Computers & Graphics xxx (2018) xxx–xxx

Contents lists available at ScienceDirect

Computers & Graphics

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

Technical Section

Generalized selections for direct control in procedural buildings[☆]

Q1 Diego Jesus[a,*], Gustavo Patow[b], António Coelho[a], António Augusto Sousa[a]

Q2 [a] *INESC TEC and DEI/Faculty of Engineering, University of Porto, Portugal*
[b] *ViRVIG, Universitat de Girona, Spain*

ARTICLE INFO

Article history:
Received 7 August 2017
Revised 4 February 2018
Accepted 5 February 2018
Available online xxx

Keywords:
Procedural modeling
3D model
Semantics
Genetic algorithms

ABSTRACT

Procedural modeling techniques reduce the effort of creating large virtual cities. However, current methodologies do not allow direct user control over the generated models. Associated with this problem, we face the additional problem related to intrinsic ambiguity existing in user selections. In this paper, we propose to address this problem by using a genetic algorithm to generalize user-provided point-and-click selections of building elements. From a few user-selected elements, the system infers new sets of elements that potentially correspond to the user's intention, including the ones manually selected. These sets are obtained by queries over the shape trees generated by the procedural rules, thus exploiting shape semantics, hierarchy and geometric properties. Our system also provides a complete selection-action paradigm that allows users to edit procedurally generated buildings without necessarily explicitly writing queries. The pairs of user selections and procedural operations (the actions) are stored in a tree-like structure, which is easily evaluated. Results show that the selection inference is capable of generating sets of shapes that closely match the user intention and queries are able to perform complex selections that would be difficult to achieve in other systems. User studies confirm this result.

1. Introduction

Content creation remains one of the most important challenges in Computer Graphics today. One of the most common approaches is to use 3D modeling tools like Autodesk Maya or 3DS Max, which is time consuming, tedious and repetitive but gives the designer full control of the whole process. However, this approach scales badly with the number of assets to model, like in the case of a large virtual city. In that case, the most efficient solution for modeling urban landscapes is to use procedural techniques [1,2]. The introduction of a visual paradigm that allows the easy editing of the rule-set [3–6] has shifted the current trend towards simpler, yet effective, tools. Its results have been used for movies (using pioneering software like Esri's CityEngine [3]), video-games (e.g., with Epic's UDK [4]), and many more applications like urban planning, cultural heritage, or emergency evacuation planning, among others.

Notwithstanding, procedural modeling can be challenging, even for expert users, as there are no efficient solutions to create procedural models with grammar-based systems that also allow direct control at the same time. One of the open issues regards the ambiguity of user selections in a procedural model. Given a set of user selected architectural elements (e.g., some windows) there is a multitude of possible interpretations (e.g., windows in even floors or columns, on a specific facade, facing a given direction, etc.).

Buildings are particular structures that incorporate rich semantic content based on its architectural and engineering frameworks that derive from its function and shape. This has been explored in works such as BIM (Building Information Model) [7], an emerging technological and procedural shift within the Architecture, Engineering, Construction and Operations (AECO) industry involving the generation and management of digital representations of physical and functional characteristics of places [8], and also CityGML [9], that lead to the development of the current OGC (Open Geospatial Consortium) standard.[1] That information can be amplified in procedural modeling techniques, as the distinct elements that are generated interactively at each step have semantic and topological relationship with their predecessors, actually building a hierarchical structure (a shape tree) which can be further explored. Shape trees can also express fairly well a building structure in terms of hierarchical organization of its elements: as a wall is subdivided into floors and floors instantiate windows, there are implicit relationships that inform which windows belong to a

☆ This article was recommended for publication by B Benes.

* Corresponding author.
E-mail addresses: diego.jesus@fe.up.pt (D. Jesus), dagush@imae.udg.edu (G. Patow), acoelho@fe.up.pt (A. Coelho), aas@fe.up.pt (A.A. Sousa).

[1] http://www.opengeospatial.org/standards/citygml.

https://doi.org/10.1016/j.cag.2018.02.003

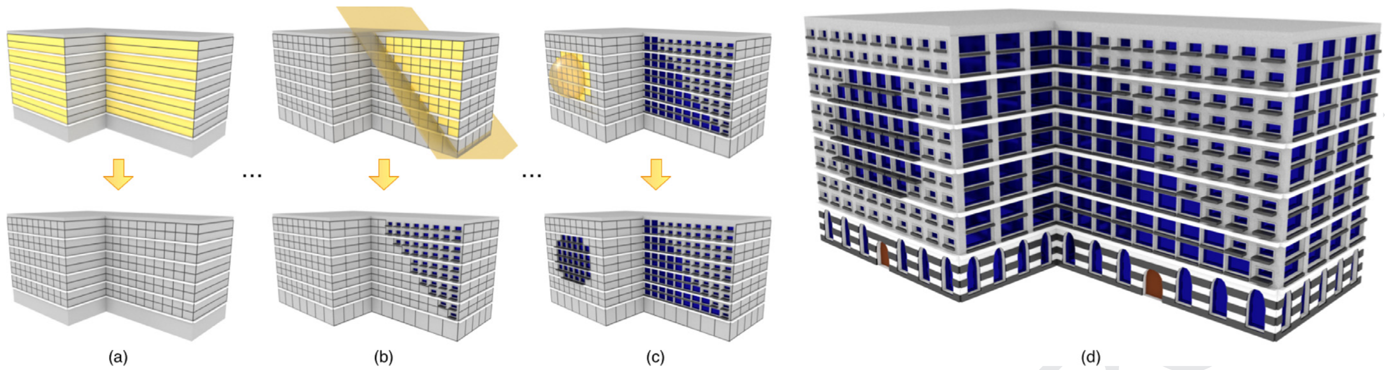

Fig. 1. Our system provides a mechanism to allow the user generate powerful generalized queries capable of selecting complex sets of shapes to which actions are applied. To generate the building on the right (d), the queries used include selecting all floors facing south (a), all floor tiles above or below a given plane (b) or those that are inside or outside a sphere (c).

certain floor or which are the windows of the front façade (those on the same wall as the front door).

In this paper, we present a generalized selection system for procedural building creation, that allows users to incorporate direct control in order to edit complex buildings. This methodology allows complex selections to be made over a shape tree enabling modeling complex configurations of shapes such as those present in Fig. 1. Our main contributions in this paper are:

- A system to infer queries from a user-provided point-and-click selection of building elements based on genetic algorithms.
- A selection mechanism based on queries over shape trees exploiting shape semantics, hierarchy and geometric properties.
- A complete selection-action paradigm that allows users to edit procedurally generated buildings without necessarily explicitly writing queries.

Genetic algorithms were chosen as they generate a broad range of candidates that explore the ambiguity of user selections without a known search direction.

Providing a procedural modeling methodology that relies directly on queries allows a manageable rule-set that can, more easily, accommodate changes.

2. Related work

In this section, we analyze some previous work in the field of urban procedural modeling and describe, with more detail, grammars and shape trees. One concept that is crucial throughout the paper is the shape tree, because the presented methodology fully exploits the implicit information that is present in such construct.

2.1. Previous work

The seminal work by Parish and Müller [10] about urban modeling, followed by the key work by Wonka et al. [1] and Müller et al. [2] about procedural buildings, produced a blossom in urban modeling research. All these efforts resulted in the origin of commercial packages, like Esri's CityEngine [3], or Epic's UDK [4], focused on, or with modules for, procedural urban design. More recently, Schwarz and Müller [11] extended shape grammars to make shapes and shape trees first class citizens allowing more complex constructions. The interested reader is referred to the surveys by Watson et al. [12], and Vanegas et al. [13] for an in-depth review of the state of the art literature in urban modeling.

Lipp et al. [14] presented one of the first attempts to improve editing operations for procedural buildings using an interactive visual system, which completely avoids editing text rules. Later, CityEngine [3], Epic's UDK [4] and Patow [5] independently developed visual representations for the rulesets, considerably easing the development process. However, tool development has not gone much further in this direction, reducing the user options to only a few simple operations [5]. Krecklau and Kobbelt [15] added specific rules selecting a given label with a given shape index to apply the different changes. In the context of urban models, Lipp et al. [16] also presented a solution for interactive modeling of procedural city layouts that allows intuitive manipulation using drag and drop operations. Also, a simpler type of distributed selections based on unsupervised learning as well as application of distributed actions to distributed groups of shapes across tree-hierarchies has been also introduced by Musialski et al. [17].

The introduction of layers in the procedural facade modeling pipeline was done by Li et al. [18], and later Ilčík et al. [19] introduced the automated merging of different layers in the form of a mixed discrete and continuous optimization problem in procedural modeling. More recently, Jesus et al. [20] proposed to use layers as an extension to shape grammars.

Leblanc et al. [21] have already introduced the concept of queries in procedural modeling. However, the queries presented there can be considered as a reduced subset of the ones allowed with our system. Also, Guerrero et al. [22] introduced a method to learn and propagate shape placements in 2D polygonal scenes from a few examples provided by a user, using machine learning techniques. Later, Guerrero et al. [23] proposed a method for propagating edit operations in 2D vector graphics, based on geometric relationship functions. Segmentation and selection have been addressed before, mainly by Bokeloh et al. [24], where the authors studied the relation between partial symmetry and inverse procedural modeling; Demir et al. [25], where problems of segmentation and similarity detection were studied in the context of architectural models; and Bokeloh et al. [26], where an algebraic model for parameterized shape editing was introduced. One of the main strength of the proposed technique is that it can be used in addition to all former techniques, as it increases the control of the process of generating building models and can benefit, as additional tool, to any procedural design system.

In the realm of interactive geometry creation, there are also applications that take user sketches as input. The work by Funkhouser et al. [27] allow the user to interactively segment and replace parts of a mesh with ones retrieved from a database. In procedural modeling, however, few are the contributions in this area. Recently, Nishida et al. [28] presented a system for the interactive design of procedural buildings using user-provided sketches and a machine-learning algorithm to match procedural snippets to the user sketch. As a result, only the trained examples are recognized.

Please cite this article as: D. Jesus et al., Generalized selections for direct control in procedural buildings, Computers & Graphics (2018), https://doi.org/10.1016/j.cag.2018.02.003