Automatic Production and Visualization of Urban Models from Building Allocation Plans

Oğuzcan Oğuz, Medeni Erol Aran, Türker Yılmaz, Uğur Güdükbay
Department of Computer Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey
{*oguzcan,merol,yturker,gudukbay*}*@cs.bilkent.edu.tr*

Abstract

This paper presents a method for automatic generation of building models to be used in virtual city models and a system for the visualization of generated city models. The method incorporates randomness. The derivation process could be steered by the help of derivation rules and assigned attributes. The derivation method is inspired by shape grammars. During the derivation process, floor plans of the actual cities are used to generate 3D city models. The visualization system uses view frustum culling, back face removal, and occlusion culling to speed up the visualization.

1. Introduction

The original motivation for this work is to generate 3D city models using 2D city plans consistent with the real shapes of the buildings as much as possible and to visualize the city models in real time. Most countries have ground plans of the actual cities in digital format; these city ground plans are used to produce city models. The objects are first separated into subparts according to the pre-defined rules and the building model is produced from the final objects, such as walls, windows, balconies, etc. The shapes of the buildings are defined by specifying the rule set and the initial building models on which the rule set is to be applied.

The building models are represented using data structure to facilitate the visualization process. Three graphics acceleration techniques (occlusion culling, view frustum culling, back face culling) to achieve real time performance.

2. Previous Work

Previous approaches to the reconstruction of city models extensively use computer vision based techniques on aerial imagery and laser range scanners to extract the buildings and streets. While these methods produce excellent city models with high accuracy, they require post-processing to accurately model individual buildings in a detailed way.

Context-free grammars, mainly L-systems, are used to model streets [1, 2]. Derivation of detailed building models using split grammars is demonstrated to be highly successful [3].

3. Building Model Production

The city plans that are composed of building plans are given as input to the system and the system extracts individual building floor plans to generate buildings.

Produced building models are composed of several facades, one for each edge of the floor plan. Each edge of the floor plan is handled at a time. A facade that corresponds to an edge of the floor plan is composed of a number of floors. A facade could be composed of multiple copies of the same type of floor or different type of floors.

In the facade derivation process, a facade is split into floors that are actually two dimensional floor face objects. Then, predefined split rules are applied to these floor objects to generate new two dimensional temporary objects. This process continues until all paths end to a terminal object. A building is generated when all the edges of the floor plan are transformed to a corresponding facade.

3.1. Shapes

Currently, there are two types of objects defined in the system. Terminal shapes are the shapes that are defined in a DXF file and stand for the basic shapes that are designed by a 3D design program. Temporary shapes are the shapes that are split into other shapes by the rules defined in the configuration file. Terminal shapes are composed of any number of planar surfaces. They contain various attributes that are used to control the splitting process.

3.2. Rules

Temporary objects are split by the rules until all the objects become terminal objects. We could give higher probabilities to some rules if they are to be selected more frequently. Alternatively, we can use weights based on previous rule usage statistics to favor the rules that are frequently used. There are two types of split rules: *random split* and *fixed split*. Given a temporary shape, a random rule splits the object into a 2D grid that is composed of randomly placed rows and columns. Each new object can be a terminal shape or a temporary shape. Fixed rules split the given object to a fixed number of rows and columns whose sizes are defined in the rule.

4. Results of Building Production

Building models produced by using only height information and floor plans neither have enough details nor reflect the architectural style of the actual building. The proposed method allows fast generation of building models that reflect the intended architectural style.

Building models generated by using our approach could be seen in Figure 1. In our approach the floors of a building are not necessarily of the same type. A building model that is covered with textures is shown in Figure 2.

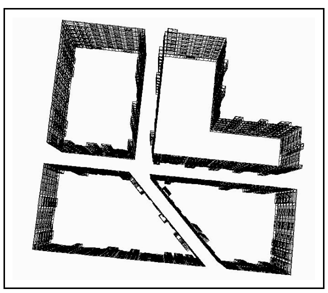

Figure 1. A block of four buildings.

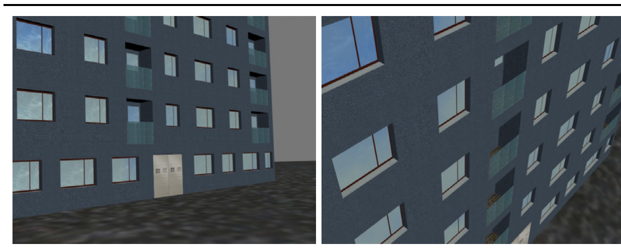

Figure 2. Two views of a building model covered with textures.

5. Visualization

City visualization systems strongly need culling of unnecessary data that do not contribute to the final image. The visualization system performs three graphics acceleration techniques (occlusion culling, view frustum culling, back face culling) to achieve real time performance.

In order to navigate correctly in an urban scene, we need navigation space information. Because of this, we developed an algorithm for the extraction of navigation space. The navigable space extraction algorithm represents the whole scene as an octree structure. The individual buildings are also represented by separate slice-wise clusters. Our navigable space extraction algorithm is described in [4]. Hardware occlusion queries are used for occlusion culling.

We can achieve real-time visualizations of city models containing a few thousands of buildings, each containing 2K-30K polygons, on an Intel Pentium IV-3.4 GHz. computer with 4 GB of RAM.

6. Summary and Conclusions

We present a system that is capable of producing building models to be used for populating virtual cities. Buildings are produced both deterministically and stochastically by means of footprints and shape grammars. The architecture of a virtual building depends on the architecture of the predefined (initial) objects and the rule set that will be applied during production.

For the visualization of the city models, we represent the buildings using an octree. To this end, we first represent a building using a regular subdivision and then we cluster the cells occupied by the building as axis aligned slices. This alternative structure provides better granularity for the visibility calculations as compared to visibility calculations where the individual buildings are regarded as visible or invisible.

We plan to generate class libraries capable of modeling different styles of architectural constructions. Then, it will be possible to model cities in a more realistic way.

Acknowledgments

This work is supported by the Scientific and Research Council of Turkey (TÜBITAK) under grant no EEEAG-104E029 and under grant no EEEAG-105E065.

References

[1] X. Decoret and F. X. Sillion, “Street generation for city modeling,” in *Architectural and Urban Ambient Environments*, 2002.

[2] Y. I. H. Parish and P. Müller, “Procedural modeling of cities,” in *Proceedings of SIGGRAPH’01*, 2001, pp. 301–308.

[3] P. Wonka, M. Wimmer, F. X. Sillion, and W. Ribarsky, “Instant architecture,” in *Proceedings of SIGGRAPH’03*, 2003.

[4] T. Yılmaz and U. Güdükbay, “Extraction of 3D navigation space in virtual urban environments,” in *Proceedings of 13th European Signal Processing Conference*, 2005.