Computers & Graphics xxx (2018) xxx–xxx

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Guided proceduralization: Optimizing geometry processing and grammar extraction for architectural models

Q1 İlke Demir*, Daniel G. Aliaga

Purdue University, West Lafayette, IN, US

A R T I C L E I N F O

Article history:
Received 1 November 2017
Revised 11 April 2018
Accepted 15 May 2018
Available online xxx

Keywords:
Inverse procedural modeling
Proceduralization
Reconstruction
Geometry processing

A B S T R A C T

We describe a guided proceduralization framework that optimizes geometry processing on architectural input models to extract target grammars. We aim to provide efficient artistic workflows by creating procedural representations from existing 3D models, where the procedural expressiveness is controlled by the user. Architectural reconstruction and modeling tasks have been handled as either time consuming manual processes or procedural generation with difficult control and artistic influence. We bridge the gap between creation and generation by converting existing manually modeled architecture to procedurally editable parametrized models, and carrying the guidance to procedural domain by letting the user define the target procedural representation. Additionally, we propose various applications of such procedural representations, including guided completion of point cloud models, controllable 3D city modeling, and other benefits of procedural modeling.

1. Introduction

The recent popularity of 3D environments and models for augmented and virtual reality environments puts high expectations on the complexity and quality of such assets, with a desire to replicate the real world. Urban planning, remote sensing, and 3D reconstruction researchers have been focusing on bringing the digitized and physical world together, with an emphasis on urban models. In parallel to the demand for city-scale 3D urban models, the availability of 3D data acquisition systems and image-based solutions have also increased. Although using the 3D data obtained from different sources such as images, laser scans, time-of-flight cameras, and manual modeling databases is an option, the results of these approaches usually do not expose an easily modifiable model with structural parts and thus obstructs architectural reconstruction and modeling tasks.

Aiming for automatic generation, procedural representations are highly parameterized, compact, and powerful, especially in urban domain [1,2]. The pioneering work of Parish and Mueller [3], and subsequent urban modeling papers (e.g., see surveys [4,5]) focused on forward and inverse procedural modeling approaches. While procedural modeling (PM) accomplishes providing architectural structures of required detail, creating complex realistic building templates needs time, extensive coding, and significant domain expertise. Inverse procedural modeling (IPM) addresses the shortcomings of procedural modeling by controlling and adapting the procedural generation to a given target model [6,7]. In this sense, IPM can be regarded as an optimization problem over the space of derivations, to guide procedural modeling.

We want to carry this problem one step further by (1) removing the dependency on the space of derivations, and (2) switching the control domain. *Proceduralization* [8] takes care of the first motivation, by converting existing geometric models into a procedural representation, with no a priori knowledge about the underlying grammar.

However, as this procedural representation aims to serve as the minimal description of the model, evaluating for the best description (e.g., the description with the best expressiveness) requires determining its Kolmogorov complexity, which is uncomputable. Our solution is to let the user guide the system to find the best grammar per use case. This also handles the second motivation by enabling the user to control the characteristics of the extracted grammar. In other words, inverse procedural modeling enhances procedural modeling by producing the best instance, while guided proceduralization enhances proceduralization by producing the best grammar.

In this paper, we focus on guided proceduralization for 3D urban modeling and reconstruction of architectural meshes, building point clouds, and textured urban areas. Our framework provides a feedback loop which under user control seeks out the hidden

* Corresponding author.
E-mail address: idemir@purdue.edu (İ. Demir).

https://doi.org/10.1016/j.cag.2018.05.013

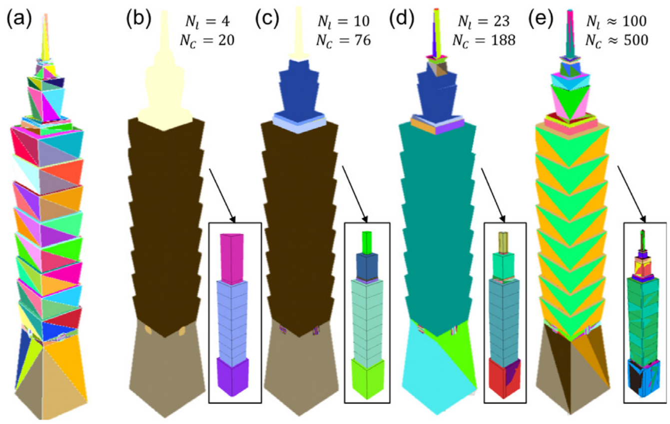

Fig. 1. *Guided proceduralization.* (a) The input model of *Taipei 101*, (b–e) colored procedural elements ordered by increasing target parameter values (e.g., number of components N_c and number of similarity groups N_l). As the user specification on the target grammar changes, different grammars of the model are revealed. Insets indicate representative instances of rules and terminals.

high-level hierarchical and structural information coherent with the target specification and the application objectives (Fig. 1). Although guided procedural modeling approaches [9,10], and proceduralization methods [11,12] have been introduced, we propose the first approach to guide the proceduralization process using specifications of a target grammar. We start with definitions and functions for guided proceduralization, then introduce geometry processing and grammar extraction steps of generalized proceduralization in the controlled setting. Afterwards, we demonstrate applications of the obtained grammars in completion, reconstruction, synthesis, modeling, querying, simplification, and rendering.

Altogether, our main contributions include:

- a generalized guided proceduralization framework that extracts procedural representations across different 3D data types,
- an optimization process to evaluate and diversify the grammars output by our proceduralization,
- a feedback loop to enable guidance to control proceduralization for obtaining the most expressive grammar and for various aims, and
- several applications of the guided proceduralization framework for editing and merging various models.

Using our controlled proceduralization framework, we have extracted procedural models from complex polygonal models (i.e., Turning Torso, Taipei 101, Saint Basil's Cathedral), from point clouds of architectural scenes (i.e., Staatsoper Hannover, Wilhelm Busch Museum), and from textured massive city models (i.e., 180 km^2 metropolitan area of New York with more than 4000 buildings, San Francisco, Chicago). We have used these models to create more complete reconstructions, controlled procedural generation, and easier, efficient, and structure-aware urban modeling.

2. Related work

2.1. Procedural modeling

Procedural modeling P generates a model M from a given grammar G ($P(G) = M$). Starting with the pattern language of Alexander et al. [13], PM has been utilized by many approaches [14–17] in urban settings. More PM approaches are surveyed by Smelik et al. [4]. However, coding procedural details is a cumbersome process needing domain expertise and codification skills. Interactive editing systems, such as Lipp et al. [18] and CityEngine, have been added on top of existing procedural systems to facilitate grammar editing. These can be considered as first guidance solutions for procedural systems.

2.2. Inverse procedural modeling

In contrast, inverse procedural modeling discovers the set of parameters, probabilities, and rules from a given grammar to generate a target instance [5] ($P(G, T_M) = M_{opt}$ as per the second row of Fig. 2). Initial works provided semi-automatic and automatic building (e.g., [19–21]) and facade solutions (e.g., [22–26]). Given some exemplar derivations and labeled designs, Talton et al. [27] use Bayesian induction to capture probabilistic grammars. Similarly, Monte Carlo Markov Chain (MCMC) optimization is used to discover the optimized parameters for a target instance of a procedural representation of buildings (Talton et al. [28] and Nishida et al. [29]) and cities (Vanegas et al. [30]). Most of those solutions support synthesizing similar models that best fit the given guidance. However they rely on pre-segmented components, known grammars, and known layouts to generate the derivation space. This is an important drawback, since it constrains reconstruction and modeling to the limited space of the initial grammar. In contrast, we want to carry the guidance from the geometric space to the procedural space, thus the *desired control* is defined rather than the *desired model*.

2.3. Proceduralization

Proceduralization starts with only geometry and no knowledge on the grammar ($P^{-1}(M) = G$ as per the third row of Fig. 2). Some image based techniques use deep learning for extracting simple and-or template grammars [31], or generative schemes [32]. In urban scenes, Bokeloh et al. [33] use partially symmetric structures to search for transformations that map one partition to another based on r-similar surfaces. It enables building model synthesis though not formally yielding a procedural model. For point clouds, Toshev et al. [21] segment a building into planar parts and join them using a hierarchical representation that separates roof, rest of the building, and non-building structures. Demir et al. [11] focus on point clouds, and user control is explicit in the geometric domain at the semi-automatic segmentation step. Martinovic et al. [6] use Bayesian induction to obtain facade grammars, and Kalojanov et al. [34] divide the input structure into microtiles to detect partial similarities. Demir et al. [12] introduces proceduralization, automatically creating a set of terminals, non-terminals, and rules (blue path in Fig. 3). Although evaluating the expressiveness of this automatic encoding is uncomputable (see Section 6.1), the expressiveness of a fixed grammar per model is still limited with regard to the modeler's use case. Thus, their approach has smaller tolerance for noise, directed by the one-pass segmentation and labeling, is not flexible for different use-cases, works only on triangular models, and does not allow user control (for the grammar generation).

2.4. Guided proceduralization

In contrast, the key motivation behind our research is that, if we have some insights about the desired grammar, we can evaluate the proceduralization outputs to suggest candidate grammars. This creates a pioneering framework that is the first to provide guided proceduralization for synthesis of arbitrary 3D architectural structures ($P^{-1}(M, T) = G$ as per the last row of Fig. 2). As guidance in procedural modeling enables finding *the best instance*, guidance in proceduralization enables finding *the best grammar*. This guidance enables the procedural representation to be optimized by user specification (orange path in Fig. 3), so that the resulting grammars are robust to model noise, flexible for different

Please cite this article as: İ. Demir, D.G. Aliaga, Guided proceduralization: Optimizing geometry processing and grammar extraction for architectural models, Computers & Graphics (2018), https://doi.org/10.1016/j.cag.2018.05.013