Proceduralization of Urban Models

İlke Demir
Computer Science Department
Purdue University
West Lafayette, IN, USA
idemir@purdue.edu

Daniel G. Aliaga
Computer Science Department
Purdue University
West Lafayette, IN, USA
aliaga@purdue.edu

Bedrich Benes
Computer Graphics Technology
Purdue University
West Lafayette, IN, USA
bbenes@purdue.edu

***Abstract*—Architectural models have always played an important role in computer graphics, virtual environments, and urban planning; due to the size, detail and complexity of such models. Creating detailed and realistic buildings needs time and extensive coding as well as significant domain expertise. Observing the content creation problem and the challenges of procedural modeling, we identified and developed a generalized proceduralization framework, that converts existing 3D urban models into an easy to manipulate procedural form. Our proceduralization framework for urban spaces allows re-use of existing models in various formats by first conducting shape processing methods to find and exploit the repetitions and similarities to reveal the hidden high-level structural information, and then by using grammar discovery methods on those components to extract a representative grammar of the model. Additionally, we have proposed applications of such procedural representations, including completion of point cloud models and structure-aware synthesis tools for procedural editing.**

***Keywords*—proceduralization, architectural editing, procedural modeling, inverse procedural modeling, shape analysis, geometry processing, grammars.**

I. Introduction

The technological developments in the last century have carried us from a few pixels per screen to infinite worlds. Consecutively, recent bottleneck in graphics industry has been changing from the 'means' to the 'objects', in other words, from the technology to the content. In particular, architectural models have always played an important role in areas like computer graphics, virtual environments, and urban planning. Because of the proliferation of such areas, the demand for city-scale 3D urban models has significantly increased, as well as the availability of 3D data sets of buildings and other urban structures.

One option is to manually create such content, however humans are expensive and slow. Another option is to efficiently re-use the existing large set of 3D polygonal models available from public databases, created by scans, images, time-of-flight cameras, manual modeling, etc. However the results of these approaches usually lack high-level grouping or segmentation information which hampers efficient re-use and synthesis. Procedural models are known to be an effective solution to create massive amount of content from powerful and compact parameterized representations (Figure 1). While procedural modeling provides compelling architectural structures, creating detailed and realistic buildings needs time and extensive coding as well as significant domain expertise.

978-1-5090-6494-6/17/$31.00

Observing the content creation problem and the challenges of procedural modeling, we identified and developed a set of *proceduralization* tools, that convert existing models into an easy to manipulate procedural form and allow quick synthesis of visually similar objects. The central idea of our research is that we can *automate* and *assist* modeling and creation tasks by proceduralization of existing models such as architectural meshes, building point clouds, or textured urban areas. The geometrical and visual information hidden in already existing models actually contain high-level semantic and structural information, so a proceduralization framework can reveal such information for a variety of purposes.

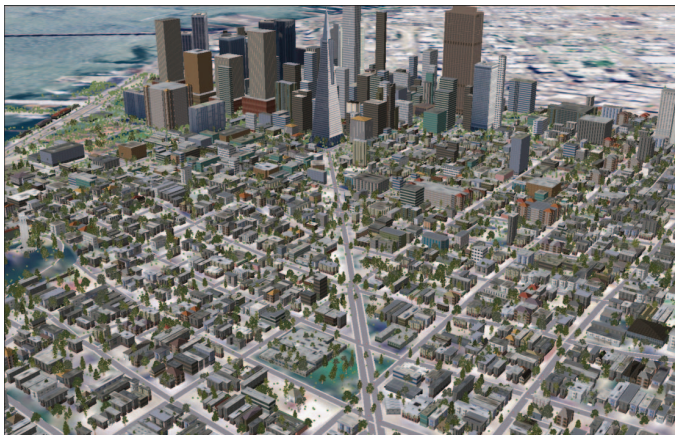

Fig. 1: **A Proceduralized World.** A procedurally generated view of San Francisco by UrbanVision [1].

We review and combine our previous work in a generalized proceduralization framework introduced in this paper. Our research builds upon a general proceduralization framework for urban spaces that allows the re-use of existing models in various formats, by conducting shape analysis methods on such models and proposing grammar discovery methods on those components to extract a representative grammar (or a procedural form) of the model. We have shown that the structural information and repetitions within already existing models can be exploited to obtain the high-level representation hidden in such models for making the design and modeling of urban content more efficient and intuitive. Initial applications

of our work enables the benefits of procedural modeling on existing models such as compression, rendering, and content generation. Additionally, we have proposed new applications of such procedural representations, including completion of incomplete models, geo-localization within urban areas, and structure preserving synthesis systems to overcome the control problem of procedural generation. We expect our effort will enlighten the artists‘ and designers‘ creation process by converting the existing modeling tools into faster, easier to implement, more interactive, and more intuitive procedural modeling systems, using the power of proceduralization.

II. Previous Work

An urban space is a complex collection of architectural structures arranged into buildings, blocks, parcels and neighborhoods interconnected by streets, and distributed over a terrain. Understanding, describing, and predicting the appearance and behavior of cities is useful in a growing number of applications. Traditionally, modeling urban spaces has been a rather manual task that consumes significant amounts of resources. With the very fast growth of urban areas, there is an imperative need for alternative solutions that allow for fast, automatic modeling of urban environments. In contrast, inverse procedural modeling converts an existing building model into an easy to synthesize procedural form. This methodology has been successfully applied in several architectural settings, and Aliaga et al. [2] presents a survey of such. We aim to introduce a general framework that explores existing models in various formats and facilitates parsing an architectural model and generates a compact, efficient, and re-usable procedural representation that can be used for structure-aware editing and synthesis of new models to satisfy the need of automatic or directed generation of urban spaces.

Some previous work (e.g., [3], [4], [5]) focus on forward creation algorithms that produce intricate geometry quickly from compact specifications and on inverse creation algorithms that infer the underlying geometric models from images, LIDAR, and other sensor modalities. On the other hand, previous inverse procedural modeling work of architectural structures has focused on facades, point clouds, buildings, or cities. Building point cloud approaches (e.g., [6], [5]) focus on segmentation and on symmetry analysis for finding repetitions. Inverse methods working with geometric building models mostly start with some assumptions. These assumptions can be having segments and labels of the model with structural constraints (e.g., [7]), having multiple exemplars of the same building style in different configurations, and having segments cutting along curves within symmetric areas limiting the ability to process arbitrary building geometries (e.g., [5]). Further, many of the above methods do not include integrated synthesis and editing tools.

III. A Generalized Proceduralization Framework

The key motivation behind this research is to create a pioneering new framework that is the first to provide automatic proceduralization for synthesis of arbitrary 3D architectural structures, by exploiting repetitions and organizing similarities into urban procedural models. We can consider the proceduralization problem as finding the lego pieces and their possible connections, making up a model. Our approach processes an architectural model provided as a polygonal model, a massive city model, or a point cloud, without the need of any hierarchy, constraint, or shape restriction. Our framework consists of several steps: First the geometry is divided into representative segments. Then the segments are labeled for classification. Afterwards, labeled segments are organized in a parse tree of the geometric instance to encode the spatial relations. Finally, rules, non-terminals, terminals, and patterns are inferred from the collection. The output is an instantiation of a context-free grammar including procedural hierarchies with encouraged reuse of grammar elements.

We propose a proceduralization framework (Figure 2) that converts 3D models into a procedural representation, which is (i) flexible to adapt to different types of models (i.e., meshes, point clouds.), (ii) independent of the granularity of the urban space (namely, applicable to facades, buildings, cities, as examples), (iii) adaptive to different metrics for the procedural model to represent, and (iv) useful for further procedural applications such as editing, compression, reconstruction, localization, and rendering.

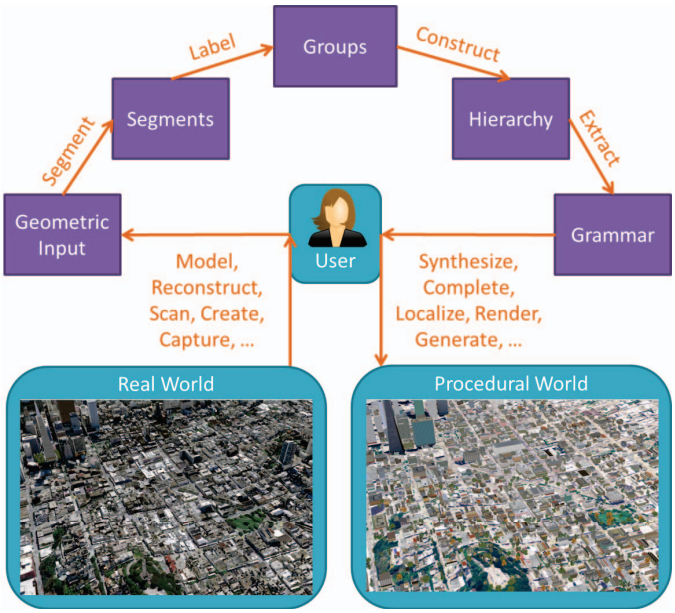

Fig. 2: **The Proceduralization Framework.** Our work provides a complete proceduralization framework that allows users to manipulate real world objects in a proceduralized world.

A. Shape Processing for Proceduralization

We can consider the proceduralization problem as finding the lego pieces and their possible connections, making up a model. Thus, to begin with, the models usually lack high-level grouping information which hampers efficient reuse and synthesis, concluding that shape-based segmentation of architectural models is a critical research problem. That search ended up in surprisingly diverse set of questions, like "What are the common properties of components?", "How to detect their similarities?", and "Can we improve by integrating segmentation and labeling?". We have observed that, although the problem itself is hard, it can be directed and optimized based on the needs of the grammar elements.

Specific solutions for the shape analysis phase of the proceduralization framework are introduced in our previous publications (Figure 3) (e.g., [8], [9], [10]). Our proposed methods extend the previously existing set of shape segmentation and labeling techniques by carrying the problem into a procedural domain where the properties on the segments are well-defined for the purpose of creating terminals of a representative grammar. This enhanced set of shape analysis methods is used to create the segments and repetitions mentioned in the general framework. In particular, we propose the following tools: (i) a semi-automatic approach for segmenting and labeling building point clouds; (ii) a method for finding components by rendering and clustering those components by a feature-vector based similarity labeling; (iii) a hybrid approach to couple the segmentation and similarity detection on architectural meshes.

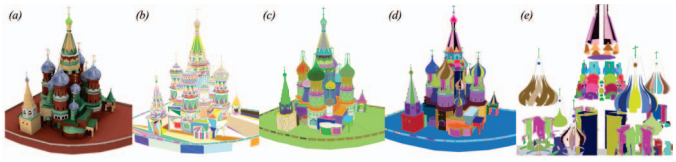

Fig. 3: **Shape Processing for Proceduralization.** (a) An input model (displayed as (b) color coded triangles), is partitioned into (c) search spaces, and segmented and labeled into (d,e) components automatically by our method [9].

B. Grammar Discovery for Proceduralization

We introduce specific solutions for the grammar discovery phase of the proceduralization framework (Figure 4) (e.g., [8], [10], [11]). Our proposed methods extend the previously-existing inverse procedural modeling techniques by eliminating the dependency on the assumption of any underlying rules or grammar. These independent grammar discovery methods are used to create the organization and the grammar steps introduced in the general framework.

In particular, we propose the following algorithms: (i) an automatic algorithm for converting the previously found segments into a procedural representation for architectural meshes that allows intuitive and interactive structure preserving editing on such models, (ii) a novel approach that converts the building point clouds into a procedural representation while providing completion of such models that refines reconstruction of such models, and (iii) a large-scale method to find the patterns and the grammar from the previously discovered components in a city that enables model synthesis, querying, and simplification of large urban areas.

Fig. 4: **Grammar discovery from components.** (left) Original input model versus new buildings synthesized by editing the procedural representation [11].

C. Proceduralization Applications

Our framework has been applied to obtain proceduralizations of various models, in various domains, under various scenarios, and for various aims. The goal of using the procedural representations in such systems is to benefit from the higher structural information and to provide, ease and improve structure preserving editing on various architectural models such as Saint Basil's Cathedral in Russia to Blue Mosque in Istanbul [11], completion and reconstruction (Figure 5) of building point clouds such as Staatsoper Hannover Opera House to Wilhelm Busch Museum in Germany [10], localization in metropolitan areas such as 180km2 metropolitan area of New York, and rendering and compression of huge detailed city models such as 4,000 buildings in New York, 2,500 buildings in Chicago, and 2,000 buildings in San Francisco, having millions of polygons [8].

We also show that a large procedural system that is able to procedurally re-create and simulate nine counties in San Francisco Bay Area [1] (Figure 1), which demonstrates to be a very useful large-scale application in urban planning and design. We have also developed an approach to create textured 2.5D urban models from aerial images, using volumetric reconstruction and surface graph-cuts [12], which can be considered as another proceduralization approach that inputs aerial images to create textured 3D models.

The key question to answer for all use-cases is how the collection of geometric entities that describe the spatial configuration of a model can be organized and represented efficiently, and how that efficient representation (namely, the grammar) can be used to automate generative processes. To accomplish

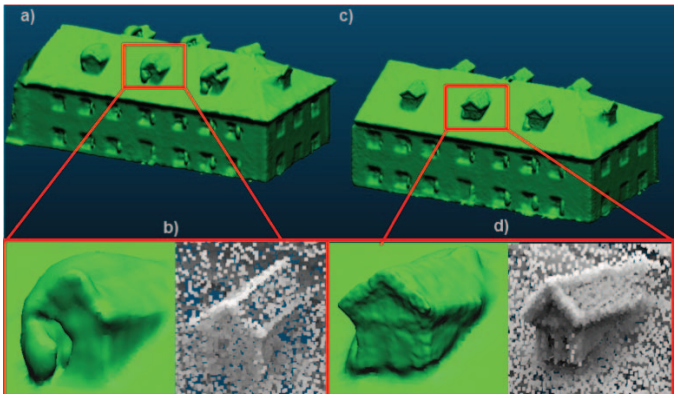

Fig. 5: **Completion via Proceduralization.** (a,b) Original model and (c,d) completed model after proceduralization [10].

answering this question, we explore and demonstrate how we applied our proceduralization framework to satisfy those different motivations.

IV. Conclusion

We have identified the following items of future work that we are interested in to our broader purpose of placing our research contributions as some of the key technologies to satisfy the automated content generation by proceduralization: (i) using neural networks to learn the grammar elements, (ii) carrying the proceduralization framework to other domains, and (iii) proposing evaluation tools for output grammars to assess their generative capacity.

In summary, our research shows how existing models can be proceduralized to free the artistic and efficient face of both the modelers and the computers, bridging the gap between manual and procedural modeling. As the graphics and vision communities converge in the automation phase for asset creation in almost all domains, many fascinating problems await to be solved by procedural approaches instead of spending the precious time of humans.

Acknowledgment

This research was funded in part by NSF CBET 1250232, NSF IIS 1302172, NSF CDS&E 1608762 and NSF EEC 1606396 awards. We also thank our modelers, users and open model databases for contributing to this research.

References

[1] P. Waddell, "Urbansim: Modeling urban development for land use, transportation, and environmental planning." *Journal of American Planning Association.*, 2002.

[2] D. G. Aliaga, İ. Demir, B. Benes, and M. Wand, "Inverse procedural modeling of 3d models for virtual worlds," in *ACM SIGGRAPH 2016 Courses*, ser. SIGGRAPH '16. New York, NY, USA: ACM, 2016, pp. 16:1–16:316. [Online]. Available: http://doi.acm.org/10.1145/2897826.2927323

[3] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool, "Procedural modeling of buildings," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 614–623, Jul. 2006. [Online]. Available: http://doi.acm.org/10.1145/1141911.1141931

[4] D. G. Aliaga, P. A. Rosen, and D. R. Bekins, "Style grammars for interactive visualization of architecture," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 786–797, Jul. 2007. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2007.1024

[5] M. Bokeloh, M. Wand, and H.-P. Seidel, "A connection between partial symmetry and inverse procedural modeling," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 104:1–104:10, Jul. 2010. [Online]. Available: http://doi.acm.org/10.1145/1778765.1778841

[6] J. Talton, L. Yang, R. Kumar, M. Lim, N. Goodman, and R. Měch, "Learning design patterns with bayesian grammar induction," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '12. New York, NY, USA: ACM, 2012, pp. 63–74.

[7] J. Lin, D. Cohen-Or, H. Zhang, C. Liang, A. Sharf, O. Deussen, and B. Chen, "Structure-preserving retargeting of irregular 3d architecture," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 183:1–183:10, Dec. 2011. [Online]. Available: http://doi.acm.org/10.1145/2070781.2024217

[8] İ. Demir, D. G. Aliaga, and B. Benes, "Proceduralization of buildings at city scale," in *3D Vision (3DV), 2014 2nd International Conference on*, vol. 1, Dec 2014, pp. 456–463.

[9] İ. Demir, D. G. Aliaga, and B. Benes, "Coupled segmentation and similarity detection for architectural models," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 104:1–104:11, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766923

[10] İ. Demir, D. G. Aliaga, and B. Benes, "Procedural editing of 3d building point clouds," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2147–2155.

[11] İ. Demir, D. G. Aliaga, and B. Benes, "Proceduralization for editing 3d architectural models," in *2016 Fourth International Conference on 3D Vision (3DV)*, Oct 2016, pp. 194–202.

[12] I. Garcia-Dorado, İ. Demir, and D. G. Aliaga, "Automatic urban modeling using volumetric reconstruction with surface graph cuts," *Computers & Graphics*, vol. 37, no. 7, pp. 896 – 910, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0097849313001131