Grace Hopper Celebration of Women in Computing (GHC)
GFX Track - Research Presentation
Houston, Texas, USA, 14 - 16 October, 2015

Proceduralization of Urban Models

Ilke Demir
Purdue University
idemir@purdue.edu

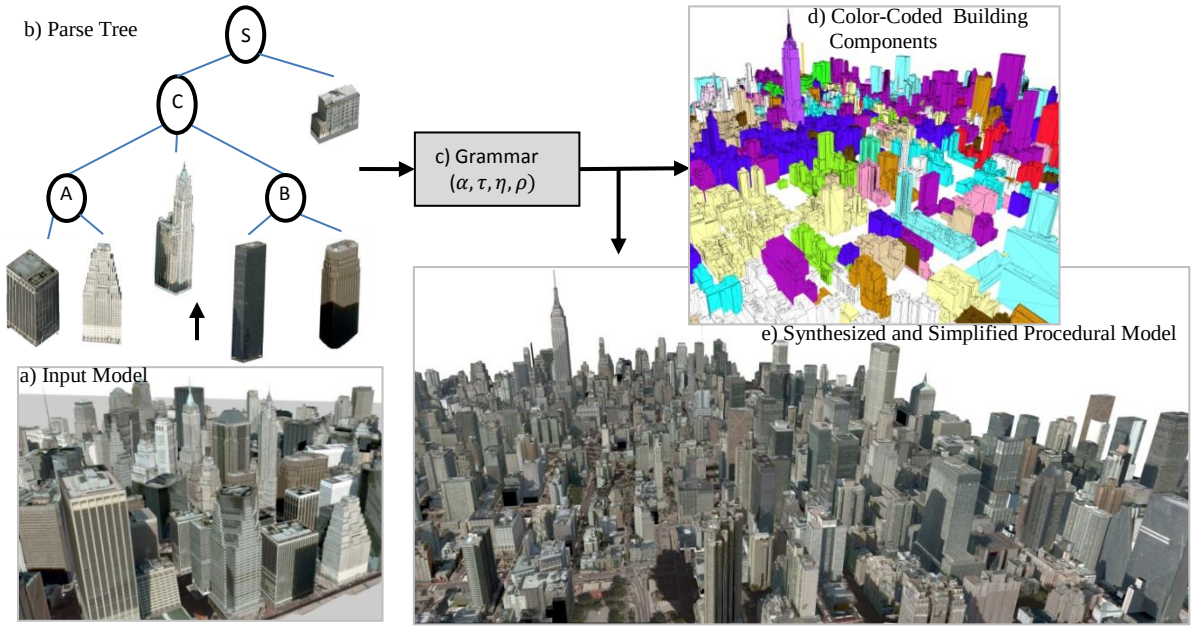

Figure 1. City-Scale Building Proceduralization*. Our approach uses a) an unorganized 3D model as input, b) computes a hierarchical clustering of building components, and c) extracts a context-free grammar of the urban area. We can procedurally generate (d-e) structurally-similar cities, at a chosen de-instancing level.*

ABSTRACT

An inverse modeling framework is presented for converting existing urban models into compact procedural representations that enable synthesis, querying, and simplification. During the de-instancing phase, dissimilarity clustering is performed to obtain the building components. During the proceduralization phase, the components are arranged into a context-free grammar, which can be edited text-based or interactively. Our approach proceduralizes several cities, up to 19,000 components over$180^2 km$, into compact city grammars.

AUDIENCE: [Computer Graphics/Computer Vision], [Procedural Modeling], [Urban Synthesis], [Interactive Editing Techniques]

[Intermediate Technical Talk]

INTRODUCTION

The demand for city-scale 3D urban models has significantly increased due to the proliferation of urban planning, city navigation, and interactive applications [6]. Manual and reconstructed models are detailed and realistic; but lack of structure causes inefficient editing, storage and rendering [3, 5]. While procedural models are known to be a powerful and compact parameterized representation, it requires significant manual expertise to procedurally code a city [1, 8, 9].

We present a framework for finding geometric components, repetition, and a high-level structural organization of a large collection of buildings and for creating a compact procedural representation that enables their synthesis, querying, and efficient rendering (Figure 1). Our input is an existing 3D city model represented as an unorganized but renderable collection of 3D buildings. Our automatic pipeline has two phases.

1) **De-Instancing:** A similarity-based clustering of instances of building components is performed to obtain a set of components and component types.
2) **Proceduralization:** Repetitive spatial patterns are discovered amongst the components and result in a compact context-free grammar applicable to synthesis, querying, and other procedural applications.

We have used our approach to convert parts of several large cities (e.g., New York, Chicago, and San Francisco) into procedural models, containing 10,000 to 19,000 building components. The resulting grammars can be directly edited (e.g., as a text file), interactively manipulated, and several types of information queries can be performed. Moreover, proceduralized models have been demonstrated to be friendly to very efficient GPU-based rendering [4, 7].

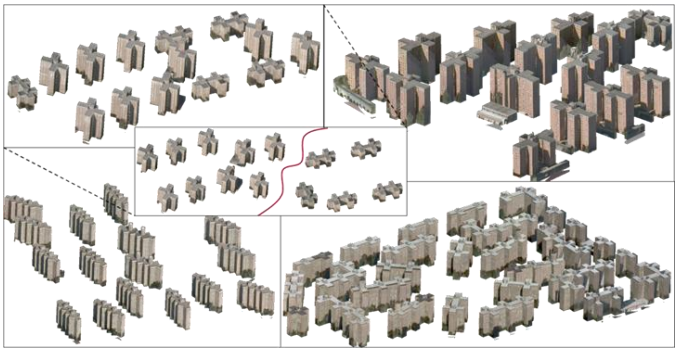

Figure 2. Building Clusters. Four building clusters computed automatically based on feature vector similarity. Inset: if the feature weights are adjusted such that building height is emphasized, the first cluster is divided into two separate categories.

Our main contributions include:

- a framework to convert an existing collection of 3D building models into a procedural representation,
- a feature-based de-instancing algorithm to determine building components and component types,
- a hierarchical clustering algorithm that finds repetition and spatial patterns of building components and enables extracting a compact context-free grammar.

METHOD

Our framework consists of two main parts:

1. De-instancing

This phase finds the terminal and non-terminal symbols of the grammar. First, a cut-plane estimation is employed to initialize the components by rendering a building and "cutting" it whenever a change is observed. Then, from each component, their geometric and visual properties are extracted into an eleven element feature vector. Finally, component types are determined by performing a dissimilarity-based k-means clustering of those feature vectors (Figure 2). The simplification level of the city depends on the de-instancing threshold level in the tree.

2. Proceduralization

Proceduralization phase starts with defining a distance metric between a pair of component types, based on their repetition, closeness, and consistency of instances. That distance metric is used as edge weights on the terminal types graph (Figure 3), which is then partitioned using Girvan-Newman clustering [2] iteratively to compute the hierarchical parse tree. After a pre-order traversal of the parse tree, current instantiation of grammar G is exported by exporting rules and placing a relative transformation matrix before each non-terminal/terminal.

RESULTS and DISCUSSION

Our approach enables a variety of applications, as well as compaction of models. Our framework reduces geometry and texture sizes by 2 to 21 times, producing grammars with 177 to a few thousand terminals and 5 to approximately 100 non-terminals. Preprocessing takes 2-4 hours for a city of 3,000 to 6,000 buildings.

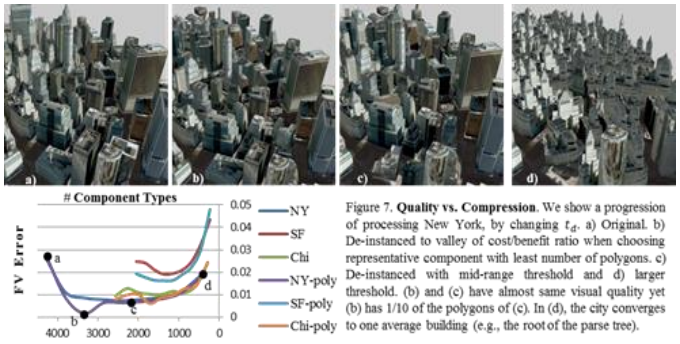

Figure 7. **Quality vs. Compression**. We show a progression of processing New York, by changing t_d. a) Original. b) De-instanced to valley of cost/benefit ratio when choosing representative component with least number of polygons. c) De-instanced with mid-range threshold and d) larger threshold. (b) and (c) have almost same visual quality yet (b) has 1/10 of the polygons of (c). In (d), the city converges to one average building (e.g., the root of the parse tree).

The graph in Figure 4a shows two cost/benefit curves for each or our models. The cost is the average standard deviation of all clusters (i.e., feature vector error) and the benefit is the number of polygons eliminated while de-instancing for various threshold values td. For the curve labels ending in "-poly", the representative component (per cluster) is chosen as the one with the least number of polygons for the other curves the representative component is the component closest to the mean of the cluster. The cost/benefit curves imply that a good trade off occurs near the bottom of the "u" shape. Moreover, the selection of the least cost representative components does yield an overall benefit (i.e., 10x fewer polygons at about same clustering error). Figures 4b-e demonstrate the model at various locations along the de-instancing progression.

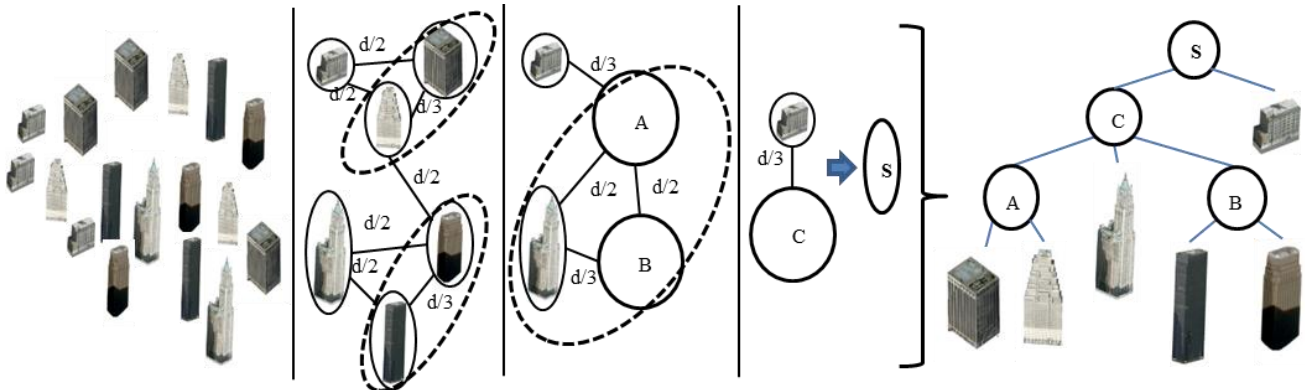

Figure 3. Proceduralization. a) The initial terminals in a city. b) The initial component type graph for Girvan-Newman clustering. For edge values $w_m = 1$ and $w_s = 0$ and for edges of buildings within "d" distance; we also do not show unique edges. Dashed ovals are the first clusters (i.e., nonterminals A and B). c) Another level producing nonterminal C. d) Final clustering producing the axiom S. e) The parse tree of the small city.

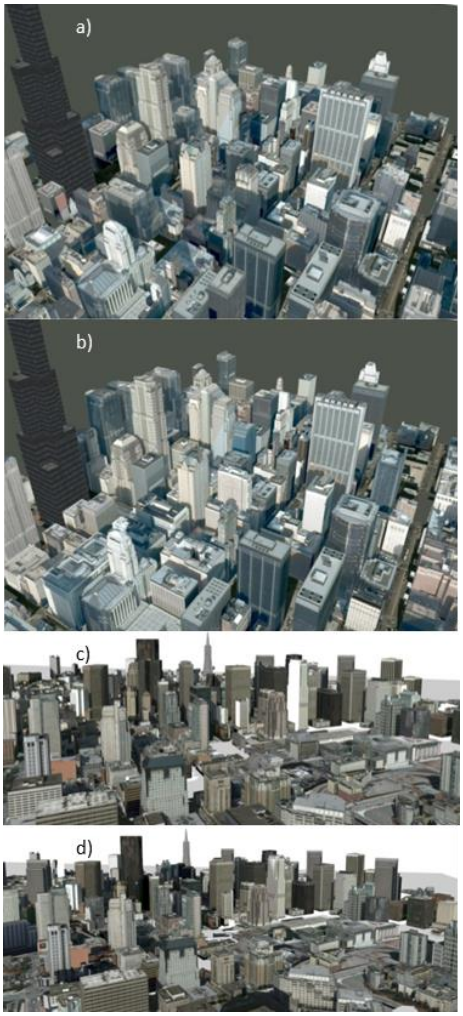

Figure 5. City Views. We show a-c) original Chicago and San Francisco, and b-d) de-instanced and proceduralized versions.

Figure 5 shows original versus proceduralized versions of Chicago and San Francisco, where the component types are reduces to ½ to ¼ times. Our framework also enables interactive synthesis of new cities, with the style of the original. Given a starting symbol (as "GHC" in Figure 6), our method best fills it with the grammar of New York and generates new content, while keeping the style and neighborhoods.

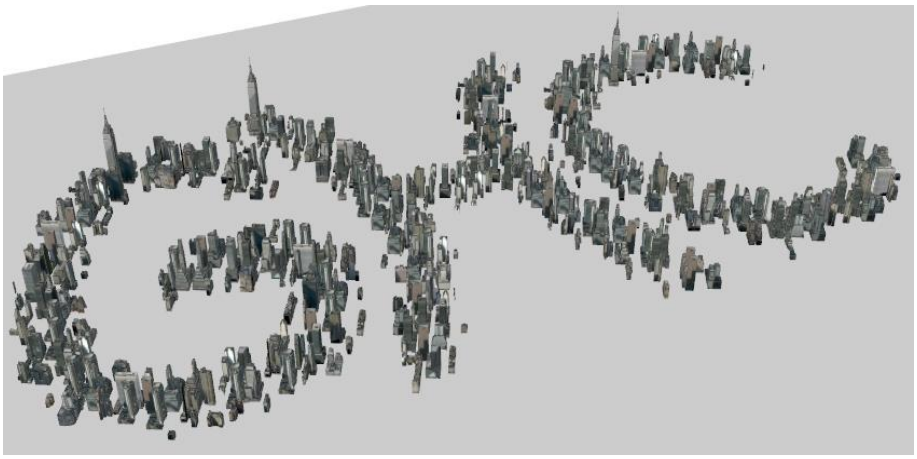

Figure 6. User-Controlled Synthesis. The user specifies a starting symbol ("GHC") and our method best fills it with the grammar and generates new cities.

PARTICIPATION STATEMENT

I will attend the conference if the submission is accepted.

BIO

Ilke Demir is currently a PhD Candidate in Purdue University, and a research assistant in Computer Graphics and Visualization Lab. Her research highlights include procedural modeling, 3D reconstruction and structure preserving interactive editing. She also obtained her M.S. degree in Computer Science from Purdue, and her B.S. degree in Computer Engineering from Middle East Technical University (METU, Turkey) in 2010 with a minor in Electrics and Electronics Engineering. Starting her research carrier in KOVAN Robotics Lab, she has also served as the student system admin of the department for 2.5 years, and did a software internship at Havelsan Inc.

REFERENCES

[1] M. Bokeloh, M. Wand, And H.-P. Seidel. A connection between partial symmetry and inverse procedural modeling, ACM Trans. on Graphics., 29(4), 2010.

[2] M. Girvan, And M. E. J. Newman. Community structure in social and biological networks, Natl. Acad. Sci. USA, 2002.

[3] B. Hohmann, U. Krispel, S. Havemann, And D. Fellner, CityFit: High-quality urban reconstructions by fitting shape grammars to images and derived textured point clouds, ISPRS Workshop, 8 pages, 2009.

[4] Z. Kuang, B. Chan, Y. Yu, And W. Wang. A Compact Random-Access Representation for Urban Modeling and Rendering, ACM Trans. on Graphics, 32(6), 172, 2013.

[5] F. Lafarge, And C. Mallet. Building large scale urban environments from unstructured point data, IEEE Intl Conference on Computer Vision, 1068-1075, 2011.

[6] R. M. Smelik, T. Tutenel, R. Bidarra, And B. Benes. A Survey on Procedural Modeling for Virtual Worlds, Computer Graphics Forum, 2014.

[7] M. Steinberger, M. Kenzel, B. Kainz, P. Wonka, and D. Schmalstieg. Combined Derivation and Rendering of Shape-Grammars on the GPU, Computer Graphics Forum (Eurographics), 10 pages, 2014b.

[8] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. Mech, And V. Koltun. Metropolis procedural modeling. ACM Trans. on Graphics, 30(2), 11, 2010.

[9] C. Vanegas, D. Aliaga, P. Mueller, P. Waddell, B. Watson, And P. Wonka. Modeling the Appearance and Behavior of Urban Spaces, Eurographics, STAR 17 pages, 2009.