Procedural 3D Modeling of Cityscapes

Ariane Middel
University of Kaiserslautern
Computer Science Department
D-67653 Kaiserslautern, Germany
middel@informatik.uni-kl.de

Abstract: The problem of modeling large-scale virtual urban environments has remained challenging for computer graphics researchers and urban planners. Cities are difficult to model in detail, since they hold diverse and complex geometries. Building large-scale 3D city models by means of photogrammetric reconstruction is a time and resource intensive, often semi-automatic process and does not provide data for modeling future cityscapes. Recently, a lot of research in the area of comuter graphics has been dedicated to developing alternative modeling techniques.
This survey provides an overview of state-of-the-art procedural modeling techniques for generating virtual cities. Research projects using grammar-based and agent-based models, statistical approaches, and real-time procedural modeling techniques are presented and discussed.

1 Introduction

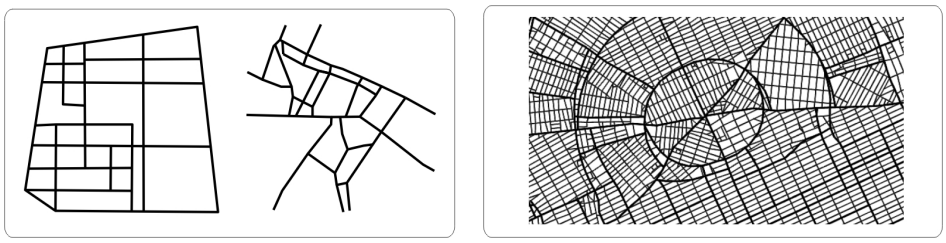
In consequence of ever-growing hardware performance and increasing user demands, automatic 3D city modeling has become more and more important for a number of application areas. In *architecture* and *urban planning*, 3D models are standard tools for visualizing urban design and development. Furthermore, virtual cityscapes are an important basis of decision making for participants in planning processes. They communicate future impacts of planning decisions and allow envisioning alternative futures. 3D city modeling is also applied in *environmental planning* where it is crucial for the spatial analysis and simulation of 3-dimensional phenomena like air pollution, noise, and earthquakes. *Computer graphics* make use of virtual cityscapes in 3D computer games and visualization applications, just as *entertainment industry* benefits from 3D urban environments in animated movie productions.
The application areas mentioned make the same demands on the 3D model generation: it has to be low-cost, fast, resource-saving, and automatic. In the past decade, research has focused on the 3D building reconstruction from sensor data using photogrammetric methods, active sensors, and hybrid systems [HYN03]. Photogrammetry provides techniques for constructing 3D city models from aerial photographs and terrestrial images. Using active sensors, building geometry is recognized from dense point clouds, e.g. acquired with the airborne laser scanning technology LIDAR (Light Detection and Ranging). Despite of

intensive research in these areas, both reconstruction methods do not automatically deliver full 3D city models yet [Bre05]. Hybrid systems combining several sensor-based methods like the CC-Modeler developed by CyberCity AG, Zrich [Ulm06] are promising but still semi-automatic approaches.
All things considered, the generation of large-scale 3D urban environments from sensor-based data has remained a significantly time-intensive and resource-consuming task. Furthermore, it is inapplicable for modeling urban futures, since it is exclusively based on real-world data.
For a lot of applications the appearance and the socio-statistical representativeness of an environmental model is more relevant then the geometric accuracy. Focusing mainly on land usage, building density or other socio-statistical parameters, the 3D visualization of look-alike cities is sufficient. This can be achieved with a technique known as procedural modeling. Procedural modeling utilizes code segments or algorithms to generate 3D geometries rather than storing an enormous amount of low-level primitives [Ebe96]. Creating abstract and formalized models is an efficient and flexible way to reduce the amount of stored data. Furthermore, the models are controlled by parameters in the generation process to increase the level of detail (LoD). Procedural modeling yields good results for repeating and random processes as well as for self-similar features like fractals. The fractal nature of cities is comprehensively discussed in Michael Batty's book "Cities and Complexity - Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals" [Bat05].

The following sections of this survey address different techniques for procedural city modeling, covering grammar-based models and agent-based techniques. After briefly introducing a statistical approach, real-time procedural modeling will be presented. Although not categorizable into the techniques previously examined, last-mentioned approach makes a major contribution to procedural modeling in terms of interactive real-time rendering. The survey concludes with a discussion of relative advantages and disadvantages between the classified techniques and gives an outlook on the perspective of procedural 3D modeling.

2 Grammar-based models

In the context of procedural city modeling, formal grammars have proven to be a powerful modeling tool. Recent approaches use so-called L-Systems for generating a variety of geometric elements in 3D city models [CdSF05]. An L-System or Lindenmayer-System is a parallel string rewriting mechanism that alters a string iteratively according to specified production rules. The resulting string can be interpreted geometrically to produce graphical output. L-Systems were conceived by Aristid Lindenmayer [Lin68] to describe the development of multicellular organisms. In the 90's, they have become a sophisticated computer graphics tool for simulating and visualizing plant geometry [PL90].
Kato et al [KOO$^+$98] are the first to reveal a substantial similarity between the growth of branching structures and the development of street networks. They introduce a virtual city modeling technique using stochastic parametric L-Systems to generate varying road networks. Their technique supports hierarchical street systems and can produce both linear



(a) Map and Tree L-Systems [KOO+98]

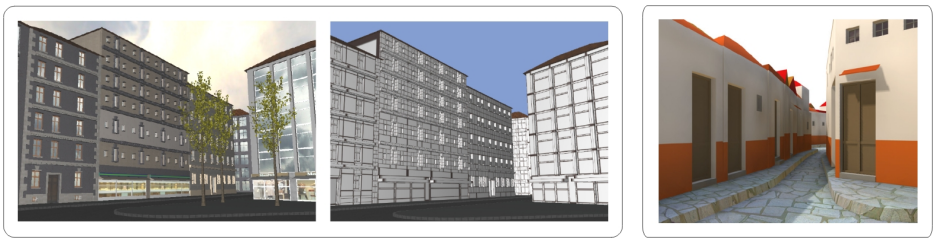
(b) Self-sensitive L-Systems [PM01]

Figure 1: Street networks generated with L-Systems

flow systems and cellular networks (see figure 1(a)).

The city modeling system CITYENGINE by Parish and Müller [PM01] incorporates an advanced street generation algorithm based on extended L-Systems. Unlike previous Lindenmayer-Systems, the enhanced grammar allows for the creation of closed loops and intersecting road branches. This is accomplished by adding self-sensitiveness to the nature of L-Systems. CITYENGINE employs a hierarchical set of production rules and enables the generation of streets that follow superimposed patterns. To derive a large-scale road map (compare figure 1(b)), geographical image maps on elevation, vegetation, and land-water boundaries as well as geostatistical maps on population density, zones, and land-use serve as input data. Since the introduction of CITYENGINE, L-Systems have been widely used for both reproducing existing street networks [GMB06] and creating fictional road maps [HMFN04]. Yet for modeling geometrically detailed buildings, L-Systems are difficult to adapt since they emulate growth-like processes in open spaces. CITYENGINE implements an L-System to generate simple buildings consisting of translated and rotated boxes. In this way, large urban environments emerge, but with a low resulting LoD.

INSTANT ARCHITECTURE is a technique developed by Wonka et al. [WWSR03] for automatic modeling of geometrically detailed buildings. Their approach uses parametric split grammars, derived from the concept of shape grammars [Sti80] which have been successfully applied in architecture to construct and analyze architectural designs. Split grammars operate with production rules consisting of geometric split operations. The idea is to generate geometrically rich 3D building layouts by hierarchically subdividing build-



(a) Instant Architecture [WWSR03]

(b) Roman Housing [MVUG05]

Figure 2: Architectural models generated with shape grammars

ing facades into simple attributed shapes (figure 2(a)). Wonka et al. set up a large grammar rule database to model various buildings in different architectural styles. INSTANT ARCHITECTURE yields to high LoD buildings, but is only applicable for small-scale urban areas.
Inspired by INSTANT ARCHITECTURE is the virtual 3D model of Roman housing architecture presented in [MVUG05]. Müller et al. reproduce ancient sites by extending the functional range of the CITYENGINE system to shape grammars like introduced in [WWSR03]. The production rules for the shape grammars are deduced from archaeological and historical data to ensure a faithful reproduction of Roman architecture. Plausibility is further enhanced by importing real building footprints and streets as ground truth.
Also integrated in the CITYENGINE framework is a sophisticated technique for procedural modeling of computer graphics architecture evolved by Müller et al. [MWH+06]. Their approach uses extended set grammars, so-called CGA SHAPES, combining the benefits of [WWSR03] and [PM01]. CGA SHAPES are suited for creating large-scale and at the same time geometrically detailed 3D cityscapes. Intrinsic context sensitive shape rules are applied sequentially to building footprints, the axioms of the productions, in order to generate mass models of the buildings. A mass model is a union of simple volumes and can consist of highly complex polygonal faces. In the next step, 2D building facades are extracted from the 3D shapes and structured into their elements. Here, CGA SHAPES re-use the volumetric information to solve intersection conflicts between adjacent facades. After adding details for ornaments, doors and windows the buildings are finally roofed with different types of house tops. In this manner, CGA SHAPES are applicable to model diverse urban areas like office districts, suburban environments and ancient cities (see figure 3).

Figure 3: Pompeii and Beverly Hills, modeled with CGA SHAPES [MWH+06]

3 Agent-based models

Agent-based models are computational models for simulating real world phenomena and are closely related to cellular automata and multi agent systems. The main modules of agent-based models are rule based agents, situated in space and time. They reside in artificial environments, e.g. virtual cities, are free to explore their surroundings, and dynamically interact with their environment and other agents. Simple transition rules drive the agents' behavior and result in purposeful, intelligent, far more complex reactions. For a

detailed introduction to computational agent-based modeling, particularly regarding multi-agent systems, and a broad introduction to automata-based urban modeling the reader is referred to [BT04].

Figure 4: Agent-based modeling of virtual cities [LWR+04]

Until recently, agent-based modeling has been largely restricted to 2D. For Instance, URBANSIM [Wad02] is a sophisticated 2D agent-based simulation model for large-scale urban environments that incorporates the interactions between land use, transportation, and public policy. Lately, systems have been designed to combine agent-based urban geosimulation with 3D visualization of urban environments. Lechner et al. [LWR+04] present an approach to procedurally generate virtual cities using an agent-based simulation. Their system only depends on a terrain description as low-level input and accepts optional parameters like water level and road density. Three types of agents are responsible for generating the tertiary and primary road network by exploring the virtual space. Primary road agents connect highly populated regions, extenders expand the existing street network to urban areas not serviced by a road, and connectors interlink poorly accessible areas with tertiary streets. To output a land usage map (compare figure 4) developer agents generate parcels and land use for residential, commercial, and industrial zones. Finally, the map is visualized using the SimCity 3000 graphics engine, as shown in figure 4.

4 Statistical models

As opposed to grammar-based and agent-based approaches, statistical models utilize statistical propagation techniques to procedurally generate urban environments. Statistical models are often employed to complement other procedural modeling techniques and are rarely used stand-alone. An example for the automatic generation of large geometric models based on statistical parameters is “A Different Manhattan Project” [YBH+02]. Yap et al. reconstruct the city of Manhattan based on the TIGER data set and diverse physical parameters like average size and height of buildings, zoning classification of land use, and dominant architectural styles. The parameters are statistically propagated over the city, using different parameter scripts for varying districts in order to capture the uniqueness of each neighbourhood. Landmarks such as the Empire State Building are hand-coded into the geometric model to accomplish a realistic view of the city skyline (see figure 5).

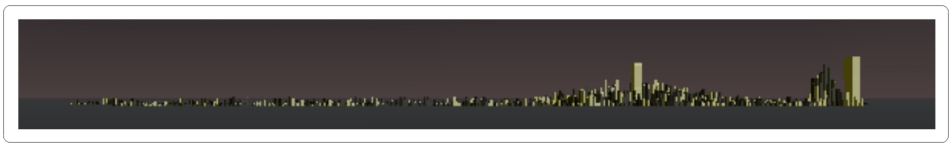


Figure 5: Statistical model of Manhattan [YBH+02]

5 Real-time procedural modeling

Besides believability and complexity of the scene, important characteristics for the rendering of procedural models in computer graphics are real-time and interactivity. While the modeling approach by DiLorenzo et al. [DZT04] is focused on the interactive animation of evolving cities, the technique introduced in [GPSL03] concentrates on dynamic geometry rendering in real-time. Greuter et al. have developed a method for generating pseudo-infinite virtual cities on-the-fly. Randomly generated regular polygons are merged into floor plans, extruded to parameterized buildings and textured, resulting in high LoD office districts. The model's building geometry is generated dynamically as needed inside the user's cone of sight while the user is interactively exploring the city. This view frustum filling (see figure 6) provides for the generation of pseudo-infinite cityscapes in real-time that would take a lifetime to explore.

Figure 6: Real-time generation of 'pseudo-infinite' city [GPSL03]

6 Discussion

Previous sections reviewed state-of-the-art techniques in procedural modeling, including grammar-based, agent-based, statistical, and dynamic models. The presented approaches mainly differ in the following characteristics (see figure 7 for comparison):

- scalability/LoD (large-scale models vs. architectural models)
- realness (existent cities vs. fictitious cities)

- dynamics (static vs. dynamic cities)
- input data (extensive vs. little input data)

These characteristics are conducive to the believability of the procedural model, whereas authenticity has a slightly different focus in different application areas. Besides visual fidelity, the realistic projection of socio-statistical and economic input parameters is a decisive factor for believability in planning applications. In contrast, gaming industry attaches importance to interactive, dynamic procedural modeling in real-time.

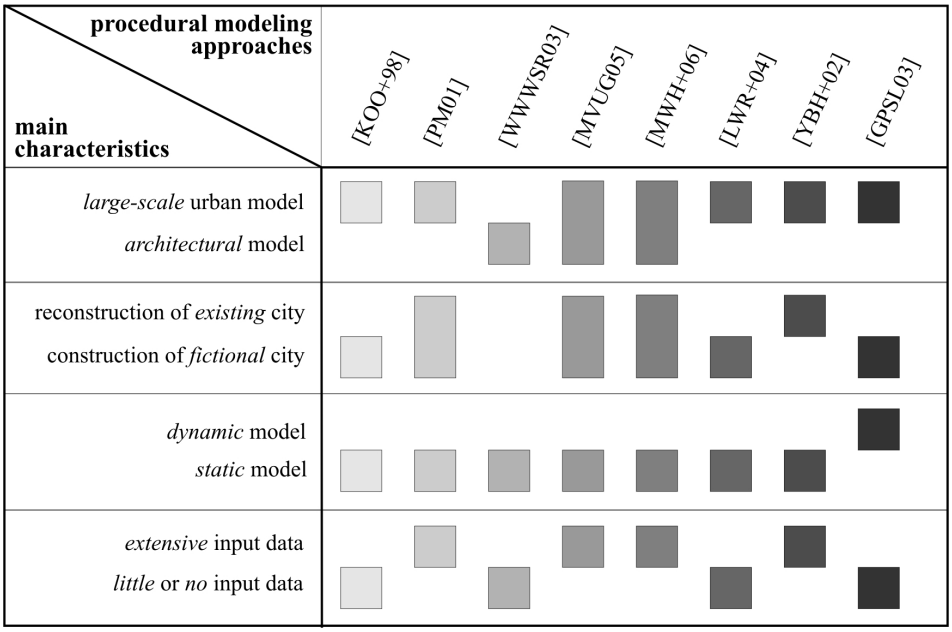


Figure 7: Predominant characteristics of procedural modeling techniques

Greuter et al. [GPSL03] mainly contribute to the applicability of procedural city modeling in computer graphics and visualization applications. They dynamically generate visually interesting and complex buildings in real-time with view frustum filling, still their interactive model is inapplicable for planning purposes. It only supports one building type, office skyscrapers, and the transportation network does not correspond to a realistic city, since streets are uniformly gridded. The procedural model of Yap et al. [YBH$^+$02] overcomes this drawback by including transportation ground truth, the TIGER data set. On that account, the implemented method only works for existing cities, not for future scenarios or fictitious cities. Apart from this, the "Different Manhattan Project" does not take into account the economic conditions of the modeled cityscape as well as the dynamics of residents' activities. In considering these aspects, the agent-based approach by Lechner et al. [LWR$^+$04] simulates the development of different land use zones Their model is suitable for planning applications where no socio-statistical and economical data is acces-

sible. The believability of the agent-based simulation benefits from a more realistic urban layout, whereas it suffers from an authentic road network with street patterns and visual attractiveness.
While Lechner et al. focus on land usage and building distribution, most grammar-based approaches aim at colonizing road networks. Kato et al. [KOO+98] generate road maps holding two different street patterns, whereas Parish et al. [PM01] enhance L-systems and implement production rules being easily extendible to several different street patterns. Their method is applicable for both reproducing existing cities and creating fictitious or future cities. Nevertheless, the L-system production rules presented do not allow a proper representation of the buildings' functionality. This considerable shortcoming is solved by INSTANT ARCHITECTURE [WWSR03], offering a variety of different architectural styles and designs for individual buildings. The complex geometric representation is at the expense of scalability, the approach only focuses on architecture and disregards urban layout and streets. Procedural modeling techniques combining large-scale models with geometrically detailed architecture are introduced by Müller et al. In [MVUG05] grammar-based techniques are combined to recreate ancient Roman cities, in [MWH+06] CGA SHAPES generate extensive urban models with up to a billion polygons. Both approaches offer high scalability as well as strong visual fidelity and are suited for the generation of ancient, existing, and future cityscapes. Yet, to visualize potential future impacts of planning decisions in urban planning applications, additional input data is needed. The models have to incorporate correctly projected demographic and economic data.

7 Conclusion

Procedural content generation is a promising technique in many application areas where the visual and socio-statistical believability of a model is more important than its geometric authenticity. Procedural models are useful for visualization and computer graphics applications, entertainment industry, and especially architectural, urban, and environmental planning. With continuously increasing hardware performance, the benefit from procedural modeling techniques will also increase. Future research issues, especially in the context of urban planning, include the integration of different procedural modeling techniques. A combination of agent-based simulation tools like UrbanSim with grammar-based approaches like CGA SHAPES will result in a comprehensive procedural modeling tool for the simulation and visualization of existing and developing cities.

8 Acknowledgements

This work was supported by the German Science Foundation (DFG, grant number 1131) as part of the International Graduate School (IRTG) in Kaiserslautern on "Visualization of Large and Unstructured Data Sets. Applications in Geospatial Planning, Modelling, and Engineering".

References

[Bat05] Michael Batty. *Cities and Complexity*. MIT Press, September 2005.

[Bre05] Claus Brenner. Building Reconstruction from Images and Laser Scanning. *International Journal of Applied Earth Observation and Geoinformation*, 6(3-4):187–198, March 2005.

[BT04] Itzhak Benenson and Paul M. Torrens. *Geosimulation: Automata-Based Modeling of Urban Phenomena*. John Wiley & Sons, July 2004.

[CdSF05] António Fernando Coelho, António Augusto de Sousa, and Fernando Nunes Ferreira. Modelling Urban Scenes for LBMS. In *Proceedings of the tenth international conference on 3D Web technology*, pages 37 – 46, 2005.

[DZT04] Paul DiLorenzo, Victor B. Zordan, and Duong Tran. Interactive Animation of Cities Over Time. In *17th International Conference on Computer Animation and Social Agents*, Geneva, Switzerland, 2004.

[Ebe96] David S. Ebert. Advanced Modeling Techniques for Computer Graphics,. *ACM Computing Surveys*, 28(1):153–156, March 1996.

[GMB06] Kevin R. Glass, Chantelle Morkel, and Chaun D. Bangay. Duplicating Road Patterns in South African Informal Settlements Using Procedural Techniques. In Stephen N. Spencer, editor, *Proceedings of the 4th International Conference on Virtual Reality, Computer Graphics, Visualisation and Interaction in Africa, Afrigraph 2006*, Cape Town, South Africa, January 2006. ACM.

[GPSL03] Stefan Greuter, Jeremy Parker, Nigel Stewart, and Geoff Leach. Real-time Procedural Generation of 'Pseudo Infinite' Cities. In *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Austalasia and South East Asia*, Melbourne, Australia, February 2003. ACM Press.

[HMFN04] Masanobu Honda, Kaznori Mizuno, Yukio Fukui, and Seiichi Nishihara. Generating Autonomous Time-Varying Viirtual Cities. In *Third International Conference on Cyberworlds (CW'04)*, pages 45–52, 2004.

[HYN03] Jinhui Hu, Suya You, and Ulrich Neumann. Approaches to Large-Scale Urban Modeling. *Computer Graphics and Applications, IEEE*, 23(6):62–69, November 2003.

[KOO+98] Nobuko Kato, Tomoe Okuno, Aya Okano, Hitoshi Kanoh, and Seiichi Nishihara. An Alife Approach to Modeling Virtual Cities. *IEEE International Conference on Systems, Man, and Cybernetics*, 2:1168–1173, October 1998.

[Lin68] Aristid Lindenmayer. Mathematical Models for Cellular Interaction in Development, parts I and II. *Journal of Theoretical Biology*, 18:280–315, March 1968.

[LWR+04] Thomas Lechner, Ben Watson, Pin Ren, Uri Wilensky, Seth Tisue, and Martin Felsen. Procedural Modeling of Land Use in Cities. Technical Report NWU-CS-04-38, Northwestern University, August 2004.

[MVUG05] Pascal Müller, Tijl Vereenooghe, Andreas Ulmer, and Luc Van Gool. Automatic Reconstruction of Roman Housing Architecture. In Baltsavias et al., editor, *Recording, Modeling and Visualization of Cultural Heritage*, pages 287–297. Balkema Publishers (Taylor & Francis group), 2005.

[MWH+06] Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural Modelingof Buildings. *ACM Transactions on Graphics (TOG), SIGGRAPH 2006*, 25(3):614–623, July 2006.

[PL90] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc., 1990.

[PM01] Yoav I H Parish and Pascal Müller. Procedural Modeling of Cities. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308, New York, NY, USA, 2001. ACM Press.

[Sti80] George Stiny. Introduction to Shape and Shape Grammars. *Environment and Planning B: Planning and Design*, 7:343–351, 1980.

[Ulm06] Kilian Ulm. The Virtual City. *Position Magazine*, pages 57–58, June/July 2006.

[Wad02] Paul Waddell. UrbanSim: Modeling Urban Development for Land Use, Transportation and Environmental Planning. *Journal of the American Planning Association*, 68(3):297–314, 2002.

[WWSR03] Peter Wonka, Michael Wimmer, Francois Sillion, and William Ribarsky. Instant Architecture. *Transactions on Graphics, SIGGRAPH 2003*, 22(3):669 – 677, July 2003.

[YBH+02] Chee Yap, Henning Biermann, Aaron Hertzman, Chen Li, Jon Meyer, Hsing-Kuo Pao, and Salvatore Paxia. A Different Manhattan Project: Automatic Statistical Model Generation. In *IS&T SPIE Symposium on Electronic Imaging*, San Jose, California, January 2002.