Paper Evaluation: Procedural Modeling of Cities

1. Paper Title, Authors, and Affiliations

Title: Procedural Modeling of Cities

Authors: Yoav I. H. Parish, Pascal Müller

Affiliations: ETH Zürich, Switzerland; Central Pictures, Switzerland

2. Main Contribution

The main contribution of this paper is a procedural system, called CityEngine, that can automatically generate a complete virtual city from a small set of geographic and statistical inputs. The authors use L-systems and rule-based generation to create the road network, divide land into lots, and construct buildings with geometry and textures. The system takes image maps such as land water boundaries, elevation, and population density as input. Based on these maps, it first generates highways and streets, then subdivides blocks into allotments, and finally creates building shapes and facade textures. The authors also extend traditional L-systems to support both global goals and local constraints, which allows the road network to follow large-scale patterns while still respecting local conditions such as terrain and boundaries. The framework shows that large urban environments can be created procedurally and with limited manual work through procedural rules rather than manually modeling.

3. Outline of the Major Topics

The authors first introduce the difficulty of modeling large and complex cities and reviews related work in urban visualization and grammar based modeling. It then introduces the overall design of the CityEngine pipeline. The system first uses an extended L-system to generate highways and streets from input maps that describe population and geography. After the road network is created, the land is divided into blocks and further subdivided into smaller lots for buildings. Next, another stochastic and parametric L-system is applied to generate the geometry of different building types, such as residential houses, commercial buildings, and skyscrapers. The authors also describe a semi-procedural texture system that builds facade details using layered grids and reusable texture elements. Finally, the paper presents several results, including large city models with thousands of buildings, and discusses performance and possible future extensions

4. One Thing I Liked

One thing I liked is how the authors clearly compare their method with previous approaches in the related work section. They explain that image reconstruction methods can only rebuild existing cities but cannot create new ones. Visualization techniques focus on rendering rather than modeling. Shape grammars are not well suited for very large scale scenes, and pattern languages are difficult to automate. Space syntax mainly analyzes cities instead of generating them. This comparison makes the limitations of earlier work easy to understand and helps justify why a procedural generation system is needed.

5. What I Did Not Like

The system also relies heavily on user-defined inputs for many urban features. Elements such as parks and rivers must be specified manually and mainly serve as constraints for streets and buildings. This limits the level of automation and makes the system more suitable for users who already have a detailed plan and only need geometric modeling. Also, common urban elements such as docks, bridges, or other infrastructure are not generated procedurally. While the method works well for roads and buildings, the resulting cities can feel less lively and less realistic.

6. Questions for the Authors

1. Is it possible to design the system so that generation and simulation happen at the same time or visually no lag when users are using, allowing users to interactively modify inputs (such as population, zoning, or height maps) and immediately see how these changes affect the city layout?
2. If such real-time generation and simulation were added, would the computational cost become too high for large cities? How would you balance realism and performance?