to processes that either aim to model or draw from knowledge bases and corpora, cultural evolution, cultural dimensions, organizational culture, language and semiotics, economic impacts, taste and traditions, public policy, mass media, intellectual property, creative environments, planned obsolescence, aggregate search trends, market trends and anomalies.

MLCC level 2, Society, captures processes that account for the influence of –or seek to grow effects on– demographics, networks, migration, social influence and authority, roles and occupations, class structure, social capital, crowdsourcing, market segmentation, reputation and popularity, ethnic diversity, gender and aging, diffusion of innovations, crowd behavior.

MLCC level 3, Groups, refers to team dynamics, communities of practice, family and peer support, co-creation, artist collectives, art commission, brainstorming, change management and leadership, deliberation, collaboration/competition strategies, workplace, groupthink, game theory, adopter categories.

MLCC level 4, Product, captures intrinsic properties of creative artifacts largely determined by domain characteristics, techniques and processes, but also by technological or functional features, life-cycle, etc.

MLCC level 5, Personality, personality types, motivation, curiosity, extroversion, mental health, addictions, emotions, risk aversion, well-being, lifestyle, charisma, habit, expertise.

MLCC level 6, Cognition includes all processes related to creative cognition (intuition, insight, incubation, problem framing and solving, memory, concept formation, representation, fixation, association, analogy, divergent thinking, abductive reasoning, visual and spatial reasoning), perception, cognitive and attribution biases, heuristics.

MLCC level 7, Neural processes related to creativity including neuroanatomy (brain asymmetry), neuromodulation (risk, arousal, novelty), brain stimulation, neural network models of creative reasoning.

The final MLCC level refers to CC methods and techniques aimed at solving problems or generating creative solutions with no direct claims to model or being inspired by the other levels.

The MLCC model accounts for multiple levels of studying creativity, none of these levels is strictly new –Table 1 in fact includes references to multiple existing research programs that address creativity from each of the disciplinary traditions that specialize in such scales and units of analysis. The MLCC model brings them together and enables CC researchers to explore top-down and bottom-up connections between these levels.

Directionality of cross-level interactions in the MLCC model opens up a double opportunity in CC: on the one hand, it allows the study of generative processes between levels, i.e., how individuals create in isolation or in teams, how societal and cultural norms provide the bases for change cycles, what neural and cognitive processes help explain creative behavior, etc. On the other hand, it supports the less-explored study of evaluative processes between levels, i.e., how individuals, teams and society attributes creativeness to an artifact or a process, how cultures or subcultures accommodate for new additions or transformations, what neural or cognitive processes help explain the assessment of novel stimuli, etc.

Figure 1 provides a graphical depiction of an MLCC–inspired system showing a conventional organization of levels, i.e.: culture provides a general epistemological background where creators (individuals and teams) generate new artifacts targeted to specific audiences, a process mediated by distributors or promoters of artifacts –which are distinguished from the creators (for example, producers, market and art agents as separate stakeholders from designers and artists).

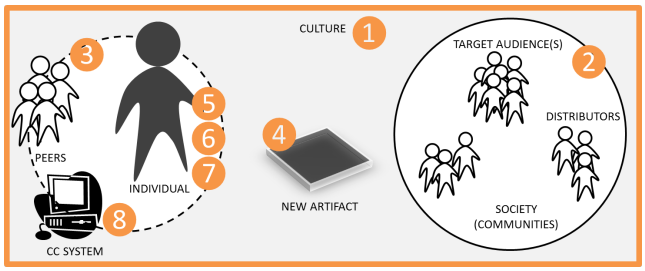


Figure 1. A system architecture to study individual creators and social evaluators interacting in a shared culture

However, the MLCC model supports a wide range of alternative modeling approaches, for example to study the 'maker' culture (Anderson 2012) or to focus on the cognitive processes of target audiences –for example how people are primed to rate and comment on the novelty and originality of artifacts in online forums (Sosa and Dong 2013). This flexibility of the MLCC model accommodates various research traditions, including minimal models where interactions between macro cultural and micro neural processes are explored –for example in cellular automata architectures (Sosa and Gero 2004).

CC presents clear advantages as a tool to advance theory building and for the systematic examination of assumptions and extraction of principles in multilevel systems (Fontaine 2006). Nonetheless, associated risks include: loss of clarity in the definition of interactions and causal relationships between levels; misalignment between disciplinary divides (research methods, units of analysis, linguistic traditions); and limited cross-level understanding between specialists.

Is the MLCC a creative artifact? It's not an entirely novel model –clear precedents were discussed going back as far as the XVII century. However, it does carry some novelty to the CC community. Its usefulness will be defined by its suitability as a modeling framework as determined firstly by the reviewers of the ICCC'13 and ultimately by the entire ICCC community. As an initial step to evaluate its relevance, the following section presents an analysis of the ICCC'12 proceedings using the MLCC model. The aim is to demonstrate its role in the analysis and discovery of trends in the current CC approaches, and identify gaps and connections between recent models of creativity.

Mapping ICCC'12 contributions

The 34 full papers published in the ICCC'12 proceedings were selected for this exercise (Maher et al 2012). They were classified in one or more of the MLCC levels according to their research aims and claims as stated by the au-

thor(s), as well as the target research agendas mentioned as part of future work. In addition to the eight MLCC levels, a ninth category was added during the review of these papers, which we named "Tools" and refers to work aimed at developing computational tools to support or enable human creativity (Gatti et al 2012, Hoover et al 2012).

Table 2 presents the 34 papers (rows) and their relation to the MLCC levels (columns). Entries related to generative processes in existing CC systems are marked by □, while entries related to evaluative processes in existing CC systems are marked by □. Examples of generative processes include a memetic algorithm "capable of open-ended and spontaneous creation of analogous cases from the ground up" (Baydin et al 2012); an evolutionary art system that generates artwork that "has been accepted and exhibited at six major galleries and museums" (Gabora and DiPaola 2012); and a system "able to generate pleasing melodies that fit well with the text of the lyrics, often doing so at a level similar to that of human ability" (Monteith et al 2012).

A paper may have multiple entries in different MLCC levels, for instance Morris et al (2012) present a "recipe engine" that draws from a corpus of online recipes published online (MLCC level 1), applies CC processes to generate new recipes (MLCC level 8), and these are subsequently analyzed by their typicality to a "recipe genre" (MLCC level 4).

Examples of evaluative processes include plans to include "feedback from journalists, critics, peers and audiences" (Burnett et al 2012); models of the cultural tastes and preferences of audiences (Indurkhya 2012); and plans to study "the cognitive processes of the viewers as they look at […] pictures" (Ogawa et al 2012).

A distinction is made when an entry refers to a future research approach that the authors identify as a valuable way forward –rather than an existing CC system. In such cases a plus sign qualifies the entry, respectively □$^+$ and □$^+$. Table 2 refers to the first author only due to space limitations. Some papers are rather comprehensive, such as Indurkhya (2012) and Maher (2012) which span across five MLCC levels each, but the overall average is 2.18 indicating a reasonable distinction among types of CC models.

Although these results systematic validation, they suggest a focus on generative processes in ICCC'12 (60 entries, including 43 existing and 17 target processes). Evaluation processes constitute a minority (14 total entries, half of them referring to target processes). These results are consistent with the preceding finding that "only a third of systems presented as creative were actually evaluated on how creative they are" (Jordanous 2011).

MLCC level 8 is the most prevalent: 40% of all papers discuss existing CC processes, and an additional 11% discuss target CC processes. Level 8 refers to methods and techniques aimed at solving problems or generating creative solutions with no direct claims to model or being inspired by the other MLCC levels. Examples include association-based computational creative systems (Grace et al 2012); small-scale "creative text generators" (Montfort and Fedorova 2012); and a music generator "inspired by non-musical audio signals" (Smith et al 2012).

MLCC level 4 is present in 30% of the papers; these present -or discuss approaches to generate- concrete artifacts identified as creative. They include Visual Narrator which constructs short visual narratives (Pérez y Pérez et al 2012); machine-composed music (Eigenfeldt et al 2012); and PIERRE which produces new crockpot recipes (Morris et al 2012).

AUTHORS	1	2	3	4	5	6	7	8	Tools
Agustini				●				●	●
Baydin	●							●	●
Burnett		○$^+$	○	●					
Charnley				●$^+$	●$^+$	●$^+$			
Colton				●				●	●
Eigenfeldt			○	●					●
Gabora	●			●		●$^+$		●	
Gatti									●
Grace				●				●	
Hoover									●
Indurkhya	○$^+$	○$^+$		●$^+$		○$^+$		●$^+$	
Jennings						●		●	
Johnson			●$^+$			●$^+$		●$^+$	
Jordanus								●$^+$	
Jursic									●
Keller									●
Li						●			
Linson						○$^+$			
Maher	●$^+$	●$^+$	●$^+$			●$^+$		●$^+$	
Monteith				●				●	
Montfort								●	
Morris	●			●				●	
Noy						●			
O'Donoghue	●					●			
Ogawa	●$^+$					○$^+$			
Pérez y Pérez			○	●				●	
Rank								●	
Ritchie						●$^+$		●$^+$	
Smith				●				●	
Sosa			●						
Toivanen	●		○					●	
Veale	○							●	
Wiggins						●		●	
Zhu	○								

Table 2. Classification of the ICCC'12 papers in MLCC levels

More than 30% of all papers address MLCC level 6, cognition. Most of these refer to the cognitive processes involved in the generation of creative artifacts, but a few do suggest the study of cognitive processes related to the evaluation of creativity (Ogawa et al 2012; Linson et al 2012; Indurkhya 2012).

MLCC level 1 is captured in 35% of all papers. In most, culture is used as a source in the creation of creative artifacts (as corpora or as evolutionary models at the cultural level). The remaining entries deal with culture as part of the evaluation of creativity. These include the application

of "literary criticism and communication theory […] to develop evaluation methods" (Zhu 2012) and "conceptual mash-ups" evaluated against "semantic structures seeking to replicate the semantic categories" (Veale 2012). Notably, MLCC level 7 –neural models of creativity- is not represented in ICCC'12, although progress is being made elsewhere (Iyer et al 2009).

Evaluation processes are scarce and gravitate mainly around MLCC levels 1 and 3 (Culture and Groups). 11% report assessment by small groups (audiences, experts) and the same number use culture as a metric for validating the results of a CC system (by comparison against or recreation of concrete cultural achievements). Only a couple of papers present potential ways of using societal factors or cognitive studies to understand how an artifact is ascribed creative value.

From an evaluation viewpoint, the ICCC'12 papers do not address the following MLCC levels: products (level 4), personality (level 5), neural processes (level 7) and CC processes (level 8). In this way, the MLCC model helps suggest future research approaches including:

- Models that incorporate explicit CC processes of evaluation of creativity, for example "automated critics" or "automated audiences" capable of replicating the assessment patterns of human judges (different scales and levels of domain expertise), as well as ultimately predicting the creativeness of computer-generated artifacts (Maher and Fischer 2012). Sample research question: "How may a computational system identify a masterpiece from mediocre artworks?"
- Models of neuro-mechanisms behind the creation as well as the evaluation of creativity. Systems that capture the connections between neural and cognitive processes. Sample research question: "How do basic functions such as short term memory or cognitive load moderate the evaluation of creative artifacts?"
- Models of the role of personality and motivation in the creation as well as the evaluation of creativity, for example systems that create or evaluate artifacts based on emotional predispositions, gender distinctions, and other personality dimensions. Models where creative behavior is moderated by environmental cues. Sample research question: "How do extraversion traits such as assertiveness moderate the assessment of creativity?"
- Models of intrinsic artifact properties identified in the evaluation of creativity according to intra and cross-domain characteristics. Sample research question: "What common assessment criteria do people apply when ascribing creativity in music, literature and architectural works?"

Beyond these "missing" levels (or ICCC gaps), this analysis leads to interesting new possibilities and distinctions in CC research:

- Culture can be approached in several ways in both generative and evaluative models: as the source of knowledge and generative techniques; as the standards against which new artifacts are evaluated by the creator and by the evaluators; as the status-quo that prevent or constrain acceptance of new artifacts; as factors exogenous to the domain from which creators can draw from and introduce novelty into their creative process; as rules and regulations that incentivize/inhibit creative processes; as market or cultural outlets and vehicles of promotion of creative value; etc.
- Societal and group levels can equally be considered in several ways: as large collectives or small groups (teams) collaborating in creative endeavors; as opinion leaders that influence both creators and evaluators; as cliques that provide support but may also polarize types of creators; as aggregate structures of behavior that lead to segmentation, migration, institutionalization; as temporal and spatial trends; etc.

As noted before, cognitive modeling may apply both to the generation and the evaluation of creativity. Likewise, although current computational tools are conceived for the creation of creative artifacts, computational tools could also support the individual and collective evaluation of artificial and human-produced artifacts –for example through the automated extraction of evaluation functions provided customer needs and requirements, which can then be used to guide either a computational system or human designers.

Discussion

How do works such as the Mona Lisa by Leonardo become icons of creativity? Elements to consider range from its intrinsic aesthetic and artistic qualities all the way to its distinctive history –including its theft from the Louvre in 1911 and the ensuing two-year international media notoriety (Scotty 2010). This illustrative case exemplifies the "entangled art-market complex" (Joy and Sherry 2003). Two CC scenarios are compared here where MLCC modeling is demonstrated:

1) "The Next Mona Lisa" CC model: a computational generative system is pursued that captures MLCC levels 6, 7 and/or 8 implementing symbolic or neural techniques (inspired or not by human capabilities) which aims to create a work of art comparable to the Mona Lisa, i.e., that receives the kind of appreciation and recognition gaining the status of a global cultural icon. The problem is that not only this approach seems rather implausible based on the current state of CC, it would also require a vast number of exogenous factors outside the reach of the system's authors –and would probably require very long time periods, considering that even *La Gioconda* path to prominence took more than four centuries (Scotti 2010).

2) "The Mona Lisa System" CC model: a multilevel computational system is based on the MLCC levels of choice (two or more from 1 to 8), which aims to capture the creation of a large number of artifacts, some of which (most) fall into complete oblivion, some of which (very few)

make it to the equivalent of mediocre galleries, local museums and living rooms of elite audiences, and some of which (an absolute minority) are preserved, disseminated and capture broad attention and consensus. Some works in this last category may gradually become part of the cultural heritage, may be used as exemplars in specialized domain training and in general education, may fetch high prices in auctions or be considered invaluable in monetary terms, and may ultimately play an influential role in shaping public taste as well as future artifacts within and beyond the domain of origin.

The latter approach opens interesting intellectual paths: What types of processes are capable of generating such diversity of artifacts? What commissioning, distribution and exchange mechanisms are sufficient to account for the observed skewed distributions of evaluation? What connections are possible, in principle, between intrinsic characteristics of artifacts and contextual conditions? What cross-level dynamics apply to creative systems from different domains and times?

Such an MLCC model can include a large number of elements, possibly derived from published studies –for example of art-market dynamics in this case (Debenedetti 2006; Joy and Sherry 2003). The output in such models may not be (only or necessarily) the creative artifact itself, but a deeper understanding of the principles that underlie creative generation and evaluation. This may include two or more MLCC levels, and over time, historical trajectories that are likely to be context and time-dependent. Thus the high relevance of CC approaches for the study of systems based on stochastic processes which can be re-run over sets of initial conditions in order to inspect causal relationships and long-term effects.

Lastly, the following guidelines are provided when building MLCC models, somehow extending the evaluation guidelines proposed by Jordanus (2011).

1) Identify levels to be modeled
   a) Define primary and complementary levels: realistically, empirical validation or data may be relevant only for one or two levels, whilst computational explorations can target other levels of interest.
   b) Identify level variables (experimental and dependent) that represent target factors and observable behaviors or patterns of interest.
   c) Define inputs and outputs at target levels, establishing the bootstrapping strategies of the model.
2) Define relationships of interest between levels
   a) Establish explicit connections above/below primary levels in the model
   b) Define irreducible factors, causal links and whether the model is being used for holistic or reductionistic purposes.
   c) Identify internal/exogenous factors to the system.
3) Depending on modeling aims, define outputs
   a) Define type and range of outputs, identifying extreme points such as non-creative to creative artifacts
   b) Capture and analyze aggregate data, model tuning and refinement
4) Evaluation of a MLCC system
   a) Validity may be achievable in some models where relevant empirical data exists at the primary level(s) of interest, but this may be inaccessible and even undesirable for exploratory models.

Acknowledgements

This work was supported in part by the US National Science Foundation under grant number SBE-0915482. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Adesope, O., Lavin, T., Thompson, T., and Ungerleider, C. 2010. A systematic review and meta-analysis of the cognitive correlates of bilingualism. *Review of Educational Research* 80(2):207–245.

Anderson, C. 2012. *Makers: The New Industrial Revolution*. London: Random House Business Books.

Anderson, C., and Kilduff, G. J. 2009. Why do dominant personalities attain influence in face-to-face groups? The competence-signaling effects of trait dominance. *Journal of Personality and Social Psychology*, 96(2):491–503.

Bassett-Jones, N. 2005. The paradox of diversity management, creativity and innovation. *Creativity and Innovation Management* 14(2):169–175.

Cohen, H. 1999. Colouring without seeing: a problem in machine creativity. *AISB Quarterly* 102:26–35.

Debenedetti, S. 2006. The role of media critics in the cultural industries. *International Journal of Arts Management* 8:30–42.

Dollinger, S. J., Urban, K. K., and James, T. A. 2004. Creativity and openness: Further validation of two creative product measures. *Creativity Research Journal* 16(1):35–47.

Duflou, J. R., and Verhaegen, P. A. 2011. Systematic innovation through patent based product aspect analysis. *CIRP Annals-Manufacturing Technology* 60(1):203–206.

Finke, R. A., Ward, T. B., and Smith, S. M. 1996. *Creative Cognition: Theory, Research, and Applications*. Cambridge: MIT Press.

Fischer, G., Scharff, E., and Ye, Y. 2004. Fostering social creativity by increasing social capital. In Huysman, M. and Wulf, V., eds., *Social Capital and Information*. Cambridge: MIT Press. 55–399.

Fontaine, R. G. 2006. Applying systems principles to models of social information processing and aggressive behavior in youth. *Aggression and Violent Behavior* 11:64–76.

Gero, J. S. 1990. Design prototypes: a knowledge represen-

tation schema for design, *AI Magazine* 11(4):26–36.

Glăveanu, V. P. 2010. Paradigms in the study of creativity: Introducing the perspective of cultural psychology. *New Ideas in Psychology* 28(1):79–93.

Hansen, H. K., and Niedomysl, T. 2009. Migration of the creative class: evidence from Sweden. *Journal of Economic Geography* 9(2):191–206.

Hennessey, B. A. 2003. Is the social psychology of creativity really social? Moving beyond a focus on the individual. In Paulus, P. B., and Nijstad, B. A. *Group Creativity: Innovation through Collaboration*. Oxford University Press. 181–201.

Indurkhya, B. 2012. Whence is Creativity? In *International Conference on Computational Creativity*, 62–66.

Isaksen, S. G., Puccio, G. J., and Treffinger, D. J. 1993. An ecological approach to creativity research: Profiling for creative problem solving. *The Journal of Creative Behavior* 27(3):149–170.

Iyer, L. R., Doboli, S., Minai, A. A., Brown, V. R., Levine, D. S., and Paulus, P. B. 2009. Neural dynamics of idea generation and the effects of priming. *Neural Networks* 22(5):674–686.

Jordanous, A. 2011. Evaluating evaluation: Assessing progress in computational creativity research. In *Proceedings of the Second International Conference on Computational Creativity*, 102–107.

Joy, A., and Sherry Jr, J. F. 2003. Disentangling the paradoxical alliances between art market and art world. *Consumption, Markets and Culture* 6(3):155–181.

Jung, R. E., Segall, J. M., Jeremy Bockholt, H., Flores, R. A., Smith, S. M., Chavez, R. S., and Haier, R. J. 2010. Neuroanatomy of creativity. *Human Brain Mapping* 31(3): 398–409.

Kaufman, J. C., and Beghetto, R. A. 2009. Beyond big and little: The four c model of creativity. *Review of General Psychology* 13(1):1–12.

Lessig, L. 2008. *Remix: Making Art and Commerce Thrive in the Hybrid Economy*. Penguin Press HC.

Liu, H., Tang, M., and Frazer, J. H. 2004. Supporting creative design in a visual evolutionary computing environment. *Advances in Engineering Software* 35(5):261–271.

Lubart, T. 2010. Cross-cultural perspectives on creativity. In Kaufman, J. C., and Sternberg, R. J., eds. 2010. *The Cambridge Handbook of Creativity*. Cambridge University Press. 265–276.

Maher, M.L. 2010. Design creativity research: from the individual to the crowd, In *Design Creativity 2010*, 41–47. London: Springer-Verlag.

Maher, M.L. 2012. Computational and collective creativity: Who's being creative?. In *International Conference on Computational Creativity*, 67–71.

Maher, M.L. and Fisher, D.H. 2012. Using AI to Evaluate Creative Designs. In *Proceedings of International Conference on Creative Design*, 45–54.

Maher, M.L., Hammond, K., Pease, A., Perez y Perez, R., Ventura, D., and Wiggins, G. (Eds.) 2012. *Proceedings of the Third International Conference on Computational Creativity*. http://computationalcreativity.net/iccc2012

McCoy, J. M., and Evans, G. W. 2002. The potential role of the physical environment in fostering creativity. *Creativity Research Journal* 14(3-4):409–426.

McGrew, S. 2012. Creativity in Nature. In Swan, L;, Gordon, R., and Seckbach, J., eds., *Origin(s) of Design in Nature*. Springer Netherlands. 43–55.

Montfort, N., and Fedorova, N. 2012. Small-Scale Systems and Computational Creativity. In *International Conference on Computational Creativity,* 82–86.

Moran, S., and John-Steiner, V. 2003. Creativity in the making. In Sawyer, R. K., John-Steiner, V., Moran, S., Sternberg, R. J., Feldman, D. H., Nakamura, J., and Csikszentmihalyi, M., eds., *Creativity and Development,* Oxford: Oxford University Press. 61–90.

Rogers, E. M. 1995. *Diffusion of Innovations*. Simon and Schuster.

Russell, S. J., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Scotti, R. A. 2010. *Vanished Smile: The Mysterious Theft of the Mona Lisa*. Vintage.

Sosa, R., and Dong, A. 2013. The creative assessment of rich ideas. In *Proceedings of the Ninth ACM Conference on Creativity and Cognition*. ACM Press.

Sosa, R., and Gero, J. S. 2004. A computational framework for the study of creativity and innovation in design: Effects of social ties. *Design Computing and Cognition 04*, 499–517.

Sosa, R., and Gero, J. S. 2005a. A computational study of creativity in design: the role of society. *AIEDAM Artificial Intelligence Engineering Design Analysis and Manufacturing* 19(4):229–244.

Sosa, R., and Gero, J. S. 2005b. Innovation and design: computational simulations. In *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy,* 1522-1528.

Sosa, R., Gero, J. S., and Jennings, K. 2009. Growing and destroying the worth of ideas. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, 295–304. ACM Press.

Westmeyer, H. 2009. Kreativität als relationales Konstrukt. In Witte, E.H., and Kahl, C.H., eds., *Sozialpsychologie der Kreativität und Innovation*, 11–26. Lengerich: Pabst Science Publishers.

Young, H. P. 2009. Innovation diffusion in heterogeneous populations: Contagion, social influence, and social learning. *The American Economic Review* 99(5):1899–1924.

Evaluating Human-Robot Interaction with Embodied Creative Systems

Rob Saunders
Design Lab
University of Sydney
NSW 2006 Australia
rob.saunders@sydney.edu.au

Emma Chee
Small Multiples
Surry Hills, Sydney
NSW 2010 Australia
emma@small.mu

Petra Gemeinboeck
College of Fine Art
University of NSW
NSW 2021 Australia
petra@unsw.edu.au

Abstract

As we develop interactive systems involving computational models of creativity, issues around our interaction with these systems will become increasingly important. In particular, the interaction between human and computational creators presents an unusual and ambiguous power relation for those familiar with typical human-computer interaction. These issues may be particularly pronounced with embodied artificial creative systems, e.g., involving groups of mobile robots, where humans and computational creators share the same physical environment and enter into social and cultural exchanges. This paper presents a first attempt to examine these issues of human-robot interaction through a series of controlled experiments with a small group of mobile robots capable of composing, performing and listening to simple songs produced either by other robots or by humans.

Introduction

Creativity is often defined as the generation of novel and valuable ideas, whether expressed as concepts, theories, literature, music, dance, sculpture, painting or any other medium of expression (Boden 2010). But creativity, whether or not it is computational, doesn't occur in a vacuum, it is a situated activity that is connected with cultural, social, personal and physical contexts that determine the nature of novelty and value against which creativity is assessed. The world offers opportunities, as well as presenting constraints: human creativity has evolved to exploit the former and overcome the latter, and in doing both, the structure of creative processes emerge (Pickering 2005).

There are three major motivations underlying the research of developing computational creativity: (1) to construct artificial entities capable of human-level creativity; (2) to better understand and formulate an understanding of creativity; and, (3) to develop tools to support human creative acts (Pease and Colton 2011). The development of artificial creative systems is driven by a desire to understand creativity as interacting systems of individuals, social groups and cultures (Saunders and Gero 2002).

The implementation of artificial creative systems using autonomous robots imposes constraints upon the hardware and software used. These constraints focus the development process on the most important aspects of the computational model to support an embodied and situated form of creativity. At the same time, embodiment provides opportunities for agents to experience the emergence of effects beyond the computational limits that they must work within. Following an embodied cognition stance, the environment may be used to offload internal representation (Clark 1996) and allow agents to take advantage of properties of the physical environment that would be difficult or impossible to simulate computationally, thereby expanding the behavioural range of the agents (Brooks 1990).

Interactions between human and artificial creators within a shared context places constraints on the design of the human-robot interaction but provides opportunities for the transfer of cultural knowledge through the sharing of artefacts. Embodiment allows computational agents to be creative in environments that humans can intuitively understand. As Penny (1997) describes, embodied cultural agents, whose function is self reflexive, engage the public in a consideration of the nature of agency itself. In the context of the study of computational creativity, this provides an opportunity for engaging a broad audience in the questions raised by models of artificial creative systems.

The 'Curious Whispers' project (Saunders et al. 2010), investigates the interaction between human and artificial agents within creative systems. This paper focuses on the challenge of designing one-to-one and one-to-many interactions within a creative system consisting of humans and robots and provides a suitable method for examining these interactions. In particular, the research presented in this paper explores how humans interacting with an artificial creative system construe the agency of the robots and how the embodiment of simple creative agents may prolong the production of potentially interesting artefacts through the interaction of human and artificial agents. The research adopts methods from interaction design to study the interactions between participants and the robots in open-ended sessions.

Background

Gordon Pask's early experiments with electromechanical cybernetic systems provide an interesting historical precedent for the development of computational creativity (Haque 2007). Through the development of "conversational machines" Pask explored the emergence of unique interaction protocols between the machine and musicians. MusiColour,

seen in Figure 1, was constructed by Gordon Pask and Robin McKinnon-Wood in 1953. It was a performance system comprising of coloured lights that illuminated in conjunction with audio input from a human performer.

But MusiColour did more than transcode sound into light, it manipulated its coloured light outputs such that it became a co-performer with the musician, creating a unique (though non-random) output with every iteration (Glanville 1996). The sequence of the outputs not only depended on the frequencies and rhythms but also repetition: if a rhythm became too predictable then MusiColour would enter a state of 'boredom' and seek more stimulating rhythms by producing and stimulating improvisation. As such, it has been argued that MusiColour acted more like a jazz co-performer might when 'jamming' with other band members (Haque 2007).

The area of musical improvisation has since provided a number of examples of creative systems that model social interactions within creative activies, e.g., GenJam (Biles 1994), MahaDeviBot (Kapur et al. 2009). The recent development of Shimon (Hoffman and Weinberg 2010) provides a nice example of the importance of modelling social interactions alongside the musical performance.

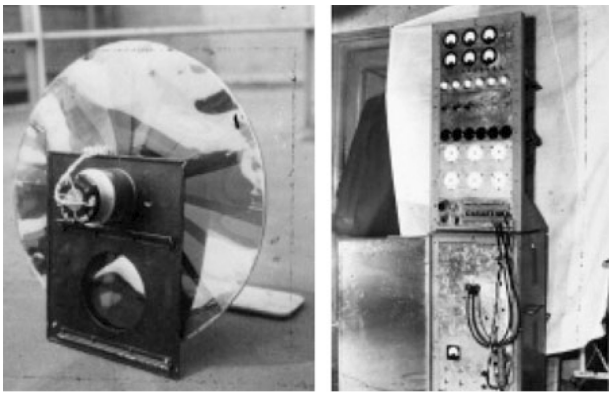


Figure 1: MusiColour: light display (left) and processing unit (right) (Glanville 1996).

'Performative Ecologies: Dancers' by Ruairi Glynn is a conversational environment, involving human and robotic agents in a dialogue using simple gestural forms (Glynn 2008). The Dancers in the installation are robots suspended in space by threads and capable of performing 'gestures' through twisting movements. The fitness of gestures is evaluated as a function of audience attention, independently determined by each robot through face tracking. Audience members can directly participate in the evolution by manipulating the robots, twisting them to record a new gesture. Successful gestures, i.e., those observed to attract an audience, are shared between the robots over a wireless network.

The robotic installation 'Zwischenräume' employs embodied curious agents that transform their environment through playful exploration and intervention (Gemeinboeck and Saunders 2011). A small group of robots is embedded in the walls of a gallery space, they investigate their wall habitat and, motivated to learn, use their motorised hammer to introduce changes to the wall and thus novel elements to study. As the wall is increasingly fragmented and broken down, the embodied agents discover, study and respond to human audiences in the gallery space. Unlike the social models embodied in MusiColour and Performative Ecologies, the social interactions in Zwischenräume focus on those between the robots. Audience members still play a significant role in their exploration of the world but in Zwischenräume visitors are considered complex elements of the environment.

Figure 2: Performative Ecologies: Dancers (Glynn 2008)

In 'The New Artist', Straschnoy (2008) explored issues of what robots making art for robots could be like. In a series of interviews, the engineers involved in the development of The New Artist expressed different interpretations of the meaning and purpose of such a system. Some questioned the validity of the enterprise, arguing that there is no reason to constructs robots to make art for other robots. While others considered it to be part of a natural progression in creative development "We started out with human art for humans, then we can think about machine art for humans, or human art for machines. But will we reach a point where there's machine art for machines, and humans don't even understand what they are doing or why they even like it." — Interview with Jeff Schneider, Associate Research Professor, Robotics Institute, Carnegie Mellon (Straschnoy 2008)

The following section describes the current implementation of the 'Curious Whispers', an embodied artificial creative system. The implemented system is much simpler than those described above, i.e., the robots employ a very simple generative system to produce short note sequences, but it provides a useful platform for the exploration of interaction design issues that arise with the development of autonomous creative systems involving multiple artificial agents.

Implementation

The current implementation of Curious Whispers (version 2.0) uses a small group of mobile robots equipped with speakers, microphones and a movable plastic hood, see Figure 3. Each robot is capable of generating simple songs, evaluating the novelty and value of a song, and performing those songs that they determine to be 'interesting' to

other members of the society – including human participants. Each robot listens to the performances of others and if it values a song attempts to compose a variation. Closing their plastic hood, allows a robot to rehearse songs using the same hardware and software that they use to analyse the songs of other robots, removing the need for simulation.

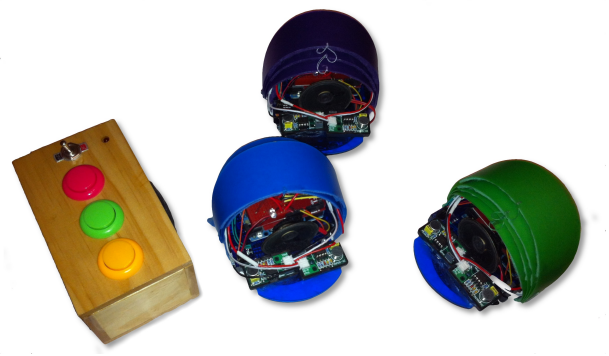


Figure 3: The implemented mobile robots and 3-button synthesiser.

A simple 3-button synthesiser allows participants to play songs that the robots can recognise and if a robot considers a participant's songs to be interesting it will adopt them. Using this simple interface, humans are free to introduce domain knowledge, e.g., fragments of well-known songs, into the collective memory of the robot society. For more information on the technical details of the implementation see Chee (2011).

Methodology

To investigate the interactions between robots and human participants we adopted a methodology from interaction design and employed a 'technology probe'. Technology probes combine methods for collecting qualitative information about user interaction, the field-testing of technology, and exploring design requirements. A well-designed technology probe should balance these different disciplinary influences (Hutchinson et al. 2003). A probe should be technically simple and flexible with respect to possible use: it is not a prototype but a tool to explore design possibilities and, as such, should be open-ended and explicitly co-adaptive (Mackay 1990). The probe used in this research involved three observational studies exploring different aspects of the human-robot interaction with the embodied creative system.

The observational studies were conducted with different arrangements of robots and human participants, allowing us to observe how interaction patterns and user assessments of the system changed in each configuration. Each session was video recorded and at the end of each session the participants were interviewed using a series of open-ended questions. The interview was based on a similar one developed by Bernsen and Dybkjær (2005) in their study of conversational agents. Employing a 'post-think-aloud' method at the end of each session the participants were first asked to describe their experiences interacting with the robot. A similar method was used in the evaluation of the Sonic City project (Gaye, Mazé, and Holmquist 2003). The video recordings were transcribed and interaction events noted on a timeline. The 'post-think-aloud' reports were correlated with events in the video recordings where possible.

Six participants were observed in the studies. The participants came from a variety of backgrounds and included 2 interaction designers, 2 engineers, 1 linguist, and 1 animator. All participants were involved in the 1:1 (1 human, 1 robot) observation study. Two participants (Participant 5 and 8) went on to be part of the 1:3 (1 human, 3 robots) observation study, the other four (Participant 6, 7, 9 and 10) were involved in the 2:3 (2 humans, 3 robots) observation study.

1:1 Interaction Observation Study The purpose of the first study was to observe the participants behaviour whilst interacting with a single robot. Each participant was given a 3-button synthesiser to communicate with the robot and allowed to interact for as long as they wished, i.e., no time limit was given.

1:3 Interaction Observation Study The second observational study involved each participant interacting with the group of 3 robots to examine how participants interacted with multiple creative agents at the same time and how the participants were influenced by the interactions between robots. This study involved 2 participants, both participants had previously completed the first observation study.

2:3 Interaction Observation Study The third observational study involved pairs of participants interacting with the system of 3 working robots. This study allowed for the participants to not only interact and observe the working system but to also interact with each other to share their experiences. This study involved 4 participants working in two groups of two. The 4 participants were chosen from those who completed the 1:1 study but were not involved in the 1:3 observation study.

Results

This section presents a brief summary of the observational studies, a more detail account can be found in Chee (2011).

1:1 Interaction The 1:1 interaction task allowed the participants to form individual theories on how single robots reacted to them, most learned that the robots did not respond to individual notes but sequences of them. Participants spent between 2 and 4 minutes interacting with the robot, much of that time was spent experimenting to determine how the robot reacted to different inputs: "[I] first tried to see how it would react, pressed a single button and then tried a sequence of notes" (Participant 6). Several of the participants learned to adopt a turn-taking behaviour with the robots, e.g., "when it started to play I stopped to watch, I only tried to play when it stopped" (Participant 5). Some of the participants interpreted the opening and closing of the hood as a cue for when they could play a song for the robot to learn, as Participant 9 commented: "I played a noise and it took that song and closed up and was like 'alright I'm gonna think of

something better'. It sounded like it was repeating what I did but like a bit different. Like it was working out what I'd done." Most of the participants assumed the role of teacher and attempted to get the robot to repeat a simple song. But in the case of Participant 8 the roles were reversed as the participant began copying the songs played by the robot.

1:3 Interaction For the 1:3 interaction studies the group of robots were placed on a table in a quiet location, as shown in Figure 4. The participants interacted with the group of robots for approximately 5 minutes. Both participants already knew the robots were responsive to them from the 1:1 study, but they found it difficult to determine which robot they were interacting with: "you knew you could interact but you were not really aware of the reaction as a group" (Participant 5). The participants noticed that the robots were different: "the green robots song was slightly different to blue and purple" (Participant 5); and, that they exhibited social behaviour amongst themselves: "Noticed they didn't rely just on the [synthesiser], the 3 of them were communicating. I thought they sang in a certain order as one started and the others would reply" (Participant 8). Both participants came to realise that system would continue to evolve new songs without their input and spent time towards the end of their sessions observing the group behaviour.

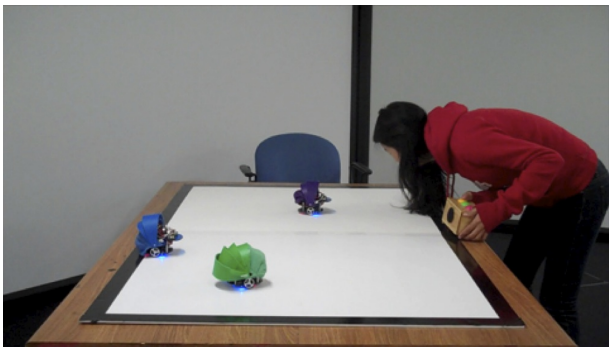


Figure 4: An example of the interaction in the 1:3 study.

2:3 Interaction Working together the participants in the third study quickly arrived at the conclusion that they needed to take turns in order to interact with the robots. Participant 6 saw that the robots moved towards Participant 7 and asked to be given one of the robots, Participant 7 replied "No, they have to go to you on their own", suggesting that Participant 7 recognised that the robots could not be commanded. Later, the participants became competitive in their attempts to attract the robots away from each other. As the participants shared observations about the system, they explored the transference of songs. By observing the interactions between Participant 7 and the robots, Participant 6 was able to determine that the robots responded to songs of exactly 8 notes and that the robots would repeat the song 3 times while it learned. At one point Participant 9 commented: "...when I pressed it like this 'beep beep beep beep' it went 'beep beep boop beep' so it was like changing what I played". These observations suggest that over time the participants were able to build relatively accurate 'mental' models of the processes of the robotic agents.

Figure 5: An example of the interaction in the 2:3 study.

Discussion

Unlike traditional interactive systems that react to human participants (Dezeuze 2010), the individual agents within artificial creative systems are continuously engaged in social interactions: the robots in our study would continue to interact and share songs without the intervention of the participants. While initially confusing, participants discovered through extended observation and interaction that they could inject songs into the society by teaching them to a single robot. Participants sometimes also assumed the role of learner and copied the songs of the robots and consequently adopted an interaction strategy more like that of a peer.

The autonomous nature of the embodied creative system runs counter to typical expectations of human-robot interactions; making interacting with a group of robots is significantly more difficult than interacting with one. The preliminary results presented here suggest that simple social policies in artificial creative systems, e.g., the turn-taking behaviour, coupled with cues that indicate state, e.g., closing the hood while practicing and composing songs, allow for conversational interactions to emerge over time.

Conclusion

The development of embodied creative system offers significant opportunities and challenges for researchers in computational creativity. This paper has presented a possible approach for the study of interaction design issues surrounding the development of artificial creative systems.

The Curious Whispers project explores the possibility of developing artificial creative systems that are open to these types of peer-to-peer interactions through the construction of a 'common ground' based on the expression and perception of artefacts. The research presented has shown that even a simple robotic platform can be designed to exploit its physical embodiment as well as its social situation, using easily obtained components.

The implemented system, while simple in terms of the computational ability of the agents, has provided a useful

platform for studying interactions between humans and artificial creative systems. The technical limitations of the robotic platform place an emphasis on the important role that communication plays in the evolution of creative systems, even with the restricted notion of what constitutes a 'song' in this initial exploration. Above all, the technology probe methodology used in our observational studies have illustrated the usefulness of implementing simple policies in artificial creative systems to allow human participants to adapt to the unusual interaction model.

Acknowledgements

The research reported in this paper was supported as part of the Bachelor of Design Computing Honours programme in the Faculty of Architecture, Design and Planning at the University of Sydney.

References

Bernsen, N., and Dybkjær, L. 2005. User interview-based progress evaluation of two successive conversational agent prototypes. In Maybury, M.; Stock, O.; and Wahlster, W., eds., *Intelligent Technologies for Interactive Entertainment*, volume 3814. Springer Berlin / Heidelberg. 220–224.

Biles, J. A. 1994. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, 131–137.

Boden, M. A. 2010. *Creativity and Art: Three Roads to Surprise*. Oxford: Oxford University Press.

Brooks, R. 1990. Elephants don't play chess. *Robotics and Autonomous Systems* 6:3–15.

Chee, E. 2011. Curious Whispers 2.0: Human-robot interaction with an embodied creative system. Honours Thesis, University of Sydney, Australia. Available online at `http://emmachee.com/Thesis/Emma_Chee_Thesis_2011.pdf`.

Clark, A. 1996. *Being There: Putting Brain, Body, and World Together Again*. Cambridge, MA, USA: MIT Press.

Dezeuze, A. 2010. *The 'do-it-yourself' artwork: Participation from the Fluxus of new media*. Manchester: Manchester University Press.

Gaye, L.; Mazé, R.; and Holmquist, L. E. 2003. Sonic city: the urban environment as a musical interface. In *Proceedings of the 2003 Conference on New interfaces For Musical Expression*, 109–115.

Gemeinboeck, P., and Saunders, R. 2011. Zwischenräume: The machine as voyeur. In *Proceedings of the First International Conference on Transdisciplinary Imaging at the Intersections between Art, Science and Culture*, 62—70.

Glanville, R. 1996. Robin mckinnon-wood and gordon pask: A lifelong conversation. *Journal of Cybernetics and Human Learning* 3(4).

Glynn, R. 2008. Performative Ecologies: Dancers, http://www.ruairiglynn.co.uk/portfolio/performative-ecologies/.

Haque, U. 2007. The architectural relevance of Gordon Pask. In *4d Social: Interactive Design Environments*. Wiley & Sons.

Hoffman, G., and Weinberg, G. 2010. Shimon: an interactive improvisational robotic marimba player. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, 3097–3102. New York, NY, USA: ACM.

Hutchinson, H.; Mackay, W.; Westerlund, B.; Bederson, B. B.; Druin, A.; Plaisant, C.; Beaudouin-Lafon, M.; Conversy, S.; Evans, H.; Hansen, H.; Roussel, N.; and Eiderbäck, B. 2003. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, 17–24. New York, NY, USA: ACM.

Kapur, A.; Eigenfeldt, A.; Bahn, C.; and Schloss, W. A. 2009. Collaborative composition for musical robots. *Journal of Science and Technology of the Arts* 1(1):48–52.

Mackay, W. 1990. *Users and Customizable Software: A Co-Adaptive Phenomenon*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Pease, A., and Colton, S. 2011. On impact and evaluation in computational creativity: A discussion of the turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy 2011*.

Penny, S. 1997. Embodied cultural agents: At the intersection of art, robotics, and cognitive science. In *Socially Intelligent Agents: Papers from the AAAI Fall Symposium*, 103–105. AAAI Press.

Pickering, J. 2005. Embodiment, constraint and the creative use of technology. In *Freedom and Constraint in the Creative Process in Digital Fine Art*.

Saunders, R., and Gero, J. S. 2002. How to study artificial creativity. In *Proceedings of Creativity and Cognition 4*.

Saunders, R.; Gemeinboeck, P.; Lombard, A.; Bourke, D.; and Kocabali, B. 2010. Curious whispers: An embodied artificial creative system. In *International Conference on Computational Creativity 2010, 7–9 January 2010*.

Straschnoy, A. 2008. The New Artist, http://www.the-new-artist.info/.

The role of motion dynamics in abstract painting

Alexander Schubert and Katja Mombaur
Interdisciplinary Center for Scientific Computing
University of Heidelberg
{alexander.schubert, katja.mombaur}@iwr.uni-heidelberg.de

Abstract

We investigate the role of dynamic motions performed by artists during the creative process of art generation. We are especially interested modern artworks inspired by the Action Painting style of Jackson Pollock.

Our aim is to evaluate and model the role of these motions in the process of art creation. We are using mathematical approaches from optimization and optimal control to capture the essence (cost functions of an optimal control problem) of these movements, study it and transfer it to feasible motions for a robot arm. Additionally, we performed studies of human responses to paintings assisted by an image analysis framework, which computes several image characteristics. We asked people to sort and cluster different action-painting images and performed PCA and Cluster Analysis in order to determine image traits that cause certain aesthetic experiences in contemplators.

By combining these approaches, we can develop a model that allows our robotic platform to monitor its painting process using a camera system and – based on an evaluation of its current status – to change its movement to create human-like paintings. This way, we enable the robot to paint in a human-like way without any further control from an operator.

Figure 1: An action painting in the style of Jackson Pollock, painted by "JacksonBot"

Introduction

The cognitive processes of generating and perceiving abstract art are – in contrast to figurative art – widely unknown. When processing representational art works, the effect of meaning is highly dominant. In abstract art, with the lack of this factor, the processes of perception are much more ambiguous, relying on a variety of more subtle qualities. In this work, we focus on the role of dynamic motions performed during the creation of an art work as one specific trait that influences our perception and aesthetic experience.

Action Paintings - Modern art works created by dynamic motions

The term "action painting" was first used in the essay "The American Action Painters" (Rosenberg 1952). While the term "action painting" is commonly used in public, art historians sometimes also use the term "Gestural Abstraction". Both terms emphasize the process of creating art, rather than the resulting art work, which reflects the key innovation that arose with this new form of painting in the 1940s to the 1960s. The style of painting includes dripping, dabbing and splashing paint on a canvas rather than being applied carefully and in a controlled way. Art encyclopedias describe these techniques as "depending on broad actions directed by the artist's sense of control interacting with chance or random occurrences." The artists often consider the physical act of painting itself as the essential aspect of the finished work. Regarding the contemplators, action paintings intend to connect to them on a subconscious level. In 1950, Pollock said "The unconscious is a very important side of modern art and I think the unconscious drives do mean a lot in looking at paintings"(Ross 1990) and later, he stated "We're all of us influenced by Freud, I guess I've been a Jungian for a long time"(Rodman 1961). Clearly, artists like Pollock do not think actively about dynamic motions performed by their bodies the way as mathematicians from the area of modeling and optimal control do. But for us, it is very exciting, that one of the main changes they applied to their painting style in order to achieve their aim of addressing the subconscious mind has been a shift in the manner they carry out their motions during the creational process.

Understanding the perception and generation of action paintings

Since a human possesses much more degrees of freedom than needed to move, human motions can often be seen as a superposition of goal directed motions and implicit, unconscious motions. The assumption, that elements of human motions can be described in this manner has been widely applied and verified, particularly in walking and running motions (Felis and Mombaur 2012),(Schultz and Mombaur 2010), but also (very recently) regarding emotional body language during human walking (Felis, Mombaur, and Berthoz 2012). If we transfer this approach to an artist, the goal-directed motions are those carried out to direct his hand (or rather a brush or tool) to the desired position, the implicit, unconscious motions are the result of an implicit solved optimal control problem with a certain cost function like maximizing stability or minimizing energy costs.

When looking at action paintings, we note, that this form of art generation is a very extreme form of this superposition model with a widely negligible goal-directed part. Therefore, it is a perfect basis to study the role of (unconscious) motion dynamics on a resulting art work. Jackson Pollock himself expressed similar thoughts when he said "The modern artist... is working and expressing an inner world – in other words – expressing the energy, the motion, and other inner forces" or "When you're working out of your unconscious, figures are bound to emerge... Painting is a state of being" (Rodman 1961).

However, the role of motion dynamics in the embodied expression of artists has been poorly described so far, supposedly due to the lack of an adequate method for the acquisition of quantitative data. The goal of our project is to use state-of-the-art tools from scientific computing to analyze the impact of motion dynamics both on the creational and perceptual side of action-painting art works. Therefore, we perform perception studies with contemplators and experimental studies concerning motion generation, which are linked by a robotic platform as a tool that can precisely reproduce different motion dynamics. Using this approach, we want to determine key motion types influencing a painting's perception.

Models of art perception

The perception of art, especially abstract art, is still an area of ongoing investigations. Therefore, no generally accepted theory including all facets of art perception exists. There are, however, different theories that can explain different aspects of art perception. One example of a theory of art perception is the one presented in (Leder et al. 2004) (see figure 2). In the past, resulting from an increasing interest in embodied cognition and embodied perception, there has been a stronger focus on the nature of human motion and its dynamics regarding neuroscience or rather neuroaesthetics as well as psychology and history of art. There are several results, showing that we perceive motion and actions with a strong involvement of those brain regions that are responsible for motion and action generation (Buccino et al. 2001). The mirror neurons located in these brain regions fire both,

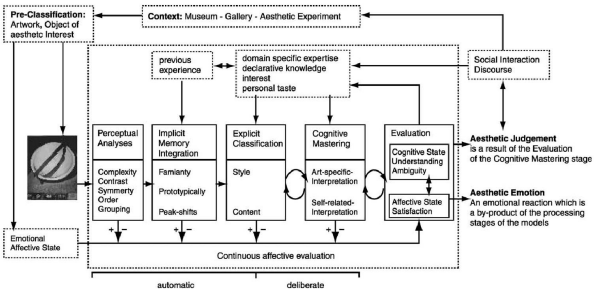


Figure 2: Overview of the aesthetic judgment model by (Leder et al. 2004)

when an action is actively performed and when the same action is being observed. These findings support the theory, that the neural representations for action perception and action production are identical (Buxbaum, Kyle, and Menon 2005). The relation between perception and embodied action simulation also exists for static scenes (Urgesi et al. 2006) and ranges even to the degree, where the motion is implied only by a static result of this very motion. For example, (Knoblich et al. 2002) showed, that the observation of a static graph sign evokes in the brain a motor simulation of the gesture, which is required to produce this graph sign. Finally, in (Freedberg and Gallese 2007), it was proposed that this effect of reconstructing motions by embodied simulation mechanisms will also be found when looking at "art works that are characterized by the particular gestural traces of the artist, as in Fontana and Pollock".

Mathematical background

To perform mathematical computations on motion dynamics, we first need to create models of a human and the robot arm. Both can be considered as systems of rigid bodies, which are connected by different types of joints (prismatic or revolute). By "model", we mean a mathematical description in terms of differential equations of the physical characteristics of the human arm an the robot accordingly. Depending on the number of bodies and joints, we end up with an certain number of degrees of freedom. For each body, we get a set of generalized variables q (coordinates), $\dot{q}$ (velocities), $\ddot{q}$ (accelerations), and τ (joint torques). Given such a model, we can fully describe its dynamics by means of

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \tag{1}$$

where $M(q)$ is the joint space inertia matrix and $N(q, \dot{q})$ contains the generalized non-linear effects. Once we have such a model, we can formulate our optimal control problem using $x = [q, \dot{q}]^T$ as states and $u = \tau$ as controls. The OCP

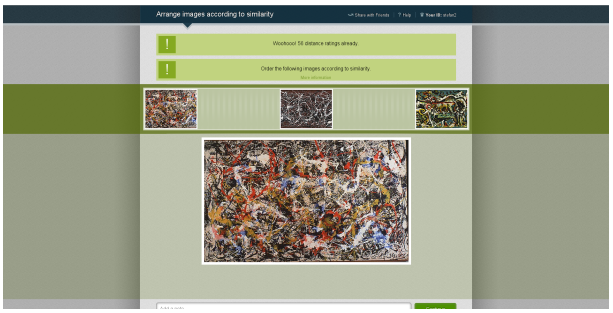


Figure 3: Interface for web-based similarity ratings

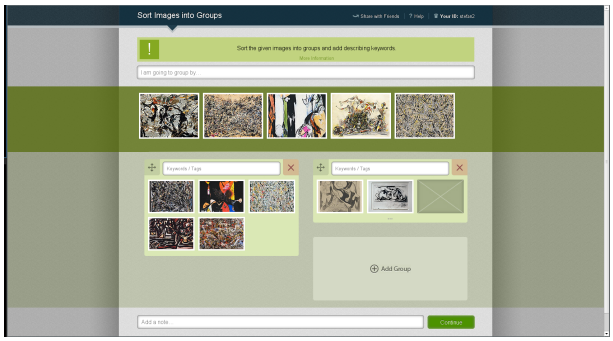


Figure 4: Interface for web-based sorting studies

can be written in its general form as:

$$\min_{x,u,T} \int_0^T L(t, x(t), u(t), p)dt + \Phi_M(T, x(T)) \tag{2}$$

subject to:

$$\dot{x} = f(t, x(t), u(t), p)$$
$$g(x(t), p) = 0$$
$$h(t, x(t), u(t), p) \geq 0$$

Note, that all the dynamic computation from our model is included in the RHS of the differential equation $\dot{x} = f(t, x(t), u(t), p)$. The first part of our objective function, $\int_0^T L(t, x(t), u(t), p)dt$ is called the Lagrange term, $\Phi_M(T, x(T))$ is called the Mayer term. The former is used to address objectives that have to be evaluated over the whole time horizon (such as minimizing jerk), the latter is used to address objectives that only need to be evaluated at the end of the time horizon (such as overall time). In our case, we will often only use the Lagrange term. To solve such a problem numerically, we apply a direct multiple shooting method which is implemented in the software package MUSCOD-II. For a more detailed description of the algorithm, see (Bock and J. 1984; Leineweber et al. 2003).

Experimental Data

Perception experiments

We performed two pre-studies to find out, whether human contemplators can distinguish robot paintings from human-made paintings and how they evaluate robot paintings created by different mathematical objective functions.

In the first study, we showed nine paintings to 29 participants, most of whom were laymen in arts and only vaguely familiar with Jackson Pollock. Seven paintings were original art works by Jackson Pollock and two paintings were generated by the robot platform JacksonBot. We asked the participants to judge, which of the paintings were original paintings by Pollock and which were not, but we intentionally did not inform them about the robotic background of the "fake" paintings. As might be expected, the original works by Pollock had a higher acceptance rate, but, very surprisingly, the difference between Pollock's and JacksonBot's paintings was not very high ($2.74 + / - 0.09$ vs. $2.85 + / - 0.76$, on a scale of 1 - 5).

In the second study, the participants were shown 10 paintings created solely by the robot platform, but with two opposite objective functions (maximum and minimum overall angular velocity in the robot arm) in the optimal control problem. The participants easily distinguished the two different painting styles.

Since the pre-studies were only conducted to get a rather rough idea on this aspect, we developed a more sophisticated web-based platform for further, more detailed investigations on this subject. The data obtained from this tool can be used to enhance the robot's ability to monitor its painting process.

The set of stimuli used for our studies consists of original action-art paintings by Pollock and other artists and images that were painted by our robot platform.

In the first task, contemplators are presented three randomly chosen paintings[1] and asked to arrange them on the screen according to their similarity (see figure 3). If they want, they are free to add a commentary to indicate their thoughts while arranging the paintings. As a result, we obtain for every set of two paintings a measure for their similarity in comparison with any other set of two paintings[2]. Using standard procedures from statistics like cluster analysis, we can determine which paintings are overall rated more "similar" than others.

In the second task, people are asked to perform a standard sorting study, i.e. they are asked to combine similar paintings in groups and to give some information on why they formed specific groups. The results of this task are used to validate the information obtained by the previous one and, additionally, they are used to gain more information about the attributes and traits, people seem to use while grouping. Therefore, the set of possible tags for the formed groups is limited and chosen by us. Is includes very basic image characteristics like colour as well as more interesting character-

[1] more precisely, the paintings are not chosen purely random but there is a slight correction to the probability of each painting to be presented in order to get many different correlations even when participants only complete few repetitions

[2] Note that we do not use the absolute values of "similarity" but quotients of these in order to avoid offset problems

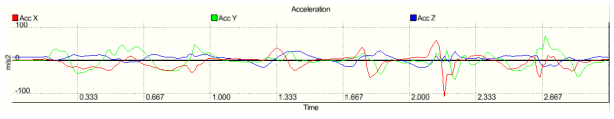


Figure 5: recorded acceleration data for a 3sec motion

istics like associated emotions.

Motion capture experiments

In order to study the way real human artists move during action-painting, we chose to do motion-capture studies with our collaborating artist. As a first approach, we used three inertia sensors to record dynamic data $D_{capture}$. For each of the three segments of the artist's arm (hand, lower arm, upper arm), we recorded accelerations, angular velocities and the rotation matrix[3] using three Xsens MTw inertial motion trackers. The sensors were placed directly above the calculated center of mass of each arm segment. Figure 5 shows an example of the raw data output obtained from the sensors.

We asked the artist to create different paintings and to describe her creative ideas as well as her thoughts and emotions during the process with her own words. That way, we can correlate identified objective functions with specific emotions or creative ideas.

Robot painting experiments

For first experiments, we created paintings with our robot platform. In order to compute the robot joint trajectories necessary to move along a desired end effector path, we use an optimal control based approach to solve the inverse kinematics problem. Using our first robotic platform, we created several paintings using different cost functions in the optimal control problem. Two of them – maximizing and minimizing the angular velocities in the robot joints – resulted in significantly different paintings. These paintings were used in the pre-study mentioned earlier.

Data Analysis

Motion reconstruction

To fit the record dynamic data $D_{capture}$ to our 9 DOF model of a human arm that is based on data from (De Leva 1996), we formulated an optimal control problem which generates the motion $x(t) = [q(t), \dot{q}(t)]^T$ and the controls $u(t) = \tau(t)$ that best fit the captured data with respect to the model dynamics f.

$$\min_{x,u} \frac{1}{2} ||D_{capture}(t) - D_{Simulated}(t)||_2^2 \qquad (3)$$

subject to:

$$\dot{x}(t) = f(t, x(t), u(t), p)$$
$$g(x, p) = 0$$
$$h(x, p) \geq 0$$

[3]recording the euler angles is not sufficient due to potential singularities in the reconstruction process

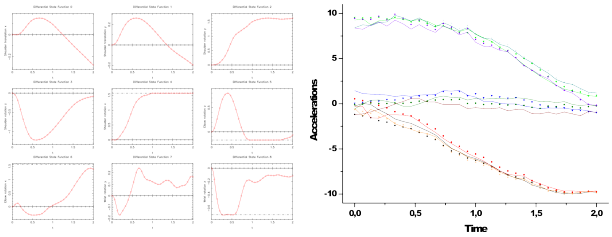


Figure 6: Computed trajectories for joint angles (left) and comparison of computed (lines) and measured (dots) accelerations (right).

The constraints in this case are given by the limited angles of the human arm joints and torque limitations of the arm muscles. The computed states and the fit quality of the acceleration data can bee seen in figure 6. Note that the angle approach to the joint limitations is plausible for this type of motion.

In the next step, we will use the motion capture data obtained from experiments with our collaborating artist not only reconstruct the motion, but use an inverse optimal control approach (like successfully used in a similar case in (Mombaur, Truong, and Laumond 2010)) to retrieve the underlying objective functions of these motions. To do so, we will use an approach developed by K.Hatz in (Hatz, Schlöder, and Bock 2012). This process is illustrated in figure 7.

Conclusion and Outlook

We introduced a new way to analyze the creative process of action painting by investigating the dynamic motions of artists. We developed a mathematical model, which we used to succesfully reconstructed an artists' action-painting-motions from inertia measurements. We used state-of-the-art optimal control techniques to create new action-painting-motions for a robotic platform and evaluated the resulting painting. Even with "artificial" objective functions, we were able to create action paintings that are indistinguishable from human-made action paintings for a human contemplator.

In the next step, we will use an inverse optimal control approach to go one step further from reconstructing an artist's motions to identifying the underlying objective functions of motion dynamics. That way, we will be able to generate specific painting motions corresponding to specific intentions as formulated by the artist.

Since several studies, e.g. (Haak et al. 2008), have shown that aesthetic experiences and judgments can – up to a certain degree – be explained by analyzing low-level image features, we chose to develop an image analysis software tool based on OpenCV that uses a variety of different filters and image processing tools that are related to aesthetic experience. Amongst other features, our tool analyzes the paintings considering its power spectrum, different symmetries, color and fractal analysis (Taylor, Micolich, and Jonas 1999). We will include the information obtained from our online perception studies in this tool and use it as feedback

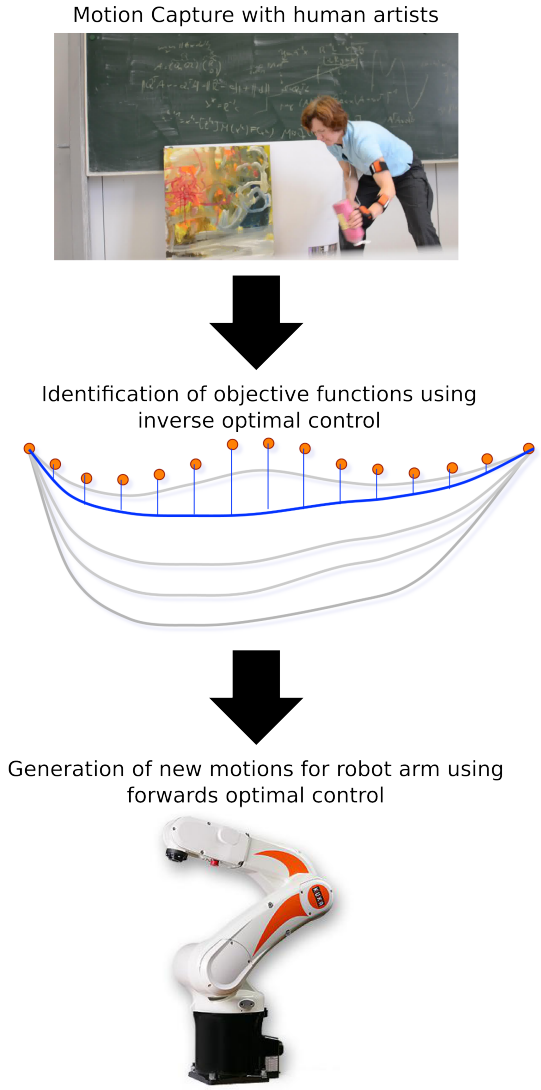


Figure 7: Transfer of human motion objectives to a robot platform (schematic overview)

for the robot platform. That way, we will enable it to paint autonomously with feedback only from an integrated camera monitoring the process.

The presented approach of capturing the essence of dynamic motions using inverse optimal control theory is not limited to the investigation of action paintings but can be used to analyze human motions in other art forms like dance or even in daily life by analyzing human gestures or full-body motions.

References

Bock, H.-G., and J., P. K. 1984. A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings 9th IFAC World Congress Budapest*.

Buccino, G.; Binkofski, F.; Fink, G. R.; Fadiga, L.; Fogassi, L.; Gallese, V.; Seitz, R. J.; Zilles, K.; Rizzolatti, G.; and Freund, H.-J. 2001. Action observation activates premotor and parietal areas in a somatotopic manner: an fmri study. *European Journal of Neuroscience* 13:400–404.

Buxbaum, L. J.; Kyle, K. M.; and Menon, R. 2005. On beyond mirror neurons: internal representations subserving imitation and recognition of skilled object-related actions in humans. *Brain research.Cognitive brain research* 25(1):226–239.

De Leva, P. 1996. Adjustments to zatsiorsky-seluyanovs segment inertia parameters. *Journal of Biomechanics* 29(9):1223–1230.

Felis, M., and Mombaur, K. 2012. Modeling and optimization of human walking. *to appear in Springer LNEE*.

Felis, M.; Mombaur, K.; and Berthoz, A. 2012. Mathematical modeling of emotional body language during human walking. *submitted to Proceedings of HPSC 2012*.

Freedberg, D., and Gallese, V. 2007. Motion, emotion and empathy in esthetic experience. *Trends in Cognitive Sciences* 11(5):197–203.

Haak, K.; Jacobs, R.; Thumfart, S.; Henson, B.; and Cornelissen, F. 2008. Aesthetics by numbers: computationally derived features of visual textures explain their aesthetics judgment. *Perception* 37.

Hatz, K.; Schlöder, J.; and Bock, H. G. 2012. Estimating parameters in optimal control problems. *SIAM J. Sci. Comput.* 34(3):A1707 – A1728.

Knoblich, G.; Seigerschmidt, E.; Flach, R.; and Prinz, W. 2002. Authorship effects in the prediction of handwriting strokes: evidence for action simulation during action perception. *Q J Exp Psychol A* 55(3):1027–46.

Leder, H.; Belke, B.; Oeberst, A.; and Augustin, D. 2004. A model of aesthetic appreciation and aesthetic judgments. *British Journal of Psychology* 95(4):489+.

Leineweber, D. B.; Bauer, I.; Bock, H.-G.; and Schlöder, J. P. 2003. An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization. part 1: theoretical aspects. *Computers & Chemical Engineering* 27(2):157–166.

Mombaur, K.; Truong, A.; and Laumond, J.-P. 2010. From human to humanoid locomotion – an inverse optimal control approach. *Autonomous Robots* 28(3):369–383.

Rodman, S. 1961. *Conversations with Artists*. Capricorn Books.

Rosenberg, H. 1952. The american action painters. *Art News* 51/8.

Ross, C. 1990. *Abstract expressionism: creators and critics: an anthology*. Abrams.

Schultz, G., and Mombaur, K. 2010. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics* 15(5):783–792.

Taylor, R. P.; Micolich, A. P.; and Jonas, D. 1999. Fractal analysis of Pollock's drip paintings. *Nature* 399(6735):422.

Urgesi, C.; Moro, V.; Candidi, M.; and Aglioti, S. 2006. Mapping implied body actions in the human motor system. *J Neurosci* 26(30):7942–9.

Creative Machine Performance: Computational Creativity and Robotic Art

Petra Gemeinboeck
College of Fine Art
University of NSW
NSW 2021 Australia
petra@unsw.edu.au

Rob Saunders
Design Lab
University of Sydney
NSW 2006 Australia
rob.saunders@sydney.edu.au

Abstract

The invention of machine performers has a long tradition as a method of philosophically probing the nature of creativity. Robotic art practices in the 20th Century have continued in this tradition, playfully engaging the public in questions of autonomy and agency. In this position paper, we explore the potential synergies between robotic art practice and computational creativity research through the development of robotic performances. This interdisciplinary approach permits the development of significantly new modes of interaction for robotic artworks, and potentially opens up computational models of creativity to rich social and cultural environments through interaction with audiences. We present our exploration of this potential with the development of Zwischenräume (In-between Spaces), an artwork that embeds curious robots into the walls of a gallery. The installation extends the traditional relationship between the audience and artwork such that visitors to the space become performers for the machine.

Introduction

This paper looks at potential synergies between the practice of robotic art and the study of computational creativity. Starting from the position that creativity and embodiment are critically linked, we argue that robotic art provides a rich experimental ground for applying models of creative agency within a public forum. From the robotic art perspective, a computational creativity approach expands the performative capacity of a robotic artwork by enhancing its potential to interact with its 'Umwelt' (Von Uexküll 1957).

In the 18th century, the Industrial Age brought with it a fascination with mechanical performers: Jacques de Vaucanson's Flute Player automaton and Baron Wolfgang von Kempelen's infamous chess playing Mechanical Turk clearly demonstrate a desire to create apparently creative automata. Through their work, both Vaucanson and von Kempelen engaged the public in philosophical questions about the nature of creativity, the possibilities of automation and, crucially, perfection.

Moving from mechanical to robotic machine performers, artists have deployed robotics to create apparently living and behaving creatures for over 40 years. The two dominant motivations for this creative practice have been to question "our premises in conceiving, building, and employing these electronic creatures" (Kac 2001), and to develop enhanced forms of interactions between machine actors and humans "via open, non-determined modes" (Reichle 2009).

The pioneering cybernetic work Senster by Edward Ihnatowicz, for example, exhibited life-like movements and was programmed to 'shy away' from loud noises. In contrast to the aforementioned automata, Ihnatowicz did not aim to conceal the Senster's inner workings, and yet "the public's response was to treat it as if it were a wild animal" (Rieser 2002). Norman White's Helpless Robot (1987–96) was a public sculpture, which asked for help to be moved, and when assisted, continued to make demands and increasingly abused its helpers (Kac 1997). Petit Mal by Simon Penny resembled a strange kind of bicycle and reacted to and pursued gallery visitors. With this work Penny aimed to explore the aesthetics of machines and their interactive behaviour in real world settings; Petit Mal was, in Penny's words, "an actor in social space" (Penny 2000). Ken Rinaldo's Autopoesis consisted of 15 robotic sculptures and evolved collective behavior based on their capability to sense each other's and the audience's presence (Huhtamo 2004). The installation Fish-Bird by Mari Velonaki comprised two robotic actors in the form of wheelchairs whose movements and written notes created a sense of persona. The relationship between the robot characters and the audience evolved based on autonomous movement, coordinated by a central controller, and what appeared to be personal, "handwritten" messages, printed by the robots (Rye et al. 2005).

Our fascination with producing artefacts that appear to be creative has created a rich history for researchers of computational creativity to draw upon. What we learn from these interdisciplinary artistic approaches is that, as performers, the artificial agents are embodied and situated in ways that can be socially accessed, shared and experienced by audiences. Likewise, embodied artificial agents gain access to shared social spaces with other creative agents, e.g., audience members.

The ability of robotic performers to interact with the audience not only relies on the robot's behaviours and responsiveness but also the embodiment and enactment of these behaviours. It can be argued that the performer is most successful if both embodiment and enactment reflect its perception of the world, that is, if it is capable of expressing and communicating its disposition. Looking at robotic art-

works that explore notions of autonomy and artificial creativity may thus offer starting points for thinking about social settings that involve humans interacting and collaborating with creative agents.

Our exploration revolves around the authors' collaboration to develop the robotic artwork Zwischenräume (In-between Spaces), a machine-augmented environment, for which we developed a practice embedding embodied curious agents into the walls of a gallery, turning them into a playground for open-ended exploration and transformation.

Zwischenräume

The installation Zwischenräume embeds autonomous robots into the architectural fabric of a gallery. The machine agents are encapsulated in the wall, sandwiched between the existing wall and a temporary wall that resembles it. At the beginning of an exhibition, the gallery space appears empty, presenting an apparently untouched familiar space. From the start, however, the robots' movements and persistent knockings suggest comprehensive machinery at work inside the wall. Over the course of the exhibition, the wall increasingly breaks open, and configurations of cracks and hole patterns mark the robots' ongoing sculpting activity (Figure 1).

Figure 1: Zwischenräume: curious robots transform our familiar environment.

The work uses robotics as a medium for intervention: it is not the spectacle of the robots that we are interested in, but rather the spectacle of the transformation of their environment. The starting point for this interdisciplinary collaboration was our common interest in the open-ended potential of creative machines to autonomously act within the human environment. From the computational creativity researcher's point of view, the embodied nature of the agents allowed for situating and studying the creative process within a complex material context. For the artist, this collaboration opened up the affective potential to materially intervene into our familiar environment, bringing about a strange force, seemingly with an agenda and beyond our control.

Each machine agent is equipped with a motorised hammer, chisel or punch, and a camera to interact and network with the other machines by re-sculpting its environment (Figure 2). The embodied agents are programmed to be curious, and as such intrinsically motivated to explore the environment. Once they have created large openings in the wall the robots may study the audience members as part of their environment. In the first version of this work, the robots used their hammer to both punch holes and for communicating amongst the collective. In a later version, we experimented with a more formal sculptural approach that used heuristic compositions of graffiti glyphs to perforate walls. Using the more stealthy movements of a chisel, the work responded to the specific urban setting of the gallery by adapting graffiti that covered the exterior of the building to become an inscription, pierced into the pristine interior walls of the gallery space (Figure 3). The final version of Zwischenräume used a punch to combine the force of the hammer and the precision of the chisel.

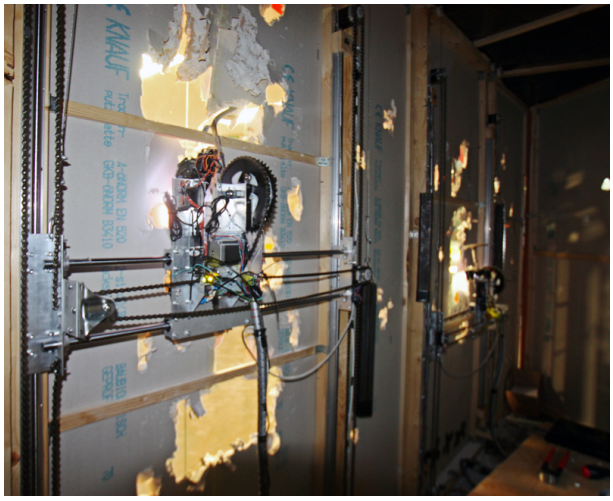


Figure 2: Robot gantries are attached to walls.

Similar to Jean Tinguely's kinetic sculptures (Hultén 1975), Zwischenräume's performance and what it produces may easily evoke a sense of dysfunctionality. As the machines' adaptive capability is driven by seemingly non-rational intentions rather than optimisation, the work, in some sense, subverts standard objectives for machine intelligence and notions of machine agency. Rather, it opens up the potential for imagining a machine that is 'free', a machine that is creative, see (Hultén 1987).

Machine Creativity

This section focuses on the development of the first version of Zwischenräume as depicted in Figures 1 and 2. Each robotic unit consisted of a carriage, mounted on a vertical gantry, equipped with a camera mounted on an articulated arm, a motorised hammer, and a contact microphone. The control system for the robots combined machine vision to detect features from the camera with audio processing to detect the knocking of other robots and computational models of intrinsic motivation based on unsupervised and reinforce-

Figure 3: Inscription of adapted graffiti glyphs.

ment machine learning to produce an adaptive, autonomous and self-directed agency.

The robot's vision system was developed to construct multiple models of the scene in front of the camera; using colour histograms to differentiate contexts, blob detection to detect individual shapes, and frame differencing to detect motion. Motion detection was only used to direct the attention of the vision system towards areas of possible interest within the field of view. Face detection is also used to recognise the presence of people to direct the attention of the robots towards visitors. While limited, these perceptual abilities provide sufficient richness for the learning algorithms to build models of the environment to determine what is different enough to be interesting.

Movements, shapes, sounds and colours are processed, learned and memorised, allowing each robotic agent to develop expectations of events in their surrounds. The machine learning techniques used in Zwischenräume combine unsupervised and reinforcement learning techniques (Russell and Norvig 2003): a self-organizing map (Kohonen 1984) is used to determine the similarity between images captured by the camera; Q-learning (Watkins 1989) is used to allow the robots to discover strategies for moving about the wall, using the hammer and positioning the camera.

Separate models are constructed for colours and shapes in images. To determine the novelty of a context, sparse histograms are constructed from captured images based on only 32 colour bins with a high threshold, so only the most significant colours are represented and compared using a self-organising map. Blob detection in low-resolution (32x32 pixel) images, relative to a typical model image of the wall, is used to discover novel shapes and encoded in a self-organising map as a binary vector. In both cases, the difference between known prototypes in the self-organising map provide a measure of novelty (Saunders 2001).

Reinforcement learning is used to learn the consequences of movements within the visual field of the camera. Error in prediction between learned models of consequences and observed results is used as a measure of surprise. As a result system that is able to learn a small repertoire of skills and appreciate the novelty of their results, e.g., knocking on wood does not produce dents. This ability is limited to immediate consequences of actions and does not current extend to sequences of actions.

The goal of the learning system is to maximise an internally generated reward for capturing 'interesting' images and to develop a policy for generating rewards through action. Interest is calculated based on a computational model that captures intuitive notions of novelty and surprise (Saunders 2001): 'novelty' is defined as a difference between an image and all previous images taken by the robot, e.g., the discovery of significant new colours or shapes; and, 'surprise' is defined as the unexpectedness of an image within a known situation, e.g., relative to a learned landmark or after having taken an action within an expected outcome (Berlyne 1960). Learning plays a critical role in both the assessment of novelty and surprise. In novelty, the robots have to learn suitably general prototypes for the different types of images that they encounter. In surprise, the 'situation' against which images are judged includes a learned model of the consequences of actions (Clancey 1997).

Consequently, intrinsic motivation to learn directs both the robot's gaze and its actions, resulting in a feedback process that increases the complexity of the environment – through the robot's knocking – relative to the perceptual abilities of the agent. Sequences of knocking actions are developed, such that the robots develop a repertoire of actions that produce significant perceived changes in terms of colour, shapes and motion. In this way, the robots explore their creative potential in re-sculpting their environment. Figure 4 presents a collage of images taken by a single robot when it discovered something 'interesting', illustrating how the evaluation of 'interesting' evolved for this robot; it shows how the agent's interest is affected by: (a) positioning of the camera, e.g., the discovery of lettering on the plasterboard wall; (b) use of the hammer, e.g., the production of dents and holes; and, (c) interaction of visitors.

Figure 4: Robot captures, showing the evolution of interesting changes in the environment.

Discussion

The robots' creative process turns the wall into a playful environment for learning, similar to a sandpit; while from the audiences' point of view, the wall is turned into a performance stage. This opens up a scenario of encounter for studying the potential of computational creativity and the role of embodiment. Following Pickering (2005), we argue that creativity cannot be properly understood, or modelled, without an account of how it emerges from the encounter between the world and intrinsically active, exploratory and productively playful agents.

Embodiment and Creativity

The agents' embodiment provides opportunities to expand their behavioural range by taking advantage of properties of the physical environment that would be difficult or impossible to simulate computationally (Brooks 1990). In Zwischenräume the machines' creative agency is not predetermined but evolves based on what happens in the environment they examine and manipulate. As the agents' embodiment evolves based on its interaction with the environment, the robots' creative agency affects processes out of which it itself is emergent.

This resonates with Barad's argument that 'agency is a matter of intra-acting: it is an enactment, not something that someone or something has' (Barad 2007). It also evokes Maturana and Varela's notion of enaction, where the act of bringing about a world occurs through the 'structural coupling' between the dynamical environment and the autonomous agents (Maturana and Varela 1987). While the machines perturb and eventually threaten the wall's structural integrity, they adapt to their changing environment, the destruction of the wall and how it changes their perception of the world outside.

The connection to creativity is two-fold: Firstly, the robots' intrinsic motivation to explore, discover and constantly produce novel changes to their environment demonstrates a simplistic level of a creative process itself, akin to the act of doodling, where the motivation is a reflective exploration of possibilities rather than purposeful communication with others. Secondly, the audiences interpret the machines' interactions based on their own context, producing a number of possible meaningful relations and associations. The agents' embodiment and situatedness becomes a portal for entering the human world, creating meaning. The agents' enacted perception also provides a window on the agents' viewpoint, thus possibly changing the perspective of the audience.

Furthermore, an enactive approach (Barad 2003; Clark 1998; Thompson 2005) opens up alternative ways of thinking about creative human-machine collaborations. It makes possible a re-thinking of human-machine creativity beyond the polarisation of human and non-human, one that promotes shared or distributed agency within the creative act.

Audience Participation

Autonomous, creative machine performances challenge the most common interaction paradigm of primarily reacting to what is sensed, often according to a pre-mapped narrative. Zwischenräume's curious agents proactively seek interaction, rather than purely responding to changes in the surrounds. Once the robots have opened up the wall, the appearance and behaviours of audience members are perceived by the system as changes in their environment and become an integral part of the agents' intrinsic motivation system.

The agents' behaviours adapt based on their perception and evaluation of their environment, including the audience, as either interesting or boring. A curious machine performer whose behaviors are motivated by what it perceives and expects can be thought of as an audience to the audiences performance. Thus, in Zwischenräume it is not only the robots that perform, but also the audience that provokes, entertains and rewards the machines' curiosity. This notion of audience participation expands common interaction paradigms in interactive art and media environments (Paul 2003). The robots don't only respond or adapt to the audience's presence and behaviours, but also have the capacity to perceive the audience with a curious disposition.

By turning around the traditional relationship between audiences and machinic performers, the use of curious robotic performers permits a re-examination of the machine spectacle. Lazardig (2008) argues that spectacle, as "a performance aimed at an audience," was central to the conception of the machine in the 17th century as a means of projecting a perception of utility; allowing the machine to become "an object of admiration and therefore guaranteed to 'function"'. Kinetic sculptures and robotic artworks exploit and promote the power of the spectacle in their relationship with the audience. This is also the case in Zwischenräume however, it is not only the machines that are the spectacle for the audience but also the audience that becomes an 'object of curiosity' for the machines (Figure 5). Thus the relationship with a curious robot extends the notion of the spectacle, and, in a way, brings it full circle.

Figure 5: Gallery visitor captured by one of the robots' cameras as he performs for the robotic wall.

Concluding Remarks

A significant aspect of Zwischenräume's specific embodiment is that it embeds the creative agents in our familiar (human) environment. This allowed us to direct both our, and the audience's, attention to the autonomous process and creative agency, rather than the spectacle of the machine. The integration of computational models of creativity into this artwork extended the range of open-ended, non-determined modes of interaction with the existing environment, as well as between the artwork and the audience.

We argue that it is both, the embodied nature of the agents and their autonomous creative capacity that allows for novel meaningful interactions and relationships between the artwork and the audience. The importance of embodiment for computational creativity can also be seen in the improvising robotic marimba player Shimon, which uses a physical gesture framework to enhance synchronised musical improvisation between human and nonhuman musicians (Hoffmann and Weinberg 2011). The robot player's movements not only produce sounds but also play a significant role in performing visually and communicatively with the other (human) band members as well as the audience.

Embodying creative agents and embedding them in our everyday or public environment is often messier and more ambiguous than purely computational simulation. What we gain, however, is not only a new shared embodied space for audience experience but also a new experimentation space for shared (human and non-human) creativity.

Acknowledgements

This research has been supported by an Australia Research Council Grant, a New Work Grant from the Austrian Federal Ministry for Education, Arts and Culture, and a Faculty Research Grant from COFA (University of NSW).

References

Barad, K. 2003. Posthumanist performativity: Toward an understanding of how matter comes to matter. *Signs: Journal of Women in Culture and Society* 23(1):801–831.

Barad, K. 2007. *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Durham, NC: Duke University Press.

Berlyne, D. E. 1960. *Conflict, Arousal, and Curiosity*. New York, NY: McGraw Hill.

Brooks, R. 1990. Elephants don't play chess. *Robotics and Autonomous Systems* 6:3–15.

Clancey, W. J. 1997. *Situated Cognition: On Human Knowledge and Computer Representations*. Cambridge, England: Cambridge University Press.

Clark, A. 1998. Where brain, body and world collide. *Daedalus: Journal of the American Academy of Arts and Sciences* 127(2):257—280.

Hoffmann, G., and Weinberg, G. 2011. Interactive improvisation with a robotic marimba player. In *Autonomous Robots 31*, 133–153. Springer.

Huhtamo, E. 2004. Trouble at the interface, or the identity crisis of interactive art. *Framework: The Finnish Art Review* (2):38–41.

Hultén, P. 1975. *Tinguely. 'Méta'*. London, UK: Thames & Hudson.

Hultén, P. 1987. *Jean Tinguely. A Magic Stronger than Death*. New York, NY: Abbeville Press.

Kac, E. 1997. Foundation and development of robotic art. *Art Journal* 56(3):60–67.

Kac, E. 2001. Towards a chronology of robotic art. *Convergence: The Journal of Research into New Media Technologies* 7(1).

Kohonen, T. 1984. *Self-Organization and Associative Memory*. Berlin: Springer.

Lazardig, J. 2008. The machine as spectacle: Function and admiration in seventeenth-century perspectives on machines. In de Gryter, W., ed., *Instruments in art and science: on the architectonics of cultural boundaries in the 17th century*. 152–175.

Maturana, H., and Varela, F. 1987. *The Tree of Knowledge: The biological roots of human understanding*. Boston, MA: Shambhala Publications.

Paul, C. 2003. *Digital Art*. London, UK: Thames & Hudson.

Penny, S. 2000. Agents as artworks and agent design as artistic practice. In Dautenhahn, K., ed., *Human Cognition and Social Agent Technology*. John Benjamins Publishing Co. 395–414.

Pickering, J. 2005. Embodiment, constraint and the creative use of technology. In *Freedom and Constraint in the Creative Process in Digital Fine Art*.

Reichle, I. 2009. *Art in the Age of Technoscience: Genetic Engineering, Robotics, and Artificial Life in Contemporary Art*. Wien: Springer.

Rieser, M. 2002. The art of interactivity: from gallery to street. In Mealing, S., ed., *Computers and Art*. Bristol, UK: Intellect. 81–96.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. New Jersey, NY: Prentice Hall.

Rye, D.; Velonaki, M.; Williams, S.; and Scheding, S. 2005. Fish-bird: Human-robot interaction in a contemporary arts setting. In *Proceedings of the 2005 Australasian Conference on Robotics and Automation*.

Saunders, R. 2001. *Curious Design Agents and Artificial Creativity*. Ph.d. thesis, Faculty of Architecture, The University of Sydney.

Thompson, E. 2005. Sensorimotor subjectivity and the enactive approach to experience. *Phenomenology and the Cognitive Sciences* 4(4):407–427.

Von Uexküll, J. 1957. A stroll through the worlds of animals and men: A picture book of invisible worlds. In Schiller, C., ed., *Instinctive Behavior: The Development of a Modern Concept*. New York, NY: Int'l Universities Press. 5–80.

Watkins, C. 1989. *Learning from Delayed Rewards*. Phd thesis, Cambridge University, Cambridge, England.

An Artificial Intelligence System to Mediate the Creation of Sound and Light Environments

Claudio Benghi
Northumbria University,
Ellison Building,
Newcastle upon Tyne, NE1 8ST, England
claudio.benghi@northumbria.ac.uk

Gloria Ronchi
Aether & Hemera,
Kingsland Studios, Priory Green,
Newcastle upon Tyne, NE6 2DW, England
hemera@aether-hemera.com

Introduction

This demonstration presents the IT elements of an art installation that exhibits intelligent reactive behaviours to participant input employing Artificial Intelligence (AI) techniques to create unique aesthetic interactions.
The audience is invited to speak into a set of microphones; the system captures all the sounds performed and uses them to seed an AI engine for creating a new soundscape in real time, on the base of a custom music knowledge repository. The compositions is played back to the users through surrounding speakers and accompanied with synchronised light events of an array of coloured LEDs.
This art work allows viewers to become active participants in creating multisensory computer-mediated experiences, with the aim of investigating the potential for creative forms of inter-authorship.

Software Application

The installation's software has been built as a custom event manager developed under the .Net framework that can respond to events from the users, timers, and the UI cascading them through the required algorithms and libraries as a function of specified interaction settings; this solution allowed swift changes to the behaviour of the artwork in response to the observation of audience interaction patterns.

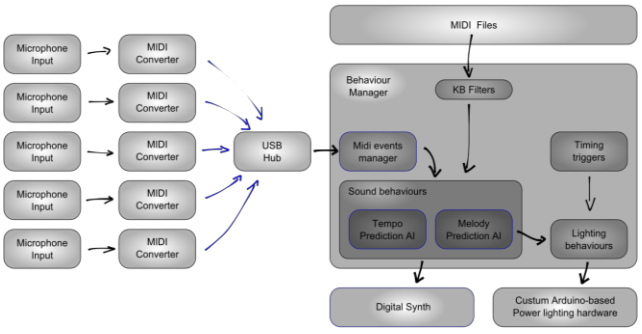


Figure 1: Scheme of the modular architecture of the system

Different portions of the data flow have been externalised to custom hardware to reduce computational load on the controlling computer: a configurable number of real-time devices converters transform the sounds of the required number of microphones into MIDI messages and channel them to the event manager; a cascade of Arduino devices control the custom multi channel lighting controllers and the sound output stage relies on MIDI standards.

A substantial amount of work has been put into the optimisation of the UI console controlling the behaviour of the installation; this turned out to be crucial for the success of the project as it allowed to make use of the important feedback gathered in the first implementation of this participatory art work.

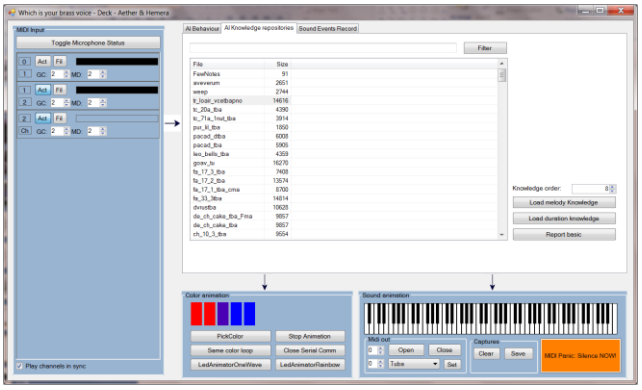


Figure 2: GUI of the controlling system

The work was first displayed as part of a public event over three weeks and allowed the co-generation of unpredictable soundscapes with varying levels of user's appreciation.
The evaluation of any public co-creation environment is itself a challenging research area and our future work will investigate and evaluate methodologies to do so; further developments to the AI are also planned to include feedback from past visitors.

More information about this project can be found at:
http://www.aether-hemera.com/s/aib

Controlling Interactive Music Performance (CIM)

Andrew Brown, Toby Gifford and Bradley Voltz
Queensland Conservatorium of Music, Griffith University
andrew.r.brown@griffith.edu.au, t.gifford@griffith.edu.au, b.voltz@griffith.edu.au

Abstract

Controlling Interactive Music (CIM) is an interactive music system for human-computer duets. Designed as a creativity support system it explores the metaphor of human-machine symbiosis, where the phenomenological experience of interacting with CIM has both a degree of instrumentality and a sense of partnership. Building on Pachet's (2006) notion of reflexivity, Young's (2009) explorations of conversational interaction protocols, and Whalley's (2012) experiments in networked human-computer music interaction, as well as our own previous work in interactive music systems (Gifford & Brown 2011), CIM applies an activity/relationality/prominence based model of musical duet interaction. Evaluation of the system from both audience and performer perspectives yielded consensus views that interacting with CIM evokes a sense of agency, stimulates creativity, and is engaging.

Description

The CIM system is an interactive music system for use in human-machine creative partnerships. It is designed to sit at a mid-point of the autonomy spectrum, according to Rowe's instrument paradigm vs player paradigm continuum. CIM accepts MIDI input from a human performer, and improvises musical accompaniment.

CIM's behaviour is directed by our model of duet interaction, which utilises various conversational, contrapuntal and accompaniment metaphors to determine appropriate musical behaviour. An important facet of this duet model is the notion of turn-taking – where the system and the human swap roles as the musical initiator.

To facilitate turn-taking, the system includes some mechanisms for detecting musical phrases, and their completion. This way the system can change roles at musically appropriate times. Our early implementation of this system simply listened for periods of silence as a cue that the human performer had finished a phrase. Whilst this method is efficient and robust, it limits duet interaction and leads to a discontinuous musical result.

This behaviour, whilst imbuing CIM with a sense of autonomy and independence, detracts from ensemble unity and interrupts musical flow. To address this deficiency, we implemented some enchronic segmentation measures, allowing for inter-part elision. Inter-part elision is where phrase-end in one voice coincides with (or is anticipated by) phrase-start in a second voice.

In order to allow for inter-part elision, opportunistic decision making, and other synchronous devices for enhancing musical flow, we have implemented some measures of musical closure as secondary segmentation indicators. Additionally these measures guide CIM's own output, facilitating generation of coherent phrase structure.

Figure 1: A musician interacting with the CIM system

References

Gifford T & Brown A R (2011). Beyond Reflexivity: Mediating between imitative and intelligent action in an interactive music system. In: *Proceedings of the British Computer Society Human-Computer Interaction Conference 2011*, Newcastle Upon Tyne.

Pachet F (2006). Enhancing individual creativity with interactive musical reflective systems. In *Musical Creativity: Current Research in Theory and Practice*. Wiggins G & Deliege I (eds) London: Psychology Press

Whalley I (2012). Internet2 and global electroacoustic music: Navigating a decision space of production, relationships and languages. *Organised Sound* **17**(01):4-15

Young M (2009). Creative Computers, Improvisation and Intimacy. In Boden M et al (eds) Dagstuhl Seminar Proceedings 09291: *Computational Creativity: An Interdisciplinary Approach*

Towards a Flowcharting System for Automated Process Invention

Simon Colton and John Charnley

Computational Creativity Group, Department of Computing, Goldsmiths, University of London
www.doc.gold.ac.uk/ccg

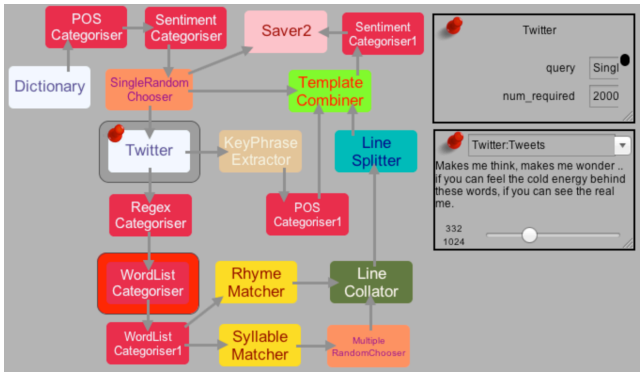


Figure 1: User-defined flowchart for poetry generation.

Flowcharts

Ironically, while automated programming has had a long and varied history in Artificial Intelligence research, automating the creative art of programming has rarely been studied within Computational Creativity research. In many senses, software writing software represents a very exciting potential avenue for research, as it addresses directly issues related to novelty, surprise, innovation at process level and the framing of activities. One reason for the lack of research in this area is the difficulty inherent in getting software to generate code. Therefore, it seems sensible to start investigating how software can innovate at the process level with an approach less than full programming, and we have chosen the classic approach to process design afforded by flowcharts. Our aim is to provide a system simple enough to be used by non-experts to craft generative flowcharts, indeed, simple enough for the software itself to create flowcharts which represent novel, and hopefully interesting new processes.

We are currently in the fourth iteration of development, having found various difficulties with three previous approaches, ranging from flexibility and expressiveness of the flowcharts to the mismatching of inputs with outputs, the storage of data between runs, and the ability to handle programmatic constructs such as conditionals and loops. In our current approach, we represent a process as a script, onto which a flowchart can be grafted. We believe this offers the best balance of flexibility, expressiveness and usability, and will pave the way to the automatic generation of scripts in the next development stage. We have so far implemented the natural language processing flowchart nodes required to model aspects of a previous poetry generation approach and a previous concept formation approach.

The Flow System

In figure 1 we present a screenshot of the system, which is tentatively called *Flow*. The flowchart shown uses 18 sub-processes which, in overview, do the following: a negative valence adjective is chosen, and used to retrieve tweets from Twitter; these are then filtered to remove various types, and pairs are matched by syllable count and rhyme; finally the lines are split where possible and combined via a template into poems of four stanzas; multiple poems are produced and the one with overall most negative valency is saved. A stanza from a poem generated using 'malevolent' is given in figure 2. Note in figure 1 that the node bordered in red (WordList Categoriser) contains the sub-process currently running, and the node bordered in grey (Twitter) has been clicked by the user, which brings up the parameters for that sub-process in the first black-bordered box and the output from it in the second black-bordered box. We see the 332nd of 1024 tweets containing the word 'cold' is on view. Note also that the user is able to put a thumb-pin into any node, which indicates that the previous output from that node should be used in the next run, rather than being calculated again.

It's our ambition to build a community of open-source developers and users around the Flow approach, so that the system can mimic the capabilities of existing generative systems in various domains, but more importantly, it can invent new processes in those domains. Moreover, we plan to install the system on various servers worldwide, constantly reacting in creative ways to new nodes which are uploaded by developers, and to new flowcharts developed by users with a variety of cultural backgrounds. We hope to show that, in addition to creating at artefact level, software can innovate at process level, test the value of new processes and intelligently frame how they work and what they produce.

I hear the souls of the damned wailing in hell.
I feel a malevolent spectre hovering just behind me.
It must be his birthday.
Is God willing to prevent evil, but not able?
Then he is not omnipotent.
Is he able, but not willing?
Then he is malevolent.
It's only when his intelligence grows and
He understands the laws of man that
He becomes malevolent and violent.
I don't find it malevolent, I find it affectionate.
Geeks do weird things and that can be hilarious for different reasons.

Figure 2: A stanza from the poem *On Being Malevolent*.

A Rogue Dream: Web-Driven Theme Generation for Games

Michael Cook
Computational Creativity Group
Imperial College, London
mtc06@doc.ic.ac.uk

ABSTRACT

A Rogue Dream is an experimental videogame developed in seven days for a roguelike development challenge. It uses techniques from computational creativity papers to attempt to theme a game dynamically using a source noun from the player, including generating images and theme information. The game is part of exploratory research into bridging the gap between generating rules-based content and theme content for videogames.

1. DOWNLOAD

While A Rogue Dream is not available to download directly, its code can be found at:

https://github.com/cutgarnetgames/roguedream

Spritely, a tool used in A Rogue Dream, can also be downloaded from:

https://github.com/gamesbyangelina/spritely

2. BACKGROUND

Procedural content generation systems mostly focus on generating structural details of a game, or arranging pre-existing contextual information (such as choosing a noun from a list of pre-approved words). This is because the relationship between the mechanics of a game and its theme is hard to define and has not been approached from a computational perspective.

For instance, in Super Mario eating a mushroom increases the player's power. We understand that food makes people stronger, therefore a mushroom is contextually appropriate. In order to procedurally replace that with another object, the system must understand the real-world concepts of food, strength, size and change. Most content generation systems for games are designed to understand games, not the real world. How can we overcome that?

3. A ROGUE DREAM

In [1] Tony Veale proposes mining Google Autocomplete using leading phrases such as "why do <*keyword*>s..." and using the autocompletions as a source of general knowledge

Figure 1: A screenshot from A Rogue Dream. The input was 'cow' - enemies were 'red', resulting in a red shoe being the enemy sprite. Abilities including 'mooing' and 'giving milk'.

or stereotypes. We refer to this as 'cold reading the Internet', and use it extensively in A Rogue Dream. We also employ *Spritely*, a tool for automatically generating sprite-based artwork by mining the web for images.

The game begins by asking the player to complete the sentence "*Last night, I dreamt I was a...*". The noun used to complete the sentence becomes a parameter for the search systems in A Rogue Dream, such as Spritely and the various text retrieval systems based on Veale's cold reading. These are subject to further filtering - queries matching "why do <keyword>s **hate**..." are used to label enemies, for example.

This work connects to other research being conducted by the author currently in direct code modification for content generation [**?**]. We hope to combine these two research tracks in order to build technology that can understand and situate abstract game concepts in a real-world context, and provide labels and fiction that describe and illustrate the game world accurately and in a thematically appropriate way.

4. REFERENCES

[1] Tony Veale. From conceptual 'mash-ups' to 'bad-ass' blends: A robust computational model of conceptual blending. In *Proceedings of the 3rd International Conference on Computational Creativity*, 2012.

A Puzzling Present: Code Modification for Game Mechanic Design

Michael Cook and Simon Colton
Computational Creativity Group
Imperial College, London
{mtc06,sgc}@doc.ic.ac.uk

Figure 1: A screenshot from *A Puzzling Present*.

ABSTRACT

A Puzzling Present is an Android and Desktop game released in December 2012. The game mechanics (that is, the player's abilities) as well as the level designs were generated using Mechanic Miner, a procedural content generator that is capable of exploring, modifying and executing codebases to create game content. It is the first game developed using direct code modification as a means of procedural mechanic generation.

1. DOWNLOAD

A Puzzling Present is available on Android and for all desktop operating systems, for free, here:

http://www.gamesbyangelina.org/downloads/app.html

The source code is also available on gamesbyangelina.org.

2. BACKGROUND

Mechanic Miner was developed as part of PhD research into automating the game design process, through a piece of software called ANGELINA. ANGELINA's ability to develop small games autonomously, including theming the game's content using social and web media, was demonstrated at ICCC 2012[1]. Mechanic Miner represents a large step forward for ANGELINA as the system becomes able to inspect and modify code directly, instead of using grammars or other intermediate representations.

ANGELINA's research has always aimed to produce playable games for general release. *Space Station Invaders* was released in early 2012 as a commission for the New Scientist, and a series of newsgames were released to coincide with several conferences in mid-2012. A Puzzling Present was the largest release to date, garnering over 6000 downloads, and entering the Android New Game charts in December, as well as coverage on Ars Technica, The New Scientist, and Phys.org.

3. A PUZZLING PRESENT

The game itself contains thirty levels split into three sets of ten. Each set of levels, or *world*, has a unique power available to the player, such as inverting gravity or becoming bouncy. These powers can be switched on and off, and must be used to complete each level. Each power was discovered by Mechanic Miner by iterative modification of code and simulation of gameplay to test the code modifications. For more information on the system, see [2].

Levels were designed using the same system - mechanics are tested against designed levels to evaluate whether the level is appropriate. This means the system is capable of designing novel levels with mechanics it has never seen before - there is no human intervention to add heuristics or evaluations for specific mechanics.

We are currently working on integrating Mechanic Miner into the newsgame generation module of ANGELINA, so that the two systems can work together to collaboratively build larger games. This initial work on code modification has also opened up major questions about the relationship between code and meaning in videogames, which we plan to explore in future work.

4. REFERENCES

[1] Michael Cook and Simon Colton. Angelina - coevolution in automated game design. In *Proceedings of the 3rd International Conference on Computational Creativity*, 2012.

[2] Michael Cook, Simon Colton, and Jeremy Gow. Nobody's a critic: On the evaluation of creative code generators. In *Proceedings of the 4th International Conference on Computational Creativity*, 2013.

Demonstration: A meta-pianist serial music comproviser

Roger T. Dean
austraLYSIS, Sydney; and MARCS Institute,
University of Western Sydney, Australia
roger.dean@uws.edu.au

Computational processes which produce meta-human as well as seemingly-human outputs are of interest. Such outputs may become apparently human as they become familiar. So I write algorithmic interfaces (often in MAXMSPJitter) for real-time performative generation of complex musical/visual features, to be part of compositions or improvisations. Here I demonstrate a musical system to generate serial 12-tone rows, their standard transforms, and then to assemble them into melodic sequences, or into two part meta-pianistic performances.

Serial rigour of pitch construction is maintained throughout. This means here that 12-note motives are made, each of which comprises all the pitches within an octave on the piano (an octave comprises a doubling of frequency of the sound, and notes at the start and end of this sequence are given the same note name CDEFGABC etc). Then a generative system creates a rigorous set of transforms of the chosen note sequences. But as in serial composition at large, when these are disposed amongst multiple voices, and to create harmonies (simultaneous notes) as well as melodies (successions of separated notes), the serial chronology is modified. Furthermore, the system allows asynchronous processing of several versions of the original series, or of several different series.

A range of complexity can result, and to enhance this I also made a companion system which uses tonal major scale melodies in a similar way. Here the original (Prime) version consists only of 12 notes taken from within an octave of the major scale (which includes only 7 rather than 12 pitches), thus permitting some repetitions. Chromatic inversion is used, so that for example, the scale of Cmajor ascending from C becomes the scale of Ab major descending from C, and major tonality with change of key centre is preserved.

The performance patch within the system provided a default stochastic rhythmic, chordal and intensity control process; all of whose features are open to real-time control by the user.The patches are used for generating components of electroacoustic or notated composition, normally with equal-tempered or alternative tuning systems performed on a physical synthesis virtual piano (PianoTeq); and also within live solo MultiPiano performances involving acoustic piano and electronics.

The outputs are meta-human in at least two senses. First, as with many computer patches, the physical limitations of playing an instrument do not apply, and Xenakian performance complexities can be realised. Second, no human improviser could achieve this precision of pitch transformation; rather we have evidence they tend to take a simplified approach to atonality, usually focusing on controlling intervals of 1, 2, 6, and 11 semitones. The products of these patches are also in use in experiments on the psychology of expectation (collaboration with Freya Bailes, Marcus Pearce and Geraint Wiggins, UK).

References

MultiPiano, by Roger Dean; Tall Poppies TP225, Double CD (2012).

assimilate - collaborative narrative construction

Damian Hills
Creativity and Cognition Studio
University of Technology, Sydney
Sydney, Australia
Damian.Hills@uts.edu.au

Abstract

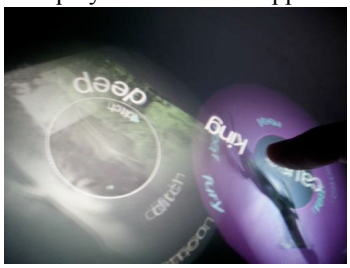
This demonstration presents the 'assimilate - collaborative narrative construction' project, that aims for a holistic system design with support for the creative possibilities of collaborative narrative construction.

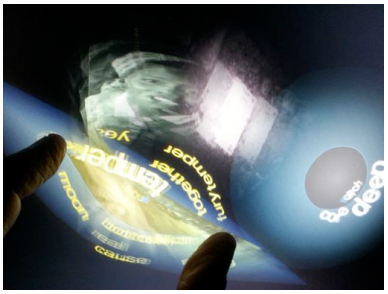
Introduction

This demonstration presents the 'assimilate - collaborative narrative construction' project (Hills 2011) that aims for a holistic system design with support for the creative possibilities of collaborative narrative construction. By incorporating interface mechanics with a flexible model of narrative template representation, the system design emphasises how mental models and intentions are understood by participants, and represents its creative knowledge outcomes based on these metaphorical and conversational exchanges.

Using a touch table interface participants collaboratively narrate and visualise narrative sequences using online media obtained through a keyword search, or by words obtained from narrative templates. The search results are styled into generative behaviours that visually self-organise while participants make aesthetic choices about the narrative outcomes and their associated behaviours.

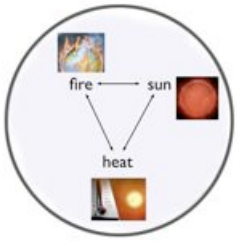
The playful interface supports collaboration through embedded mechanics that extend gestural actions commonly performed during casual conversations. By embedding metaphorical schemes associated with narrative comprehension, such as pointing, exchanging, enlarging or merging views, gestural action drives the experience and supports the conversational aspects associated with narrative exchange.

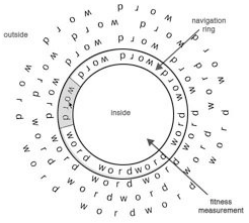




System Architecture

The system architecture models the narrative template events to allow a particular narrative perspective, globally or locally within the generated story world. This is done by modeling conversation relationships with the aim of self-organising and negotiating an agreement surrounding several themes. The system extends Conversation Theory (CT) (Pask, 1976), a theory of learning and social interaction, that outlines a formal method of conversation as a sense-making network. Based on CT entailment meshes with an added fitness metric, this develops a negotiated agreement surrounding several interrelated themes, that leads to eventual narrative coherence.





References

Hills, D., 'assimilate: An Interface for Collaborative Narrative Construction', ICIDS 2011, pp. 294-299.

Pask, G. *Conversation theory: Applications in education and epistemology*. Elsevier, Amsterdam, 1976.

Breeding on site

Tatsuo Unemi
Department of Information Systems Science
Soka University
Tangi-machi 1-236, Hachiōji, Tokyo 192-8577 Japan
unemi@iss.soka.ac.jp

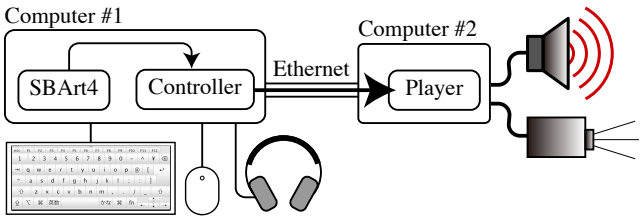


Figure 1: System setup.

This is a live-performance of improvisational productions and playbacks of a type of evolutionary art using a breeding tool, SBArt4 version 3 (Unemi 2010). The performer breeds a variety of individual animations using SBArt4 on a machine at his front in a manner of interactive evolutionary computation, and sends the genotype of his/her favorite individual to SBArt4Player through a network connection. Figure 1 is a schematic illustration of the system setups. Each individual animation that reached the remote machine is played back repeatedly with the synchronized sound effect until another one arrives. Assisted by a mechanism of automated evolution based on computational aesthetic measures as the fitness function, it is relatively easy to produce interesting animations and sound effects efficiently on site (Unemi 2011).

The player component has a functionality to composite another animation of feathery particles that reacts against the original image rendered by a genotype. Each particle moves guided by the force calculated from the HSB color value under the particle. The brightness is mapped to the strength, the hue value is mapped to the orientation, and the saturation is mapped to the fluctuation. This additional effects provide another impression for viewers.

The performance will start from a simple pattern selected from the initial population randomly generated, and then gradually shifts to complex patterns. The parameters of sound synthesis are fundamentally determined from statistic features of frame image so that it fits with the impression of visuals, but some of them are also subjects of real-time tuning. The performer is allowed to adjust several parameters such as scale, tempo, rhythm, noise, and the other modulation parameters (Unemi 2012) following his/her preference.

Because the breeding process includes spontaneous transformation by mutation and combination, the animations shown in a performance are always different from those in another occasion. This means each performance is just one time.

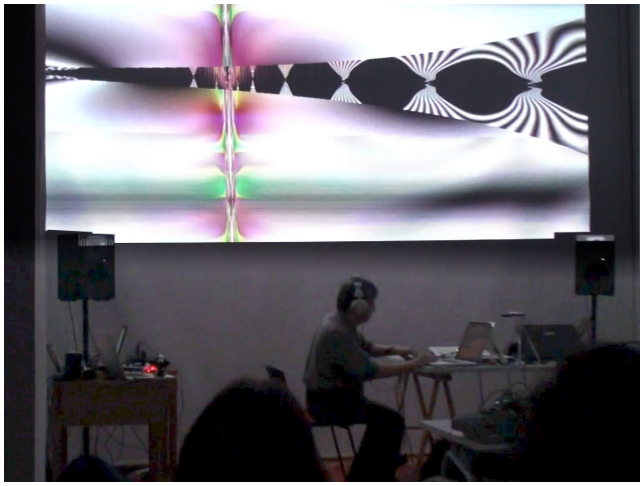


Figure 2: Live performance in Rome, December 2011.

References

Unemi, T. 2010. Sbart4 - breeding abstract animations in realtime. In *Proceedings of the IEEE World Congress on Computational Intelligence*, 4004–4009.

Unemi, T. 2011. Sbart4 as automatic art and live performance tool. In Soddu, C., ed., *Proceedings of the 14th Generative Art Conference*, 436–447.

Unemi, T. 2012. Synthesis of sound effects for generative animation. In Soddu, C., ed., *Proceedings of the 15th Generative Art Conference*, 364–376.

The projects website is:
`http://www.intlab.soka.ac.jp/~unemi/sbart/4/breedingOnSite.html`
Demo video:
`http://www.youtube.com/watch?v=1kKpWntUd8M`

A Fully Automatic Evolutionary Art

Tatsuo Unemi
Department of Information Systems Science
Soka University
Tangi-machi 1-236, Hachiōji, Tokyo 192-8577 Japan
unemi@iss.soka.ac.jp

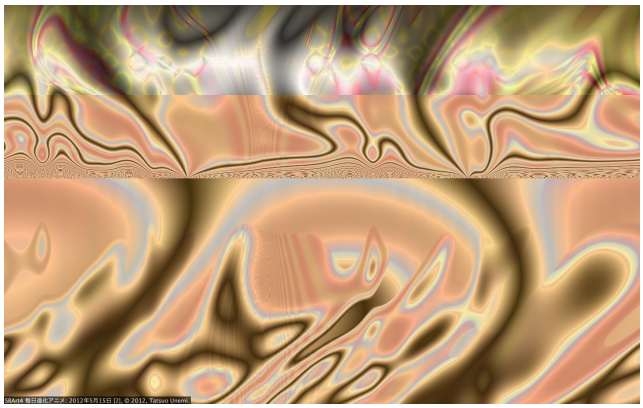


Figure 1: Sample image.

This is a project of an automatic art that the computer autonomously produces animations of a type of abstract images. Figure 1 is a typical frame image of an animation. A custom software, SBArt4 version 3, developed by the author is tanking a main role of the work, that based on a genetic algorithm utilizing computational aesthetic measures as fitness function (Unemi 2012a). The fitness value is a weighted geometric mean of measures including complexity, global contrast factor, distribution of color values, distribution of edge angles, difference of color values between consecutive frame images, and so on.

Figure 2 illustrates the system configuration using two personal computers connected by the Ethernet. The left side is for evolutionary process, and the right side is for rendering and sound synthesis. Starting from a population randomly initialized with mathematical expressions that determines the color value for each pixel in a rectangular area, a never-ending series of abstract animations are continuously displayed on the screen in turn with synchronized sound effect (Unemi 2012b). Each of the 20 seconds animation is corresponding to an individual of relatively high fitness chosen from the population in the evolutionary process.

The evolutionary part is using Minimal Generation Gap model (Satoh, Ono, and Kobayashi 1997) for the generational alternation to guarantee the time for each computation step is minimal. After 120 steps of generational alternations, the genotypes of the best ten individuals are sent to the player side in turn. To avoid convergence to lead a narrower variation of individuals in the population, the individuals of lower fitness in one forth of the population are replaced with random genotypes for each 600 steps.

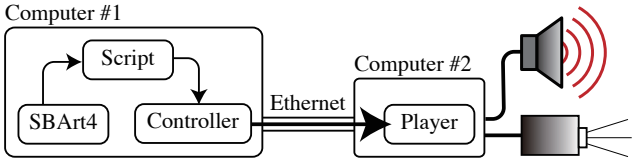


Figure 2: System setup.

The visitors will notice not only the recent progress of the power of computer technology but also will possibly be given an occasion to think what the artistic creativity is. These technologies are useful not only to build up a system that makes unpredictable interesting phenomena but also to provide an occasion for people to reconsider how we should relate to the artifacts around us. We know the nature is complex and often unpredictable, but we, people in the modern democratic society, intend to assume that artificial systems should be under our control and there must be some person who takes responsibility on the effects. The author hopes the visitors will notice that it is difficult to keep some of the complex artifacts under our control, and will learn how we can enjoy with them.

References

Satoh, H.; Ono, I.; and Kobayashi, S. 1997. A new generation alternation model of genetic algorithms and its assessment. *Journal of Japanese Society for Artificial Intelligence* 12(5):734–744.

Unemi, T. 2012a. Sbart4 for an automatic evolutionary art. In *Proceedings of the IEEE World Congress on Computational Intelligence*, 2014–2021.

Unemi, T. 2012b. Synthesis of sound effects for generative animation. In Soddu, C., ed., *Proceedings of the 15th Generative Art Conference*, 364–376.

Demo video:
`http://www.youtube.com/watch?v=XBej_nlu-Hg`

The Fourth International Conference on
Computational Creatiivity
ICCC 2013
Sydney, Australia
12-14 June 2013

in cooperation with

supported by

THE UNIVERSITY OF SYDNEY

designlab

ISBN: 978-1-74210-317-4