Eurographics Workshop on Urban Data Modelling and Visualisation (2013)
V. Tourre and G. Besuievsky (Editors)

Challenges in Procedural Modeling of Buildings

Gustavo Patow and Gonzalo Besuievsky

ViRVIG Institute, Universitat de Girona, Spain

Abstract

The use of procedural modeling for building generation has risen dramatically over the last years, being an elegant and fast way to generate huge, complex and realistically looking urban sites. However, due to its generative nature there are still unsolved problems that limits they usage. In this paper we report on the challenges still pending on procedural modeling of buildings. We provide a state of the art on most recent solution and we draw possible research avenue that could be taken for spreading the use of procedural modeling in current applications.

1. Introduction

A broad range of areas, such as games, movies or urban simulation require virtual 3D city models with detailed geometry. Procedural modeling [MWH*06] has proven to be effective for this task [WMV*08], offering a potential alternative to the labor-intensive modeling required by traditional 3D techniques. However, traditional procedural methods are not always a suitable alternative to manual modeling. With this, there is an increasing need for more advanced content creation and editing tools.

In this paper we try to provide some insights we have observed over the years about important open research areas in procedural building modeling. In particular, we try to bring attention to those areas where we have found important key aspects that are far from being completely solved, and that would drive the field closer to the objective of realizing large, detailed and complex cities, both real and synthetic. Please, observe that here we focus only on the problems in procedural modeling from the Computer Graphics side, leaving aside important aspects like urban simulations (e.g., lighting, heat, wind); statistical information about inhabitant density; etc. Also, we focus on methodologies that could bring almost completely automatic techniques, also with the possibility of a simple editing process to accommodate to the final user requirements. Of course, this is not an easy task, but we hope this paper will shed some light on the aspects that, in our opinion, will be the key to success in this objective.

2. State of the art in procedural building modeling

In this section it is not our intention to do an exhaustive review of all the literature in procedural building modeling. We focus our attention on the works that open new research areas or illustrate the key concepts we would like to present in this paper. First, we need to define Procedural Modeling, which is a term that describes a family of techniques that generate geometry from a set of rules. These approaches [WWSR03, MWH*06] have emerged as an elegant solution for the generation of massive urban landscapes from a simple ruleset. The interested reader is referred to the articles by Vanegas et al. [VAW*10] and Watson et al. [WMV*08] for a more in-depth review of the current state of the art in this topic.

The seminal works by Wonka et al. [WWSR03] and Müller et al. [MWH*06] introduced Grammar-based procedural modeling for buildings. The main concept of this technique is a shape grammar, which is based on a ruleset: starting from an initial axiom primitive (e.g. a building outline), rules are iteratively applied, replacing shapes with other shapes. A rule has a labeled shape on the left hand side, called predecessor, and one or multiple shapes (also called primitives) and commands on the right hand side, called successor:

$$predecessor \rightarrow CommandA, CommandB : labelB;$$
$$labelB \rightarrow CommandC : labelC;$$

The resulting geometry is formed by shapes that can be optionally assigned new labels to be further processed. The main commands, the macros that create new shapes in the classic approach, are: *Subdivision*, that performs a subdivision of the current shape into multiple shapes; *Repeat*, that performs a repeated subdivision of one shape multiple times; *Component split*, that creates new components shapes (faces

DOI: 10.2312/UDMV/UDMV13/025-028

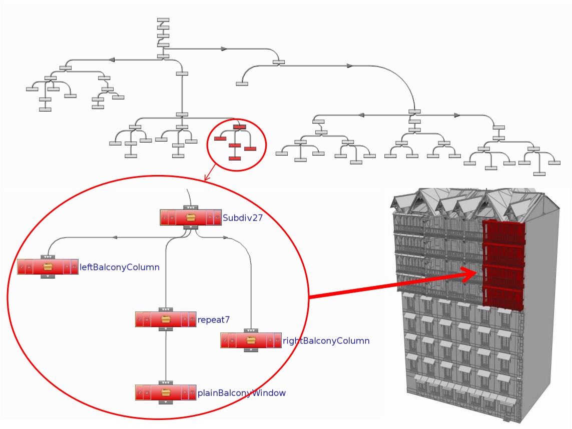

Figure 1: *A graph-based representation of a procedurally modeled building. Image courtesy of [BBP13].*

or edges) from initial volumes; and *Insert* command that replaces a pre-made asset on a current predecessor.

Traditionally, grammars create a hierarchy of shapes by processing each rule's predecessor shape, replacing it by its products (successor shapes). This process is executed until only terminal shapes are left. The whole production process can be seen as a graph where each node represents an operation applied to its incoming geometry stream and the leaf nodes are the geometry assets (see Figure 1).

Lipp et al. [LWW08] presented one of the first attempts to improve editing operations using an interactive visual system, avoiding the edition of text rules. Later, CityEngine [Esr], Epic's UDK [Epi12] and Patow [Pat12] independently developed visual representations for the rulesets, considerably easing the development process. More recently, some approaches tried to provide direct control to visual procedural languages, resulting in a simple visual traversal of the hierarchy tree plus direct assignment of the desired changes [RP12]. Krecklau and Kobbelt [KK12] added specific rules selecting a given label with a given shape index to apply a local change. Lin et al. [LCOZ*11] were the first to present a technique for retargeting irregular 3D architecture while preserving its structure.

3. Challenges

Here we will present the challenges we have observed in procedural modeling of buildings. Obviously, a single classification would not suffice, as the different challenges we will enumerate are interconnected and their boundaries are blurred. One burning question in procedural modeling of buildings is their usage in the context of a city, where thousands of different buildings are present. Procedural models are good at representing "standardized" buildings, where a few base rulesets are used for the whole city. To increase complexity, creators often parameterize these rulesets, and often add randomness to hide the underlying homogeneity.

3.1. Big Cities

One of the biggest strengths of procedural modeling when applied for buildings and cities, is also one of it most crucial weaknesses: its ability to generate large amounts of geometry from a small set of elements (the rulesets). This phenomenon, known as *Geometric Explosion*, may result in terabytes of geometry generated in a single run, which can surpass any computer processing and storage capabilities.

- On-the-fly real-time facade generation is one promising approach explored by some works [HWM*10, AYRW09, MGHS, KK11], but there still is a lot of room for improvement as the problem of visualizing complex facades with complex geometric detail is still an open issue.
- Client-server architectures [CO11], where a computer is used exclusively to generate building models while another one is used solely for caching and rendering purposes, can also be used.
- Cloud-based approaches are the next logical step after the previous one. These approaches would solve the problem for having a distributed storage system which holds the whole geometric information, serving several users at the same time. We can consider that each user visualizing the city (e.g., playing a sandbox multiplayer game) is performing a set of localized, highly coherent queries. Taking advantage of these concrete characteristics is, up to now, a completely untouched area.
- Although these techniques can generate large amounts of geometry, it is logical to ask whether all the generated geometry is really needed. Many applications, from single-user walkthroughs up to urban simulations, do not need the full generated geometry, at least not at every point in the model. A classic solution for huge models is to introduce level-of-detail techniques that use quality parameters to reduce the model complexity. However, in contrast to the classic procedure, the simplification should not be done after the building model is created, but generated automatically within the procedural creation. Improved procedural tools to automatically control both model complexity and local model frequency would be extremely helpful and should be developed (Figure 2).

3.2. Interactivity and User Interfaces

Although some efforts have been devoted to generate simple user interfaces to ease content creation, we still need ways to interactively edit the rulesets by a non-technical user. We believe that the approaches done so far [LWW08, Esr, Epi12, Pat12, KW11, BBP13] represent a few steps forward in this direction, but they are still clearly insufficient to bring procedural modeling closer to the layman. To create a new ruleset from scratch, it is needed a deep understanding of the ruleset

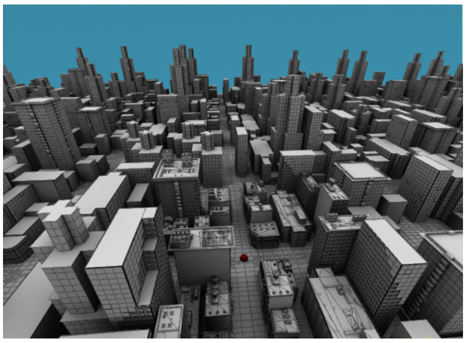

Figure 2: *A huge city model composed of millions of polygons, procedural LoD is required for efficient manipulation.*

creation process, which is conceptually quite close to programming, so ruleset creation requires some technical skills that many artists simply do not have.

3.3. Irregular architecture

Much of the literature so far has dealt with completely regular or semi-regular facades [AYLM13]. Some of the most important reconstruction or inference papers up to date have focused on the exploitation of symmetry (or partial symmetries) [MP08, BWS] to provide astonishing results. However, since the last centuries, architects have avoided perfect symmetry in their buildings, using semi-regular or even completely irregular structures. To the best of our knowledge, the only papers to deal with irregular architecture are the ones presented by Lin et al. [LCOZ*11] and by Bao et al. [BSW13]. In the first approach, the authors apply techniques that have worked for regular facades, to irregular ones: for each user interaction (limited to building stretching in one of the different directions), they instantly built a completely *regular* ruleset tailored at dealing with the one-dimensional scaling. However, this technique breaks after several iterations, transforming the resulting building into a regular one. In the second, Bao et al. took a single layout and generated facade variations, but limited to layouts where a hierarchical, rectangular decomposition exists. This way, both were able to re-use known techniques for regular buildings, translating them to the irregular domain. So, although they are very promising approaches, it becomes evident that further research is also needed in this area.

3.4. Time-dependence

Modeling and visualizing historic buildings [WOD09, Hav05, BP12] in a cost-effective and efficient way is still an open issue (see Figure 3). Also, quite related to the previous point, target users for this kind of buildings are historians or cultural heritage experts, with little or no knowledge of procedural techniques. A simple and intuitive user interface for these users is not only desirable, but can even be mandatory.

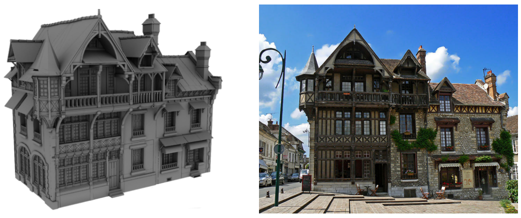

Figure 3: *Raccolet House, Moret sur Loing, France. Procedural reconstruction (left) of an historical house from photographs (left). Original model from www.daz3d.com.*

On the other hand, it must be recognized that buildings are not static objects, they have a clear evolution since their conception up to their final ruins. Neglecting this evolution may bring a distorted, biased view of their own history, so modeling these phenomena is an important issue in these domains. A procedural approach for architecture evolution was somewhat touched by Cutler et al. [CDM*02], but not specifically tailored at procedural buildings. Further research is clearly needed to generate the tools for this kind of studies.

3.5. Real-world architecture

There is an increasingly large body of literature dealing with the recovery of facades from photographs [MZWG07, KST*09, MWW12, CML*12, AYLM13] or point clouds [BWS, MPWC12]. All these represent effective approaches to recover 3D geometry form an input dataset, repairing it and making it usable. However, for large-scale scenarios like a whole city, it remains an open challenge to acquire and process large amounts of raw data. Achieving the same degree of effectiveness for thousands of building point clouds is another avenue for further research. Also, we should not forget that representing a real city is a challenge in itself. Here, procedural techniques are a promising research line in their generalization possibilities, as defining architectonic styles has already been demonstrated to be feasible [VAW*10]. Finally, procedural techniques could be used the other way round, to compress urban data into a much smaller dataset of procedural rules, which could be later used to restore the original geometric detail. As far as we know, this is an untouched possibility of urban procedural modeling.

3.6. Automatic ruleset generation

This topic, of course, is tightly coupled with the previous one, and consists of the generation of whole new rulesets starting from some initial information. Very promising approaches were described by Merrell et al. [MSK10], for residential building layouts; Talton et al. [TLL*11], who proposed a Metropolis-based procedural modeling framework which alliterated the artist from the specific ruleset declaration; and Patow [Pat12], who proposed a visual language plus a graph-rewriting approach to repair minor structural defects in the user-provided ruleset. All these approaches have in common their objective of freeing the user from the tedious ruleset generation process, but we believe there is still much pom for improvement with this goal in mind.

4. Conclusions

Although procedural modeling of buildings is a burgeoning field, we strongly believe there are several lines in which there are still pending issues to be solved. These issues cover almost the whole spectrum of applications, which guarantees fruitful research for several years to come.

Acknowledgements

This work was partially funded by the TIN2010-20590-C02-02 project from Ministerio de Ciencia e Innovación, Spain.

References

[AYLM13] ALHALAWANI S., YANG Y.-L., LIU H., MITRA N. J.: Interactive facades: Analysis and synthesis of semi-regular facades. *Computer Graphics Forum (Eurographics)* (2013). 3

[AYRW09] ALI S., YE J., RAZDAN A., WONKA P.: Compressed facade displacement maps. *IEEE Transactions on Visualization and Computer Graphics 15*, 2 (2009), 262–273. 2

[BBP13] BARROSO S., BESUIEVSKY G., PATOW G.: Visual copy & paste for procedurally modeled buildings by ruleset rewriting. *Computers & Graphics*, 0 (2013), –. 2

[BP12] BARROSO S., PATOW G.: Visual Language Generalization for Procedural Modeling of Buildings. In *XXII Spanish Computer Graphics Conference* (2012), pp. 57–66. 3

[BSW13] BAO F., SCHWARZ M., WONKA P.: Procedural facade variations from a single layout. *ACM Trans. Graph. 32*, 1 (Feb. 2013), 8:1–8:13. 3

[BWS] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. 3

[CDM*02] CUTLER B., DORSEY J., MCMILLAN L., MÜLLER M., JAGNOW R.: A procedural approach to authoring solid models. *ACM Trans. Graph. 21*, 3 (July 2002), 302–311. 3

[CML*12] CEYLAN D., MITRA N. J., LI H., WEISE T., PAULY M.: Factored facade acquisition using symmetric line arrangements. *Computer Graphics Forum* (2012). 3

[CO11] CULLEN B., O'SULLIVAN C.: A caching approach to real-time procedural generation of cities from gis data. *Journal of WSCG 19*, 3 (2011), 119–126. 2

[Epi12] EPICGAMES: Unreal development kit (udk), 2012. http://udk.com. 2

[Esr] ESRI: Cityengine. http://www.esril.com/. 2

[Hav05] HAVEMANN S.: *Generative Mesh Modeling*. PhD thesis, Braunschweig Technical University, Germany, 2005. 3

[HWM*10] HAEGLER S., WONKA P., MÜLLER S., VAN GOOL L., MÜLLER P.: Grammar-based encoding of facades. *Computer Graphics Forum 29* (2010), 1479–1487. 2

[KK11] KRECKLAU L., KOBBELT L.: Realtime compositing of procedural facade textures on the gpu. In *Invited Paper at 3D-Arch 2011 (ISPRS)* (2011). 2

[KK12] KRECKLAU L., KOBBELT L.: Interactive modeling by procedural high-level primitives. *Computers & Graphics*, 0 (2012), –. 2

[KST*09] KOUTSOURAKIS P., SIMON L., TEBOUL O., TZIRITAS G., PARAGIOS N.: Single view reconstruction using shape grammars for urban environments. In *ICCV* (2009). 3

[KW11] KELLY T., WONKA P.: Interactive architectural modeling with procedural extrusions. *ACM Trans. Graph. 30*, 2 (Apr. 2011), 14:1–14:15. 2

[LCOZ*11] LIN J., COHEN-OR D., ZHANG H., LIANG C., SHARF A., DEUSSEN O., CHEN B.: Structure-preserving retargeting of irregular 3d architecture. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 183:1–183:10. 2, 3

[LWW08] LIPP M., WONKA P., WIMMER M.: Interactive visual editing of grammars for procedural architecture. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 102:1–10. 2

[MGHS] MARVIE J.-E., GAUTRON P., HIRTZLIN P., SOURIMANT G.: Render-time procedural per-pixel geometry generation. In *Proceedings of Graphics Interface 2011*. 2

[MP08] MITRA N. J., PAULY M.: Symmetry for architectural design. In *Advances in Architectural Geometry* (2008), pp. 13–16. 3

[MPWC12] MITRA N. J., PAULY M., WAND M., CEYLAN D.: Symmetry in 3d geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-art Report* (2012). 3

[MSK10] MERRELL P., SCHKUFZA E., KOLTUN V.: Computer-generated residential building layouts. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 181:1–181:12. 3

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 1

[MWW12] MUSIALSKI P., WIMMER M., WONKA P.: Interactive Coherence-Based Façade Modeling. *Computer Graphics Forum 31*, 2 (2012), 661–670. 3

[MZWG07] MÜLLER P., ZENG G., WONKA P., GOOL L. V.: Image-based procedural modeling of facades. 3

[Pat12] PATOW G.: User-friendly graph editing for procedural modeling of buildings. *IEEE Computer Graphics and Applications 32* (2012), 66–75. 2, 3

[RP12] RIU A., PATOW G.: Bringing direct local control to interactive visual editing of procedural buildings. In *XXII Spanish Computer Graphics Conference* (2012), pp. 67–75. 2

[TLL*11] TALTON J. O., LOU Y., LESSER S., DUKE J., MĚCH R., KOLTUN V.: Metropolis procedural modeling. *ACM Trans. Graph. 30*, 2 (Apr. 2011), 11:1–11:14. 3

[VAW*10] VANEGAS C. A., ALIAGA D. G., WONKA P., MÜLLER P., WADDELL P., WATSON B.: Modelling the appearance and behaviour of urban spaces. *Comput. Graph. Forum 29*, 1 (2010), 25–42. 1, 3

[WMV*08] WATSON B., MÜLLER P., VERYOVKA O., FULLER A., WONKA P., SEXTON C.: Procedural urban modeling in practice. *IEEE Computer Graphics and Applications 28* (2008), 18–26. 1

[WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28* (December 2009), 112:1–112:9. 3

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM Transaction on Graphics 22*, 3 (July 2003), 669–677. Proceedings ACM SIGGRAPH 2003. 1