Ruled by Utopia: Procedurally Generated Mega-structures

Guilherme L. Almeida[1,*], João M. Cunha[1], Sérgio M. Rebelo[1] and Tiago Martins[1]

[1]*University of Coimbra, CISUC/LASI – Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering*

Abstract

Many building generation approaches focus on functional requirements or life-size concerns, creating a gap for aesthetic-focused building generation systems. We present a system that generates utopian mega-structures using procedural content generation techniques. Our goal is that the output of the system can serve as inspiration and visual explorations, or be materialized into dioramas, puzzles, scenographic elements, or graphic design material.

Keywords

Procedural Generation, Computational Design, Utopian Architecture, Rule-Based Systems, Generative Design, Parametric Design

1. Context

Buildings can be much more than merely functional spaces. Fictional genres such as *sci-fi* and *cyberpunk* have presented audiences with ever-growing, busy buildings of questionable functionality that strive for visual appeal and creativity, while also speculating about what the future of architecture might look like. The idea for this system stems from these utopian notions, forgoing functionality to promote creative believability. The building generation system presented in this pictorial employs several Procedural Content Generation (PCG) techniques common in Procedural Generation of Buildings (PGB) [1], such as Generative Grammars (GG) [2, 3] and Pseudo-Random Number Generation (PRNG) [4, 5], and generates complete, non-furnished building models.

2. Approach

To make the most out of PRNG and GG techniques, we felt it would be useful to define and analyze a target set of outputs from which we could abstract similarities, hierarchies, and variations. The main architectural inspiration for this project was the work of Ana Aragão [6], particularly the *Vertical Reclamation of Individual Spaces* series (see Figure 1 - left), which we used as a guideline for the system's current implementation. Those illustrations serve as a prime example of what we plan to accomplish with this system: very vertical, monumental and busy structures that border on surrealism, while still maintaining a level of believability. Other influences include the Japanese Metabolism style, the works of Shin Takamatsu, or Nigel Phelps' work on *Batman* (1989) (see Figure 1 - right), due to their vertical, modular and hierarchical nature. To the best of our knowledge, there are no building generation systems based on these architectural references in the literature.

Given our set of target outputs, we opted for a common structure for all buildings: a series of vertically disposed *sections*, comprised of several *towers*, whose walls are populated with *modules* (see Figure 2). The building generation is controlled by the following set of parameters:

- Starting Surface;
- Main Seed;

Workshop on Computational Design and Computer-aided Creativity 2025, 23 June, 2025, Campinas, BR

*Corresponding author.

guialmeida@student.dei.uc.pt (G. L. Almeida); jmacunha@dei.uc.pt (J. M. Cunha); srebelo@dei.uc.pt (S. M. Rebelo); tiagofm@dei.uc.pt (T. Martins)

0009-0001-8616-0376 (G. L. Almeida); 0000-0001-6502-3500 (J. M. Cunha); 0000-0002-7276-8727 (S. M. Rebelo); 0000-0003-2638-237X (T. Martins)

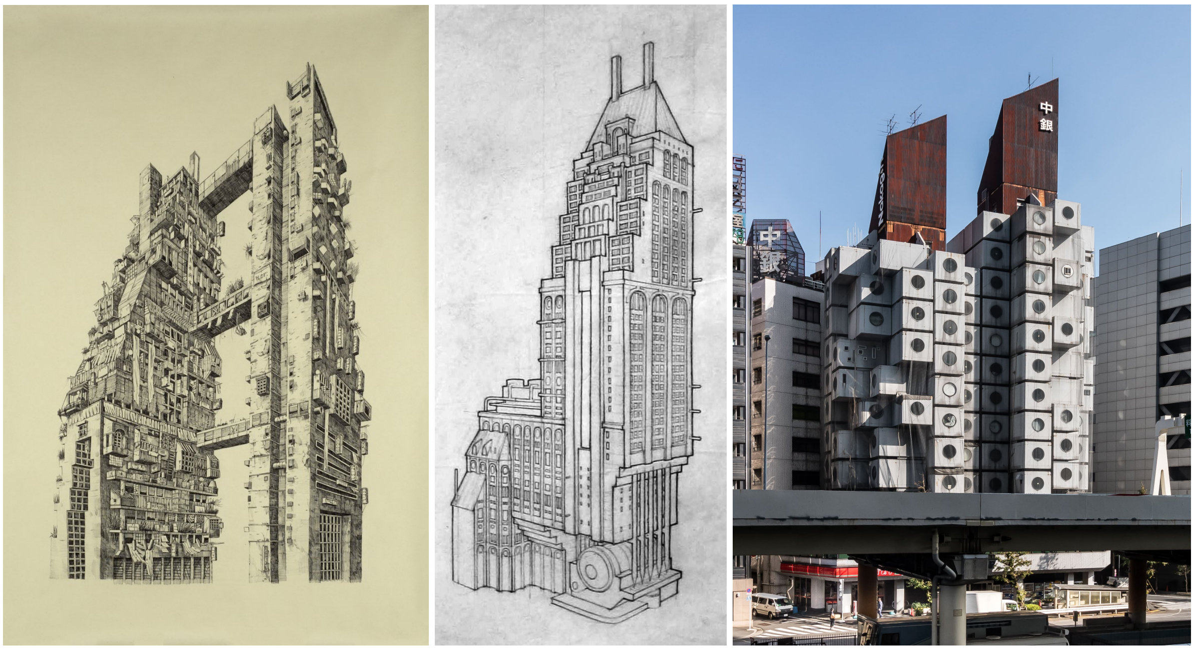

Figure 1: Left: One of the illustrations in the *Vertical Reclamation of Individual Spaces* series, by Ana Aragão. Middle: Concept art for the *Flugelheim Museum*, by Nigel Phelps. Right: The *Nakagin Capsule Tower*, an example of the Japanese Metabolism style.

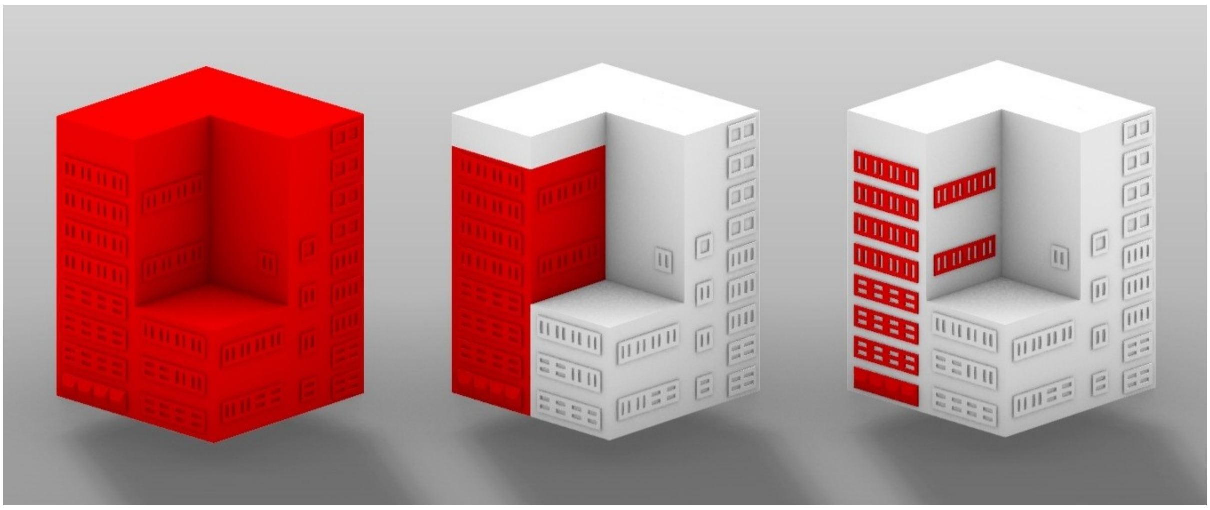

Figure 2: Main structural units, highlighted in red. In order, from left to right: *section*, *tower*, and *modules*.

- Number of Sections;
- Minimum and Maximum Tower Height;
- Filler Height Tolerance;
- Section Contraction and Expansion Limit;
- Window Minimum and Maximum Scaling;

First, the section bases are partitioned by a simple generative grammar. All terminal sections are then extruded upward by a random height factor between *MinimumTowerHeight* and *MaximumTowerHeight* to form towers. Having separated towers allows the system to individually customize and populate different towers or parts of a section with different properties, not unlike what we see in our intended outputs.

Given the height diferences between towers, simply placing the next foundation over the tallest tower would not look believable; therefore, the towers are then re-extruded if they are within a certain height threshold, defined by *FillerHeightTolerance* (see Figure 2). The bounding box of the topmost areas is then scaled up or down by a factor between *SectionContractionLimit* and *SectionExpansionLimit*, and becomes the starting surface for the next section. This process is repeated a number of times equal to *NumberOfSections*.

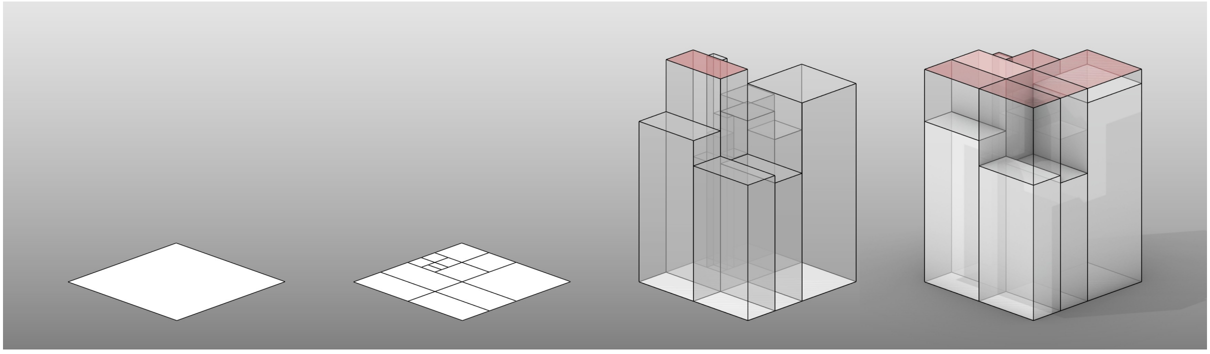

Figure 3: Main steps in the construction phase for each section. The difference between the topmost area with and without the fillers is highlighted in red in the third and fourth steps.

Once all sections are generated, the system populates the visible walls with any number of user-defined modules. First, the system filters out any faces or parts of faces that are not visible from the outside. The height and width of each resulting face defines the upper and lower limits for the number of rows and columns of a grid, and a 3D Simplex noise space selects the specific module to be placed at each grid point, resized proportionately to the wall's dimensions. In the future, we hope to accomplish the module application step using GG techniques, rather than a noise space. The source code repository can be found at: https://github.com/GuilhermeAlmeida1608/LDC-Reclaimers.

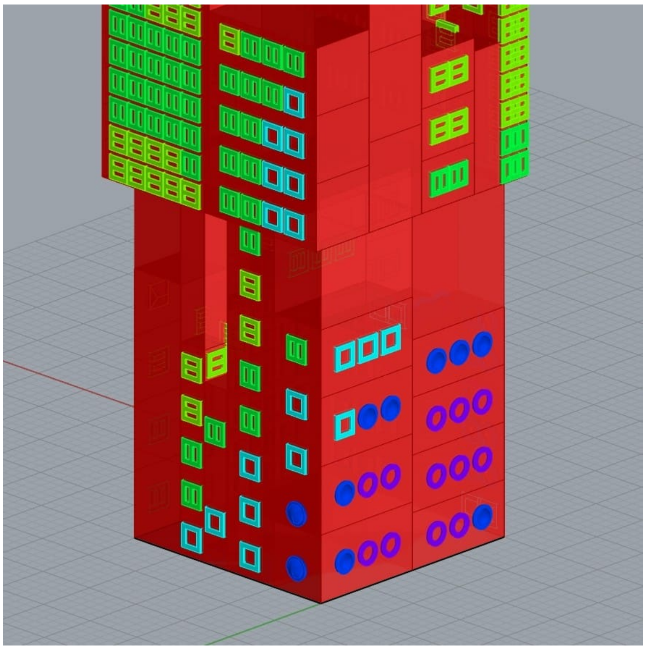

Figure 4: Example of module distribution. The color bands formed by the modules evidence the Simplex noise space.

3. Results

Our findings demonstrate that the system can create a wide variety of buildings, both with different parameter sets and within each set (see Figures 5, 6 and ??). To this effect, we will present two representative outputs for a few different parameter sets, highlighting the most relevant ones. The parameters will be written in the figure captions, following this format:

> (NumberOfSections, MinimumTowerHeight, MaximumTowerHeight, FillerHeightTolerance, SectionContractionLimit, SectionExpansionLimit, WindowMinScaling, WindowMaxScaling)

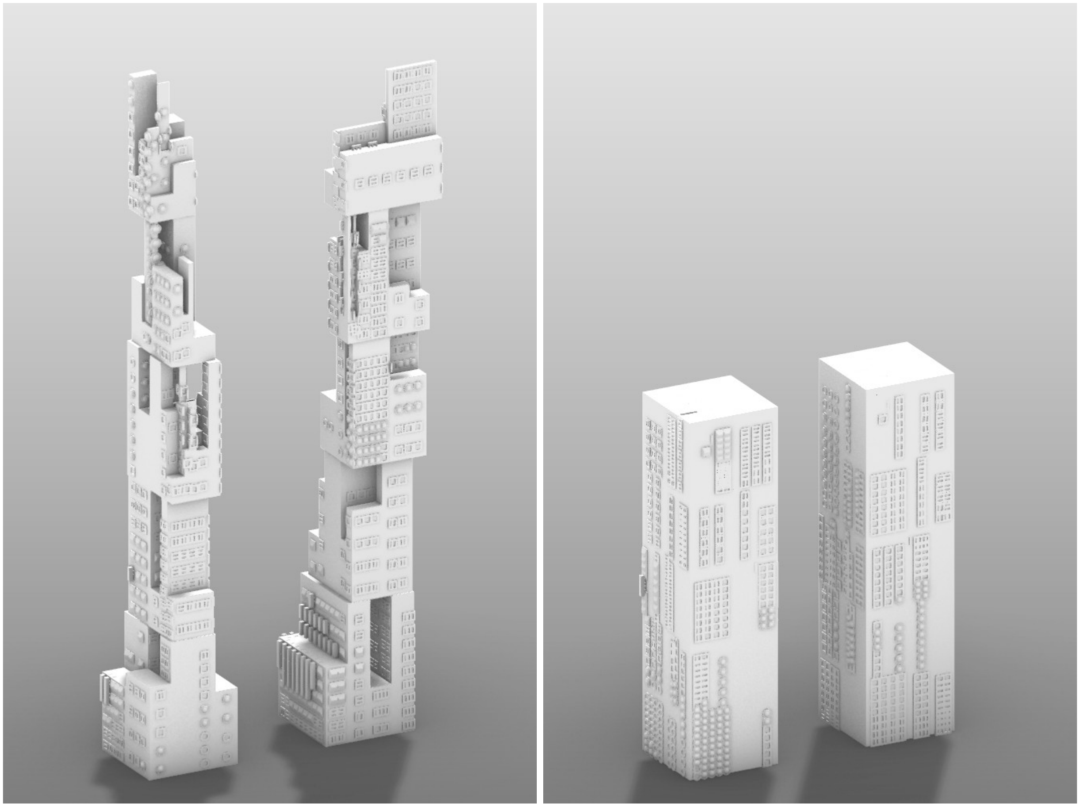

Figure 5: Set 2 (4, 10, 30, 13, 0.7, 1.5, 0.1, 0.7) (left) and Set 4(5, 10, 16, 10, 1, 1, 0.1, 0.7) (right). The most notable difference between the two is the effect caused by a higher *FillerHeightTolerance* in Set 2, which creates more recesses and thinner profiles.

The results, although preliminary, already demonstrate the system's possible applications. The final models can be 3D printed to make scenographic elements at a variety of scales or be used as decorative pieces (see Figure 7). Compositions of multiple buildings with varying styles can be created based on their arrangement, and the buildings' components can be separated into physical assembly kits or puzzles, similar to Santos et al. [7].

Acknowledgments

This work is financed through national funds by FCT – Fundação para a Ciência e a Tecnologia, I.P., in the framework of the Project UIDB/00326/2025 and UIDP/00326/2025.

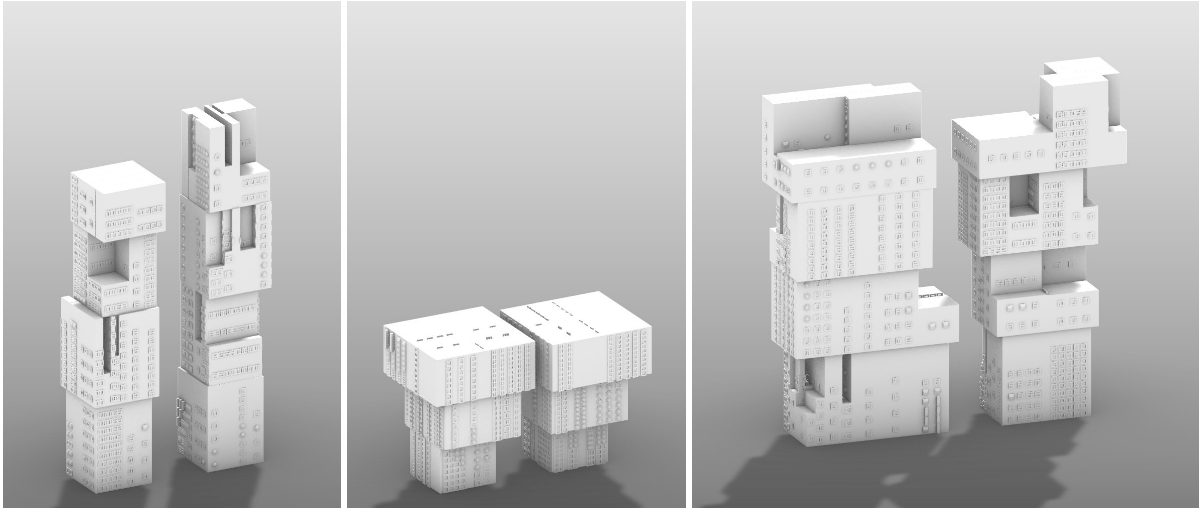

Figure 6: Set 1 (4, 10, 30, 13, 0.7, 1.5, 0.1, 0.7) (left) and Set 3 (3, 15, 16, 10, 1.1, 1.6, 0.1, 0.7) (middle). Manipulating the expansion and contraction can create buildings that get progressively wider or thinner, or a mixture of both. On the right, Set 1 applied to different starting foundations.

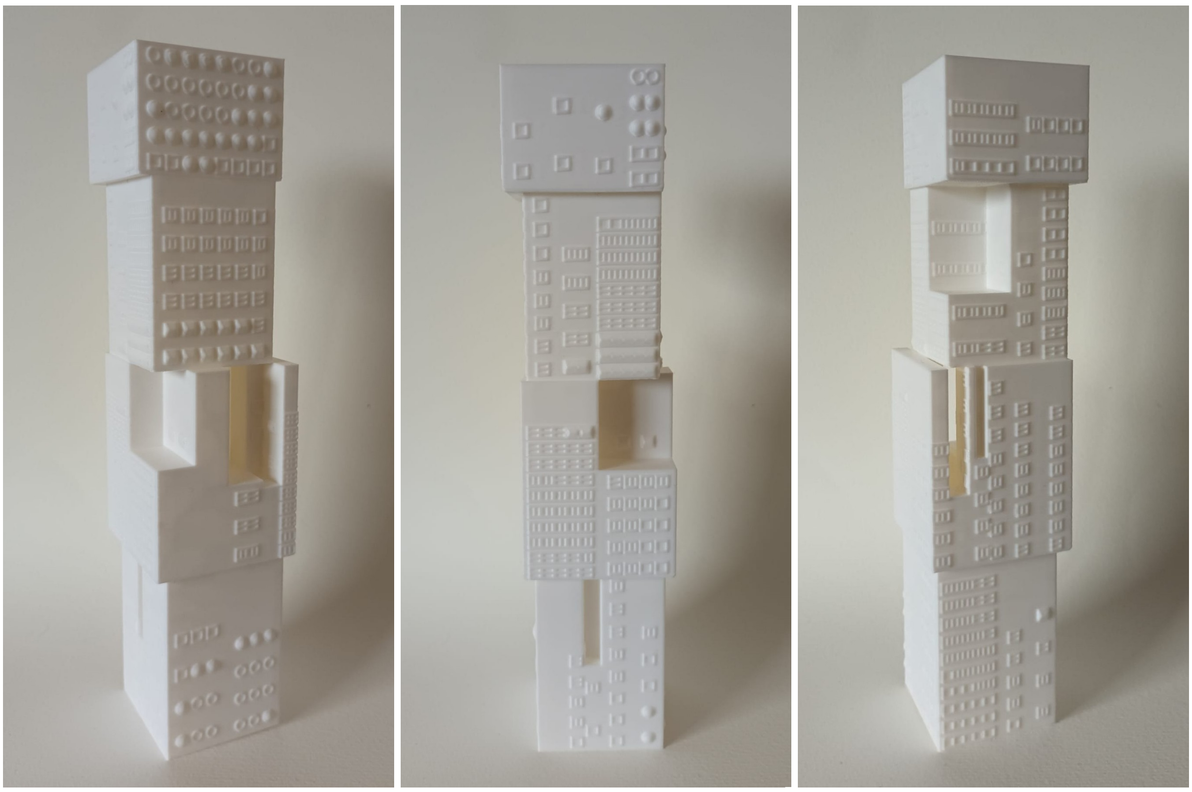

Figure 7: 3D printed model of a building generated with the Set 1 parameters, at a 24 cm height scale.

References

[1] D. Kutzias, S. von Mammen, Recent advances in procedural generation of buildings: From diversity to integration, IEEE Transactions on Games 16 (2023) 16–35.

[2] P. Müller, P. Wonka, S. Haegler, A. Ulmer, L. Van Gool, Procedural modeling of buildings, in: ACM SIGGRAPH 2006 Papers, 2006, pp. 614–623.

[3] P. Wonka, M. Wimmer, F. Sillion, W. Ribarsky, Instant architecture, ACM Transactions on Graphics

Figure 8: Vectorized render of a set of buildings generated by the system.

(TOG) 22 (2003) 669–677.
[4] G. Kelly, H. McCABE, An interactive system for procedural city generation, Institute of Technology Blanchardstown 25 (2008).
[5] S. Greuter, J. Parker, N. Stewart, G. Leach, Real-time procedural generation of pseudo-infinite cities, in: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia, 2003, pp. 87–ff.
[6] A. Aragão, https://www.anaaragao.com/, Accessed May 2025.
[7] J. Santos, R. Murta, J. M. Cunha, S. M. Rebelo, T. Martins, P. Martins, Evolving urban landscapes, in: EPIA Conference on Artificial Intelligence, Springer, 2023, pp. 64–76.