Volume 28 (2009), Number 2 pp. 1–18 COMPUTER GRAPHICS *forum*

Modeling the Appearance and Behavior of Urban Spaces

Carlos A. Vanegas[1], Daniel G. Aliaga[1], Peter Wonka[2], Pascal Müller[3], Paul Waddell[4], Benjamin Watson[5]

[1]Department of Computer Science, Purdue University, USA
[2]Department of Computer Science, Arizona State University, USA
[3]Procedural Inc., Switzerland
[4]Evans School of Public Affairs, University of Washington, USA
[5]Department of Computer Science, North Carolina State University, USA

Abstract

Urban spaces consist of a complex collection of buildings, parcels, blocks and neighborhoods interconnected by streets. Accurately modeling both the appearance and the behavior of dense urban spaces is a significant challenge. The recent surge in urban data and its availability via the Internet has fomented a significant amount of research in computer graphics and in a number of applications in urban planning, emergency management, and visualization. In this article, we seek to provide an overview of methods spanning computer graphics and related fields involved in this goal. Our article reports the most prominent methods in urban modeling and rendering, urban visualization, and urban simulation models. A reader will be well versed in the key problems and current solution methods.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism I.6.3 [Simulation and Modeling]: Applications

1. Introduction

Modeling the appearance and behavior of urban spaces is a great challenge. An urban space is a complex collection of architectural structures arranged into buildings, parcels, blocks, and neighborhoods interconnected by streets. Understanding, describing, and predicting the appearance (e.g., creating 2D/3D geometric models) and behavior (e.g., simulating urban development over time) of cities is useful in a growing number of applications. Traditionally, modeling urban spaces has been a rather manual task that consumes significant amounts of resources. With the growing requirements of quantity and quality in urban content, there is an imperative need for alternative solutions that allow for fast, semiautomatic urban modeling.

Urban modeling methods are important in a growing number of applications. Some of them are

- mapping and visualization - reconstructing existing urban spaces for mapping and navigation tools, visualizing previously-existing cities for which only partial data exists, and allowing architects to visualize a new city,
- entertainment - fast generation of detailed digital content for populating urban areas in video games and movies,
- emergency response - creating models to train emergency response personnel in current and speculative urban layouts, including planning evacuation routes for various catastrophes, and suggesting emergency deployments of resources, and
- urban planning - predicting outcomes of land use policies and their effect on existing neighborhoods, and creating hypothetical views of an urban space after applying development and growth algorithms.

1.1. Challenges

Urban spaces are difficult to model because the underlying structure is determined by a very large number of hard-to-quantify variables including land policies, market behavior, transportation infrastructure, governmental plans, and population changes. Moreover, dense urban environments are particularly complex to model because they are simultaneously dense and large, spanning from a few to hundreds of square kilometers. While works in computer vision and pho-

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

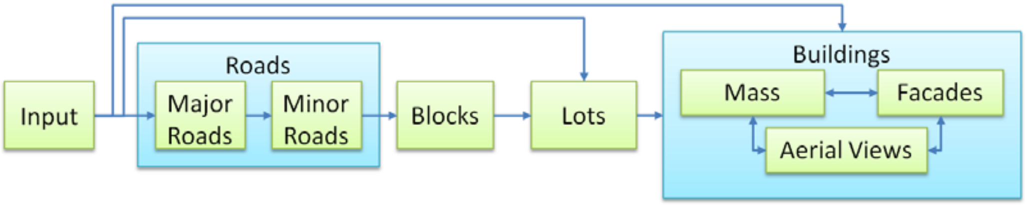

Figure 1: *General pipeline for modeling urban spaces.*

togrammetry have focused on the challenge of acquiring and reconstructing existing urban spaces (e.g., from LIDAR data, from aerial and terrestrial imagery [HYN03]), in this report we focus on the challenges of simulating, modeling, and rendering digital content of new urban spaces. As opposed to shorter term simulations of traffic flow and crowds in 3D environments, our use of the term simulation refers to imitating long term behaviors of an urban space such as urban development, the result of land use policies and the influence of the transportation network.

On one hand, the challenging problem of the 3D modeling and rendering of urban spaces is being tackled in computer graphics. We present a summary of efforts for automatic generation of visually appealing synthetic and real-world urban scenarios and for their efficient rendering. On the other hand, urban simulation and visualization are used by urban planners as a tool for decision-making regarding land use policies in current and projected urban areas. Such urban simulators generate large amounts of data that needs to be interpreted by decision-makers. The challenging problem of visualizing such data is addressed by several works that we also present. However, the above efforts have been largely independent, with computer graphics researchers focusing on complex and visually appealing 3D models, while urban planners focus on accurate urban dynamics and behaviorally-validated simulations. Thus, we also provide a short insight into a few new methods that address the challenge of bridging the gap between these two groups of approaches.

1.2. Table of Contents

The contents of this article are motivated by the recent proliferation of urban modeling publications in some of the top conferences and journals in computer graphics and related areas. We expect an even faster expansion of the field will take place within the following few years. Furthermore, the need for an integration of the multi-disciplinary efforts towards the simulation, modeling, and rendering of urban spaces is clear. Bearing this in mind, we believe that an article documenting the recent advancements and the forecasted and desired future work in urban modeling will be valuable to the research community. Thus, we have brought together researchers in urban modeling to produce a survey of the most prominent methods available and, consequently, to encourage other researchers to pursue further integrated research.

The literature review we provide includes works in the following major areas:

- urban modeling methods,
- rendering acceleration techniques, and
- urban simulation and visualization algorithms.

2. Modeling of Urban Spaces

In this section, we focus on recently developed methods for creating models of urban spaces. Although the categorization of such varied methods as those presented in this article is debatable, we have opted to group the methods based on (i) the part of the urban modeling pipeline (Figure 1) that they address and (ii) the dimension of the geometric entities that they produce. Table 1 presents a summary of the methods mentioned in this section. The urban modeling pipeline roughly consists of producing a road network, subdividing the blocks extracted from the road network into lots, and generating a building inside each lot. Different inputs to this process have been used including target architectural designs, example 3D models or imagery, socioeconomic datasets, and tensor fields. As an introductory framework for the presentation of urban modeling methods, general concepts behind procedural urban modeling are discussed in Section 2.1. Next, we present several techniques for modeling road networks and layouts in Section 2.2, which generally synthesize 2D geometric entities and aerial imagery. Last, several methods for procedural generation of 3D building mass and facades are presented in Section 2.3.

2.1. Procedural Modeling

The modeling of urban structures has been performed using several approaches including procedural modeling, synthesis methods, and other semi-automatic creation mechanisms. Procedural modeling has been used to automate the generation of complex urban structures, including buildings and houses, in order to produce digital content from a relatively simple set of parameters and rules (Figure 2). Synthesis methods have extended the concept of texture synthesis to 3D. Moreover, several forms of interactive editing have been

Method	Input	Output
Layout Modeling		
Hertzman et al., 2001	Small aerial image, small map, large map	Aerial image for the large map
Parish, Müller, 2001	User painted maps	Street network, facade shader
Aliaga et al., 2008	Example urban layout and aerial imagery	New road network and imagery with similar style
Chen et al., 2008	Interactive editing of a tensor field	Road networks
Weber et al. 2009	Simulation parameters, user interaction	Road network, parcels, land use information
Building Modeling		
Hahn et al., 2006	Building exterior, generation rules	3D geometry of building interior
Liu et al., 2006	Architectural mesh	Quad dominant mesh, offset mesh
Müller et al., 2006	Shape grammar rules	3D building and facade model
Aliaga et al., 2007	Facade images	3D building similar to example facade images
Pottman et al., 2007	Architectural surface	Beam and node layout, offset mesh
Merrell, Manocha, 2008	3D polygonal object	Larger collection of similar objects
Cabral et al. 2009	Mesh and textures, user interaction	Reshaped mesh and textures
Facade Modeling		
Legakis et al., 2001	3D polygonal mesh	Detailed 3D texture on the mesh
Wonka et al., 2003	Shape grammar rules	3D facade model
Haveman, 2005	GML rules, 3D geometry	Procedurally generated amplified geometry
Marvie et al. 2005	L-system rules, functions	3D facade models, 3D building models
Müller et al., 2007	Facade image, some user interaction	3D facade model, grammar for the facade
Finkenzeller 2008	Scripts, user interaction	3D facade models
Lipp et al., 2008	Interactive editing of shape grammar rules	3D building and facade model
Xiao et al. 2008	Multiple facade images, user interaction	3D facade models, textures

Table 1: *Urban modeling methods with respective input and output data, grouped according to the main urban modeling problem they address.*

proposed to further extend the aforementioned approaches and increase ease of use.

Figure 2: *Procedurally-generated urban models [PM01].*

Procedural architectural modeling can use one of several production systems such as Semi-Thue processes [DSWD94], Chomsky grammars [Sip96], graph grammars [HER99], shape grammars [Sti75], attributed grammars [Knu68], L-systems [PL91], or set grammars [WWSR03]. One aspect that guides the choice of production system is expressiveness. The first fundamental question is therefore: *how many different buildings can be modeled?*. If expressiveness is the main criterium it would be possible to just start modeling using C++ or Turing machines. This would enable us to compute all types of architecture and modeling would also be very flexible. However, there is also the question of efficiency: *how efficiently can a designer work with the framework?*. This second question makes simplifications and limitations more attractive. Otherwise, the complexity of the procedural model can quickly get out of hand and the design often becomes inconsistent.

In architecture, Stiny pioneered the idea of shape grammars [Sti75, Sti80]. These shape grammars were successfully used for the construction and analysis of architectural design [DF81, Dua02, Fle87, KE81, SM78]. The original formulation of the shape grammar operates directly on an arrangement of labeled lines and points. Rules of the derivation can basically be modeled by drawing lines and points and labeling them. In practice this leads to a derivation problem, because at every step of the iteration there are typically many different transformation under which a rule can be applied and additionally there are many different rules to choose from. Classical shape grammars therefore have a missing piece that prevents an automatic derivation: a control mechanism that selects which rule to apply and under what transformation. In Figure 3 we show an example of a shape grammar and one example derivation.

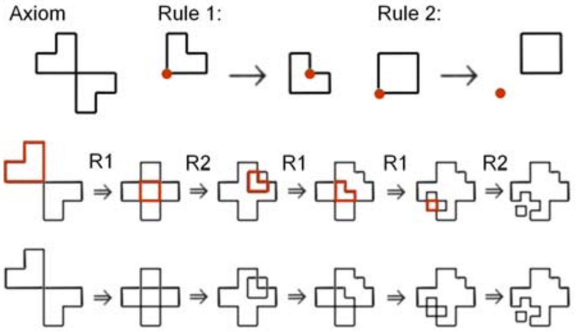

Figure 3: ***Shape Grammar.*** *This figure shows an example of a traditional shape grammar. The shape grammar operates on arrangements of lines. The first row of the figure shows the two rules of the shape grammar. The red dot is used to denote a reference coordinate system, so that the translation of the shapes used in the rule is defined. The second and the third row show a derivation of a new shape (right) starting from an axiom (the initial shape) shown to the left. Two rules are given: rule one and rule two. Which rule was chosen for a derivation step is written over the arrow: R1 means rule one and R2 means rule two. The red lines in the middle row highlight the subshape that is selected for replacement by the grammar.*

To make the shape grammar concept more applicable in computer graphics, Wonka et al. [WWSR03] and Müller et al. [MWH*06] introduced a framework that includes rules to replace shapes with zero, one, or multiple other shapes, as well as mechanisms to specify an automatic rule derivation. Only the automatic rule derivation enables large-scale procedural modeling. The framework by Müller et al. [MWH*06] was also further developed in the commercial software *CityEngine* [Pro08]. The original grammar by Wonka et al. [WWSR03] was focused mainly on size independent design rules of facades using splitting operations. The splitting operation allows breaking down elementary shapes (such as cubes and cylinders) by cutting the elementary shapes along splitting planes. The size independent design rules allow the designer to specify how the location of a splitting plane should change when the size of the elementary shape changes and how many splitting planes should be used for a shape of a certain size. In Figure 4 we show an example of the original split grammar proposed by Wonka et al. Other ideas presented in the original paper are several mechanisms for rule selection to ensure consistency while allowing interesting random variations.

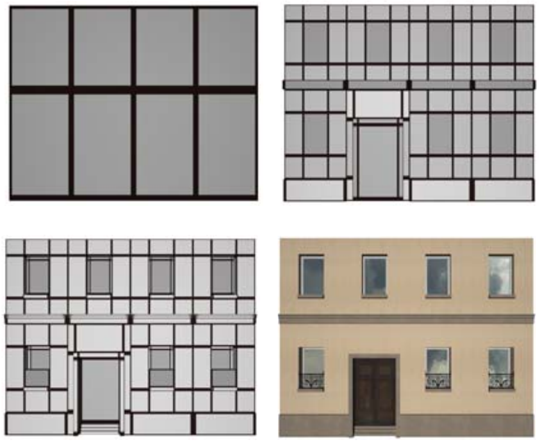

Figure 4: ***Split Grammar*** *[WWSR03]. An example derivation of a small facade using splitting rules. Note that the final model is three dimensional.*

2.2. Road Network and Layout Modeling

The modeling of roads and layouts has focused mostly on creating plausible aerial images and street networks for urban spaces. Similar to model synthesis, the creation of aerial imagery builds off the concept of texture synthesis (e.g., [WLKT09]) but in addition to the synthesis of pixel-data, associated vector data is also synthesized. The design of the streets themselves attempts to mimic the visual style of the street networks in real-world urban spaces.

Hertzmann et al. [HJO*01] introduced a two-phase design framework for image processing that can be directly applied to texture synthesis by example and is applicable to urban aerial imagery. In the first phase, a pair of images, with one image purported to be a filtered version of the other, is presented as training data. In the second phase, the learned filter is applied to some new target image in order to create an analogous filtered result. This method supports a wide variety of image filter effects including traditional image filters, super-resolution, improved texture synthesis, and texture-by-numbers. In the last of these applications, realistic scenes composed of a variety of textures are created using a painting interface. New imagery is synthesized by applying the statistics of a labeled example image to a newly labeled image. An interesting example of this works consists of synthesizing an aerial view of a city, as it was shown by the authors in their work (Figure 5).

More recently, synthesis approaches have been adapted to exploit the typical organization of an urban space. Traditional texture synthesis is not aware of the unique geometrical structure of an urban space (e.g., streets, parcels, building footprints). While the texture distortion can be measured and perhaps minimized (e.g., [SSGH01]), the resulting imagery

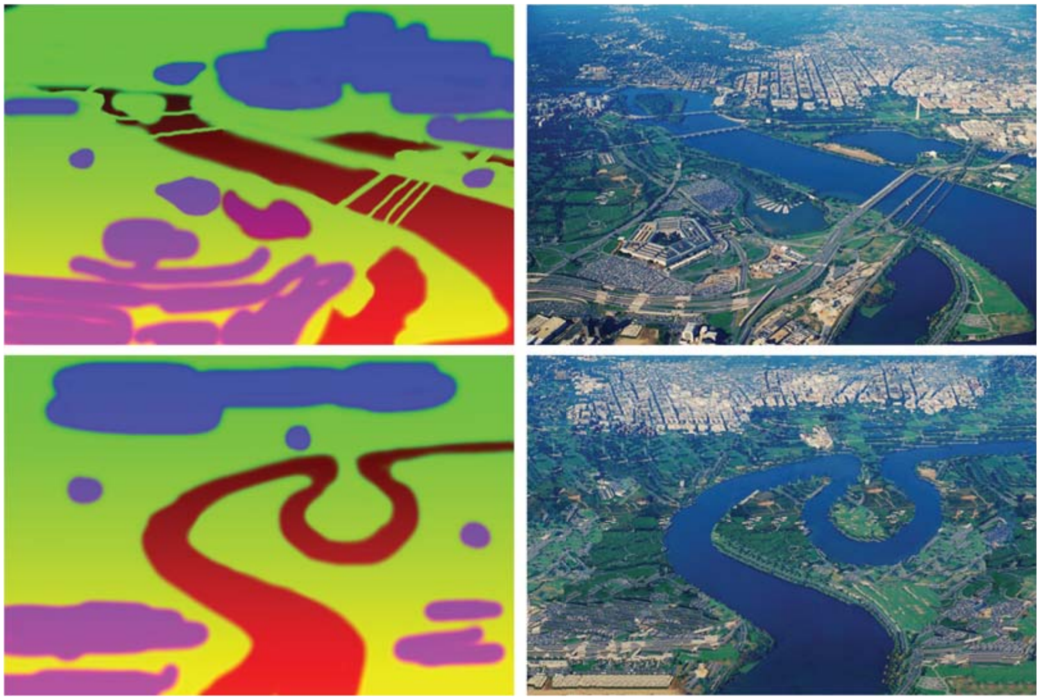

Figure 5: ***Image Analogies*** *[HJO*01]. Use of image analogies for synthesizing novel aerial views of urban spaces by example. Ordinary texture synthesis cannot reproduce the terrain in the photograph because it is not stationary: far elements are different from near elements. The images are courtesy of Aaron Hertzmann, University of Toronto, Canada, and the Corbis database.*

is not consistent and valid in the sense of possessing a plausible network of streets, parcels, and building footprints.

Aliaga et al. [AVB08] propose a method for example-based synthesis of urban layouts that is aware of the structure of an urban space. Their method uses as input a set of example urban layout fragments and simultaneously performs both a structure-based synthesis and an image-based synthesis to generate a complete urban layout with a plausible street network and with aerial-view imagery (Figure 6). Structure and image data from real-world cities are used by the synthesis algorithm to provide several high-level operations that can be used to interactively generate complex layouts by example. The user can create new urban layouts by a sequence of operations such as join, expand, and blend without being concerned about low-level structural details.

In related work, the same authors propose a method for interactive reconfiguration of urban layouts [ABVA08] (Figure 7). In that paper, the image of the urban layout can be changed, but the editing system is aware of urban structure. In particular, the method takes as input the vector data of the streets, blocks and parcels of the urban space, together with aerial-view images of the same space, and considers the connectivity and zoning of the parcels and streets. Several editing operations, such as expand, scale, replace and move, are supported. The urban layout is decomposed into a collection of adjacent tiles, separated by road or parcel boundaries. The specified transformation of the layout is performed by distributing the resulting global deformation among all the tiles, while preserving their connectivity and minimizing their individual distortion – similar to texture distortion minimization during the texture mapping process for complex objects.

Chen et al. [CEW*08] employ the concept of flow fields and tensor fields, as developed in some example-based texture synthesis methods (e.g., [KEBK05]), but rather use it to model the street layout of a city (Figure 8). They build on the observation that for many street patterns there exist two dominant directions due to the need for efficient use of space. Interestingly, tensor fields give rise to two sets of hyper-streamlines: one follows the major eigenvector field, and the other the minor eigenvector field. In their paper, Chen et al. introduce a modeling pipeline that consists of a tensor field modeling stage and a street graph generation stage. The tensor field modeling stage uses several modeling operations, including hierarchical editing, noise-based tensor field modification, smoothing, a brush interface, and the computation of tensor fields from topographical maps. The street graph generation extends existing streamline tracing algorithms to create a street graph. A visually plausible street graph has some constraints on the proximity of streets and on the number of dead ends.

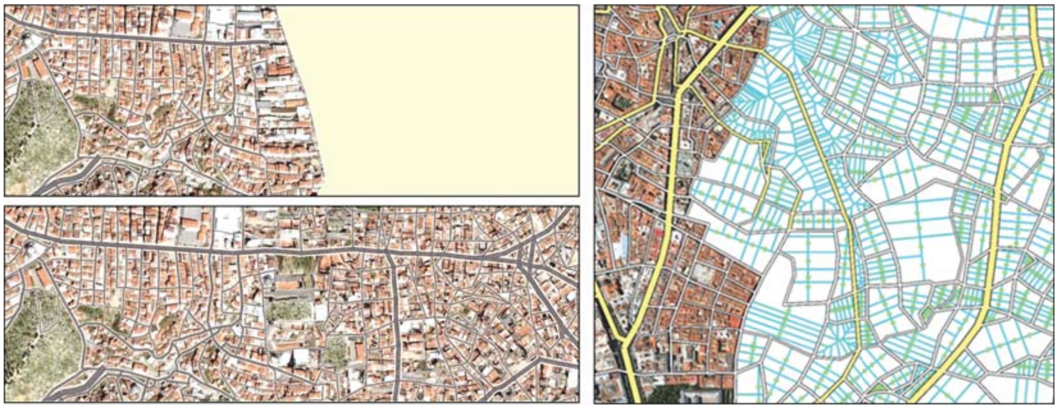

Figure 6: ***Example-based urban layout synthesis*** *[AVB08]. A new urban layout is generated by extracting and reproducing the structural attributes of the example fragment and reusing aerial-view imagery.*

In a related effort to synthesize detailed geometric features of a terrain, Bruneton and Neyret [BN08] addressed the problem of conforming a terrain shape to a set of vector features including roads, rivers, lakes and fields. Their approach is to combine a digital elevation model with layered GIS vector data to get precise feature boundaries and to enforce the constraints of the vector data on the terrain. GPU-friendly data structures and algorithms are proposed to allow for real-time editing and rendering of large terrains (Figure 9).

2.3. Building Modeling

Several works have specifically addressed the problem of generating 3D building models. Most of these methods use shape grammars for generating building mass and facades, and propose a variety of tools for efficient user interaction and controllability of the output geometry.

Müller et al. [MWH*06] built upon splitting rules, but they added several other components. First, they included shape operations for mass modeling (a kind of rough 3D sketch) by extending turtle commands used in L-systems [PL91]. The mass models are typically created by compositing several elementary shapes. This mass modeling is fairly intuitive and mirrors the actual design process used in architecture. Second, they introduced context sensitive rules for the coordinated derivation of different building masses. As a result, a wide variety of buildings can be generated procedurally. In Figure 10 we show renderings of the virtual reconstruction of the city of ancient Rome. Figure 11 shows an extrapolation of New York city 250 years into the future, inspired by the 1997 motion picture The Fifth Element.

Lipp et al. [LWW08] introduced the idea of interactive grammar editing. Instead of writing rules with a text editor, the framework of Lipp et al. makes it possible to design and edit rules entirely using a graphical user interface. This extension should make procedural modeling more accessible to a larger audience.

Recently, the idea of merging concepts from computer vision and procedural modeling has had some success. Müller et al. [MZWG07] use the idea of splitting rules to create a top down framework for facade analysis using a single orthographic image as input. First, important symmetry offsets are found and the corresponding pixels are collapsed onto each other until a small *irreducible facade* image is computed. Therein global optimization is used to find splitting lines to define rows and columns of facade elements. Further subdivision is done similar to the splitting rules introduced by Müller et al. [MWH*06] and finally shape grammar rule parameters can be extracted. This makes it possible to compute variations of the reconstructed building. Figure 12 shows an example of a reconstructed building facade from a single aerial image.

A related approach by Xiao et al. [XFT*08] also uses subdivision similar to how architecture is modeled with a grammar. In contrast to grammar-based modeling they use subdivision to oversegment the model so they also make use of merge operations. This work applies vision-based technology to reconstruct 3D facade models of high visual quality from multiple ground-level streetview images (Figure 13). Their method uses images captured along streets and relies on structure from motion to automatically recover camera positions and point clouds. The facade is initially considered as a flat rectangular plane with an associated image composited from multiple photos and is further decomposed and augmented using the 3D point cloud information. A key ingredient to creating the high quality models are interactive tools that allow the user to refine the model.

Aliaga et al. [ARB07] proposed a method to construct a grammar from photographed and subdivided buildings, en-

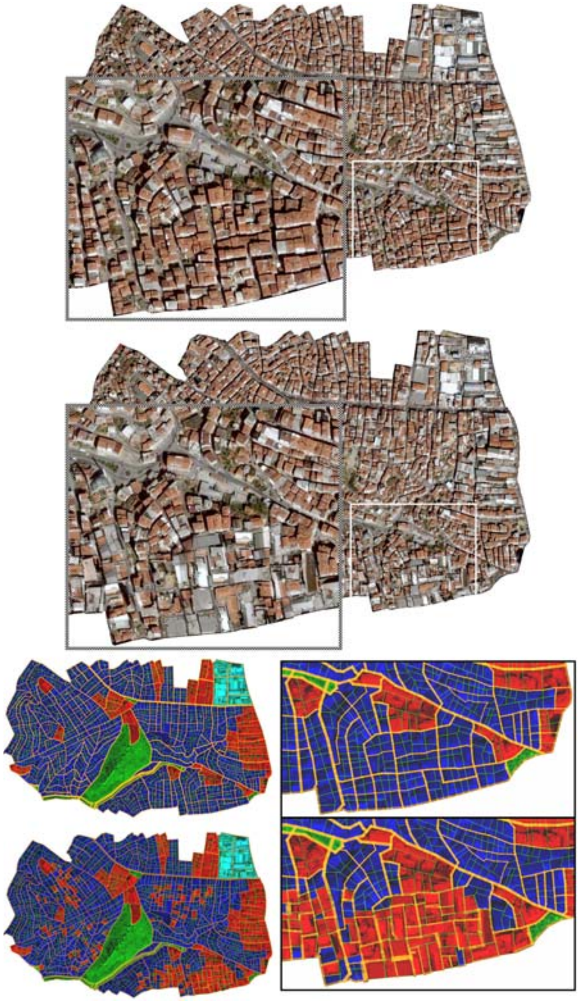

Figure 7: ***Interactive reconfiguration of urban layouts*** *[ABVA08]. Satellite images of an original and modified urban layout in which a residential zone has been converted to an industrial zone (top and middle). The process consists of recomputing the topology of the affected area to accommodate parcels of a new zoning type, and copying selected tiles from the industrial zone (red) of the city to the previously residential (blue) area (bottom).*

Figure 8: ***Interactive procedural street modeling*** *[CEW*08]. The concepts of flow fields and tensor fields are used to model the street layout of a city.*

abling the rapid sketching of novel architectural structures in the style of the original. Using data from several captured models, novel buildings can be designed very quickly and rendered photorealistically or non-photorealistically (e.g., pen-and-ink) but always in a style comparable to the original structures. Further, occlusion removal and color equalization algorithms make it possible to use highly occluded buildings in varying lighting conditions (Figure 14).

With regards to model synthesis, Merrell [Mer07] introduced a representative method for example-based 3D model synthesis. This approach can be used to create symmetric models, models that change over time, and models that fit soft constraints. A restriction of this first method is that the input objects must fit on an axis-aligned grid. In a second method [MM08], the connectivity between the adjacent boundary features of the input model is exploited to overcome the previous limitation, and models with arbitrary ori-

Figure 9: ***Real-time rendering and editing of vector-based terrains*** *[BN08]. Large terrains are populated with detailed features such as roads, rivers, lakes and fields in real time. The images are courtesy of Eric Bruneton, INRIA, France.*

entations are computed which have similar connected features and resemble the sample models (Figure 15).

While the previous papers focused more on high level modeling primitives and on designing the overall architectural structure, there are strategies better suited for modeling details and for modeling more general designs. A notable example for detail modeling are cellular textures [LDG01] that can be computed to assign brick patterns to building surfaces. A great example for a general and powerful modeling language is generative mesh modeling using GML introduced by Havemann in his Ph.D. thesis [Hav05]. GML allows the specification of commands that can refine and define a mesh.

There have been several other papers that propose alternative procedural modeling methods for architecture. Greuter et al. [GPSL03] present a method for generation of pseudo-infinite cities, in which all geometrical components of the city are generated in real time as they are encountered by the user. Marvie et al. [MPB05] propose some extensions to L-systems to make them more suitable for architectural modeling. Finkenzeller [Fin08] and Birch et al. [BBJ*01] introduce an interactive procedural modeling framework. Hahn et al. [HBW06] present a solution focusing on building interiors. Cabral et al. [CLDD09] present an approach for modeling architectural scenes by reshaping and combining existing textured models, where the manipulation of the geometry and texture are tightly coupled.

Another completely different method of computational design is necessary when the objective is to compute interesting free form surfaces that are popular in modern glass buildings that try to impress with geometric complexity. Liu et al. [LPW*06] use sequental quadratic programming to compute a quad dominant panel layout on a surface. This idea was extended by Pottmann et al. [PLW*07] and Pottmann et al. [PSB*08] to include more general layouts. Another interesting work is the computation of beam layouts by Smith et al. [SHOW02] for truss structures.

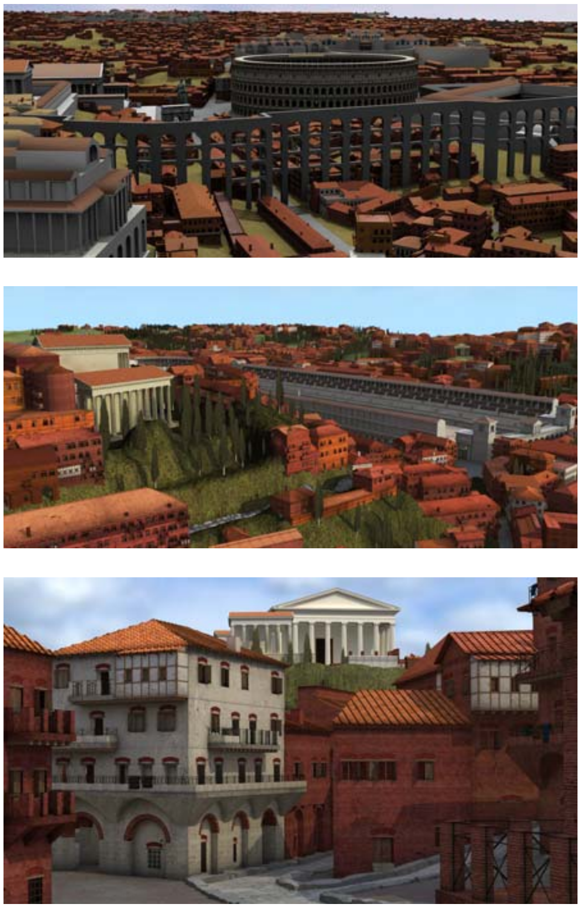

Figure 10: ***Rome Reborn.*** *Two renderings of the ancient Rome reconstruction consisting of more than 7000 procedurally generated domestic buildings. The landmarks such as the Colosseum (top) and the Circus Maximus (bottom) have been modeled manually. The images are courtesy of Bernard Frisher, IATH and Procedural Inc.*

3. Rendering Acceleration Techniques

We briefly review methods for rendering acceleration of large urban environments divided into the following categories: mesh data structures, visibility culling, and simplification.

Modern graphics hardware requires that the models are processed in a specific format. Typically indexed data structures are used to cache per-vertex computations and geometry has to be batched together in larger data structures, as the rendering of individual triangles is inefficient. Additionally,

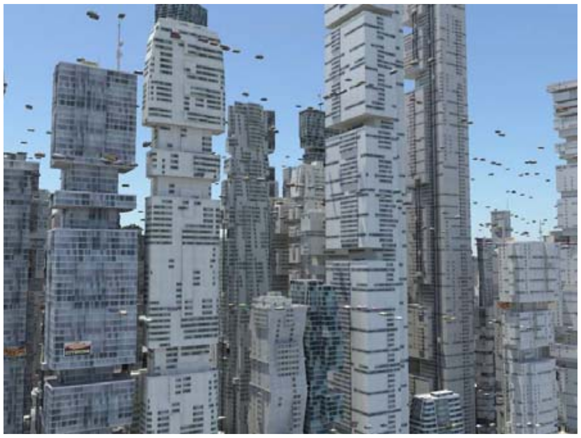

Figure 11: ***New York City 2259.*** *Procedurally generated buildings from the New York city model 250 years into the future. The images are courtesy of Procedural Inc.*

rendering needs to minimize changes such as switching of textures and shader programs. Hoppe introduced one of the first algorithms to reorder triangles to efficiently utilize the vertex cache [Hop99]. Other interesting ideas are to extend this concept to order the geometry to be additionally aware of pixel-level occlusion culling [SNB07], or to create cache-oblivious mesh layouts [YLPM05]. In general, these basic optimizations are beneficial for all interactive applications. In the context of urban environments these data-structure optimizations are beneficial for finely tesselated architectural details, but not for urban mass models with a few planar polygons and most details stored in texture maps.

Besides the well established and easily implementable techniques of back face culling and view frustum culling, occlusion culling can provide speed-ups of several orders of magnitude for most large and dense urban scenes. This is because an urban environment usually lies on a locally flat surface and the nearby urban structures easily fill the field-of-view and prevent observing distant structures. Occlusion culling either can rely on a precomputation or can be computed online. To precompute visibility, the navigable space is broken down into smaller volumetric view cells. Researchers have tackled the question of how to efficiently compute visibility for volumetric view cells and proposed multiple algorithms [SDDS00, DDTP00, WWS00, LSCO03, NS04, WWZ*06]. The advantage of precomputation is that occlusion culling needs very little runtime overhead and that the results of precomputation can be used for other preprocessing algorithms, such as level-of-detail selection and rendering time estimation. The disadvantage are the complexity of the algorithms and the long preprocessing times. Online visibility typically computes visibility for each frame of an interactive simulation from the current view point. In a seminal paper Greene et al. [GKM93] introduce the *hierarchical z-buffer* algorithm. The algorithm allows the identification of large parts of the scene that are guaranteed to be occluded and that is especially suited to rasterization based rendering. Currently, a complete implementation of the hierarchical z-buffer is not supported by graphics hardware, but the gap is closing. Today's graphics hardware includes several useful features for occlusion culling, including pixel level

Figure 12: ***Image-based procedural modeling of facades*** *[MZWG07]. A building facade image that is used as input to the algorithm (top). A wireframe overlayed over the original image (middle). The resulting 3D model rendered using relighting and shadow maps (bottom).*

Figure 13: ***Image-based facade modeling*** *[XFT*08]. 3D facades are reconstructed from multiple ground-level streetview images. The images are courtesy of Tan Ping, National University of Singapore, Singapore.*

Figure 14: ***Style grammars for visualization of architecture*** *[ARB07]. The user creates and subdivides an initial model of the building (top left). Repetitive patterns of the building features are automatically found and a representative grammar is constructed (top right). The user can then view the captured model (bottom left) and change the model on the fly producing new models (bottom right).*

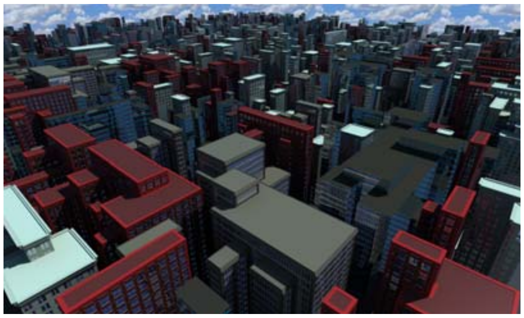

Figure 15: ***Continuous Model Synthesis*** *[MM08]. All these building models are automatically generated from a single example model. Different textures are applied to the buildings, but the shape of each building resembles the shape of the input. The images are courtesy of Paul Merrell, University of North Carolina, USA.*

occlusion culling to avoid expensive shading operations and a hierarchical z-buffer with a few levels (e.g. three) to discard smaller tiles of pixels during rasterization. The book by Möller and Haines has a good overview of the current state of the art [MH02]. Another important feature are occlusion queries that can test bounding volumes for occlusion. A major challenge in this context is to hide the latency of the occlusion queries and to utilize temporal coherence [BWPP04, GBK06, MBW08]. In all cases, when the viewpoint is far above the city occlusion culling has often little impact and other strategies need to be used that focus on simplifying the environment.

An interesting challenge is to compute geometric levels-of-detail (LOD) for urban models. There are many strong methods for general mesh and object simplification. A good introduction is the book by Luebke et al. [LRC*02]. Two fundamental early papers introduce operations to collapse edges [Hop96, GH97] in triangle meshes. While these methods work well for objects that contain smooth surfaces modeled with many triangles, there are substantial obstacles when applying them to urban environments. Leaves in vegetation and planar structures with many sharp corners are common, resulting in discontinuities that cause many problems for traditional level-of-detail techniques. Some special purpose methods have been invented for building footprints. For example, Chang et al. [CBZ*08] describes a clustering based simplification method for urban spaces inspired by Kevin Lynch's Image of the City book, but in general a high quality automatic simplification of urban environments is an unsolved problem. The main aspect of urban environments that can benefit from level-of-detail is the terrain. Early approaches, e.g. [LKR*96], built on the assumption that triangle rendering is expensive and therefore the algorithms were quite sophisticated. Current algorithms try to incorporate the fact that larger batches of triangles need to be rendered at once and focus on algorithm simplicity [LH04] and large memory management [GMC*06]. For architecture and vegetation, simple alternatives to automatic simplification are to create multiple versions of each building or tree procedurally [MWH*06] or by hand. Further, procedural modeling can help simplification, by providing semantic data. For example, it is very helpful to know what the actual facade planes are and what geometry belongs to one facade. Finally, since man-made structures are hard to simplify with geometric methods ([WFM01]), another option is to use image-based approaches. Several recent papers try to explore a hybrid rendering architecture where rasterization provides the rough object outlines and the fragment shader implements a ray tracer to generate details [POJ05]. An adaption to urban environments are block maps to ray cast collections of buildings [CDG*07] or facade displacement maps to ray cast facade details [AYRW09]. More general image-based techniques simplify scene parts and replace them by images often called impostors. An online version of this algorithm idea can dynamically create and store image-based representations in a hierarchical data structure [SLS*96, SS96], and offline versions can obtain guaranteed frame rates [AL99, JWSP05] and be integrated with level of detail and occlusion culling [ACW*99]. The advantages of image-based simplification is that it is significantly more robust than geometric simplification and it also works for difficult cases such as many disconnected plant leafs. The disadvantage of impostors are the high storage requirements, some challenges in recomputing the shading of the computed representations, and the quick generation of high quality representations. In summary, there is no simple solution to simplifying all aspects of urban environments.

4. Urban Simulation and Visualization Algorithms

Urban simulation models and the visualization of computed datasets are used to help regional planning agencies evaluate alternative transportation investments, land use regulations, and environmental protection policies. The simulation models typically output massive spatially distributed data about several variables, including number of inhabitants, land prices, and traffic. Urban simulation systems (e.g., [WBN*03]) generate predictions of real estate development, prices, and location choices of households and firms at fine-grained levels of geography such as grid cells or parcels, over entire metropolitan areas, and over planning horizons of up to 30 years. The amount of data generated by such a microscopic model over a long forecasting horizon and a large scale is overwhelming for users to easily interpret. Visualization techniques are essential to be able to render useful information from the mass of data generated by such simulations.

4.1. Urban Simulation

Urban simulation refers to the use of behavioral or process modeling of the the spatial patterns of urban economic agents and objects such as jobs, population, housing, and land use. In Figure 16 we present a generalized framework for describing the behaviors that urban simulation models attempt to represent in varying degrees of comprehensiveness and with differing approaches to temporal abstraction and detail of agents and location [Weg94, Weg04]. It highlights the interaction among household and business agents that are locating in housing and nonresidential buildings, and on the differing time scales of the evolution of buildings, transportation networks, urban form, and travel that connects the agents within the urban system.

Three dominant paradigms for creating urban simulation models have appeared in the field. Early models attempting to represent emergent dynamics adopted cellular automata (CA) as the modeling framework [TO01]. One of the most widely known is the Urban Growth Model [Cla98]. It has been applied to long-term changes in land cover patterns classified from remote sensing data [AkWS08]. Unfortunately, this modeling approach only simulates the conversion of non-urban land to urban use, based on the characteristics of cells and their immediate spatial context, and does not address changes to the built environment or its occupants, or the travel that connects agents.

Agent-based models (ABM) have extended the CA framework to include mobile, interacting agents in an urban spatial context. This work has focused on examining cities as self-organizing complex systems, and solutions have been designed to explore the emergent properties of agents with relatively simple behavioral rules embedded by the modeler [Por00]. However, relatively little attention has been paid to issues of validating models using observed data or trends, and as with CA models, most ABM urban simulation models have behavior that is influenced only by localized context.

An alternative approach to urban simulation has emerged from a combination of urban economic analysis with statistical modeling of choices made by agents in the urban environment, such as households choosing residential locations. This work builds on the pioneering work of McFadden on Random Utility Theory [Mcf74] and the development of discrete choice models, for which he recently won the Nobel Prize in Economics. Research using this approach diverges on the dimensions of temporal representation and level of aggregation. Aggregate models represent agents by grouping them into types, and locations into large zones, whereas microsimulation models represent individual agents such as households and jobs, and objects such as buildings and parcels. Similarly there are contrasting approaches to the representation of time, with earlier research focusing on equilibrium in a set of equations of locating agents and buildings, and later work exploiting a dynamic representation that uses explicit chronological time and incorporates path-dependence. Examples of the aggregate and equilibrium approach to urban simulation include spatial interaction models [Put91] and equilibrium discrete choice models [AK96, Mar96].

An example of more recent work in this area using a dynamic microsimulation approach is UrbanSim [Wad02, dPPW07], which simulates the choices of individual households, businesses, and parcel landowners and developers interacting in urban real estate markets [WUFL07]. This approach works with individual agents as is done in the ABM, with very small cells as in the CA approach, or with buildings and parcels. But it differs from these approaches by integrating discrete choice methods, an explicit representation of real estate markets, and statistical methods to estimate model parameters and to calibrate uncertainty in the model system [ŠRW07].

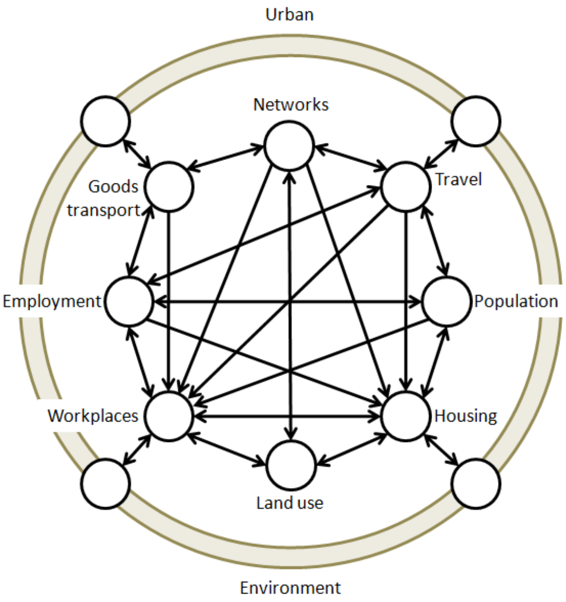

Figure 16: *Diagram of the interactions in urban simulation models (recreated from [Weg94]).*

4.2. Urban Visualization

Visualization and computer graphics have played an integral part in the development and use of urban simulations of several types. A number of works have focused in developing novel visualization techniques for better understanding the results of urban simulation models.

Several groups of population with different levels of expertise in handling urban simulation data are normally interested in these results, including urban planners, policy-makers, the public, or even the modelers running the simulation. On one hand, traditional information visualization techniques have focused in handling large urban simulation data sets and making their analysis more intuitive to urban planners. On the other hand, recent research works have proposed an interdisciplinary collaboration between computer graphics, visualization and urban modeling to produce new visualization techniques of urban simulation data sets. These techniques aim to facilitate the presentation and increase the impact of urban simulation data to different population sectors.

Traditional visualization approaches generally make use of techniques including choroplethic (colored) maps generated by exporting simulation results, summarized by a zonal geography, to a Geographical Information System (GIS) for rendering; other variants include animations generated by rendering a series of such 2D maps in a loop, viewing different time slices or quantities, and 3D renderings of simulation results by extrusion of polygonal forms to indicate density, or by spatial smoothing in the form of contour or terrain maps with the elevation representing some quantity of interest.

Batty [Bat92] first introduced various approaches that relate urban modeling, GIS, and computer graphics. The same author later described the impact of virtual reality and 3D visualization to GIS and he has demonstrated this on a variety of complex examples [BC97]. More recently, Batty presented a comprehensive view of urban dynamics in the context of complexity theory [Bat07].

While there has been a large amount of work on GIS, very little research has been done evaluating the usefulness of other types of visualizations for this domain. A study by Pinnel et al. [PDBB00] examines various visualization types and attempts to find appropriate visual representations for urban modeling tasks. The types of visualizations considered include graphs, pie chart, 2D and 3D maps, symbol charts, and bubble charts. They cross-reference each of these types with the encodings that can be effectively utilized (e.g., color intensity, bars, area/height, marker size, marker shape). Their study concludes that for urban planning and analysis, map type visualizations provide the necessary geographical information, while for quantitative tasks bar charts and summaries better present the needed information.

A widely used urban visualization technique is cartograms which use map shape warping to visualize relationships and values of urban and geospatial datasets (e.g., [KNP04]). The core idea behind cartograms is to distort a map by resizing its regions according to a statistical parameter, but in a way that keeps the map recognizable.

Chang et al. [CWK*07] propose an aggregation method that combines buildings and city blocks into legible clusters. Their goal is to visualize an urban model in a focus dependent and multi-resolution fashion, while retaining the legibility of the city. In their approach, the 3D model view and the data view are integrated so that relationships between the geospatial information of the urban model and the related urban data (e.g., census information) can be more intuitively identified. While the user-study that they conducted showed that some features introduced by their system enhanced the user's ability to better understand an urban model, they also noted that creating legible cities for users of all backgrounds is not a trivial task and would require knowledge of the user's perspective of the city prior to creating the clusters.

Dykes and Brunsdon [DB07] introduced a series of geographically weighted (gw) interactive graphics to explore spatial relationships between geographic processes. These techniques include standard color maps, maps of gw-means, gw-residual maps, and a localized version of the box-and-whisker plot. The techniques introduced reveal information about geographically weighted statistics at several scales concurrently.

Roman et al. [RGL04] presented an interactive system for constructing multi-perspective images from sideways-looking video captured from a moving vehicle. The input to their system is a set of video frames with known camera pose. Their system automatically computes an additional cross-slits camera between every pair of adjacent user-specified cameras leading to a smooth interpolation of viewpoint in the final multi-perspective image. Multi-perspective image of a whole city block can be created in a few minutes. The goal of this work is to simultaneously view real-world urban scenes that cannot be captured in a single photograph, rather than to visualize the simulation data of an urban space. New techniques could be explored that combine a multi-perspective approach for data visualization.

To date, simulation systems such as UrbanSim have been relatively limited in their scope of visualization, in spite of providing a sophisticated economic and behavioral simulation engine to model the location and travel choices of millions of agents in the system. A typical scenario is that manual post-processing of simulation results must be done by a model user to extract summary indicators from the results, export them from the simulation environment into a GIS system, establish relational joins of the indicators to existing GIS layers, and then manually rendering thematic or choroplethic maps to render the spatial variation in the resulting indicators (Figure 17). As used in the planning literature, an indicator is a variable that conveys information on the condition or trend of one or more attributes of the system considered. The work of Schwartzmann and Borning [SB07] developed a web-based indicator system for UrbanSim and evaluated design techniques including Value Sensitive Design, paper prototyping, and frequent user testing.

Virtual reality visualization techniques have also been explored for urban modeling and urban planning. For example, Drettakis et al. [DRRT07] presents one of such techniques applied to a small scale real-world scenario. The paper concludes that appropriate levels of realism such as spatialized 3D sound, high-detail vegetation and shadows, and crowds, enable better appreciation of overall ambience of the virtual environments, perception of space and physical objects as well as the sense of scale. Thus, using virtual environments for the visualization of large-scale urban simulations is a desirable line of future work with significant challenges to overcome.

4.3. Bridging the Gap between Simulation and Visualization

This process of simulation and then visualization has many limitations, not the least of which is the level of effort. As a result, too little visualization is actually done in practice, and this leads to diminished access to the simulation results, and reduced diagnostic capacity to determine when there are problems in the simulation. One option being recently explored is to more tightly integrate visualization efforts with the simulation process, achieve a beneficial and symbiotic relationship, and ultimately lead us to a more desired and integrated approach to urban modeling.

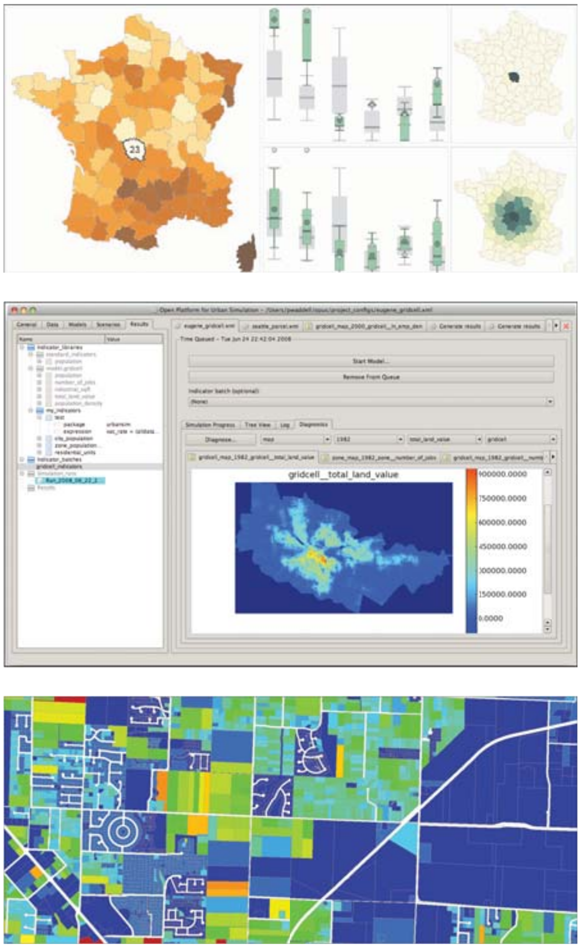

Figure 17: ***Traditional Urban Visualization.*** *(Left) Multivariate geographically-weighted boxplots are one of the data visualization tools for scale-varying exploratory analysis presented by Dykes and Brunsdon [DB07]. (Middle) A screen snapshot from the Indicator system supported by UrbanSim [SB07]. The program is a web-based interface for visualizing and browsing indicator results in the form of tables, charts, or maps. The figure shows a choroplethic visualization of the land value in the city of Eugene, Oregon. (Right) A standard choropleth map traditionally used to visualize the results of an urban simulation.*

A recent approach for larger scale problems is that of Weber et al. [WMWG09] who proposed to combine procedural modeling techniques with urban simulation to obtain three-dimensional models that change over time (Figure 18). The system includes a street expansion algorithm to place new streets, a land use simulation, and a traffic simulation as major building blocks. Although much simpler than full-featured urban simulators, the simulation is interactive and the user can make modifications during the simulation, such as controlling the major growth areas, editing road segments, editing land use, and changing parameters used to control the simulation. The goal of this work is to provide a generic framework that can be configured for different types of land use categories.

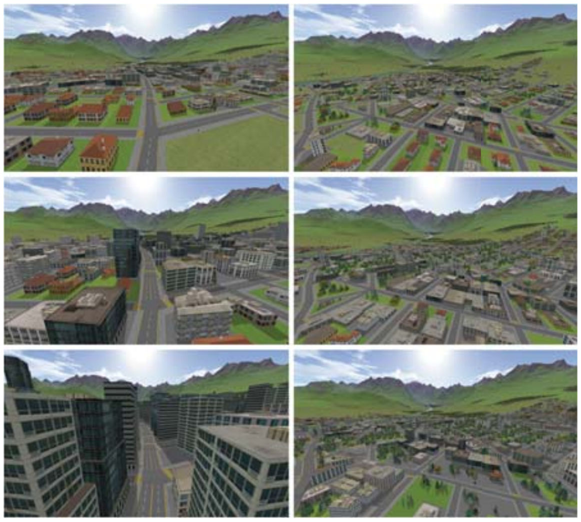

Figure 18: ***Interactive Geometric Simulation of 4D Cities*** *[WMWG09]. Two time series are shown in the columns. The left column shows the transition from low density to high density in the city center. The right column shows a transition of a city based on sustainable development with sufficient green areas.*

Another recent approach proposed by Vanegas et al. [VABW09] uses urban layouts for the visualization of urban simulation results. This work builds upon existing visualization techniques of urban simulations and extends them by automatically inferring new urban layouts for any time step of the simulation sequence, by considering both the values of the state variables of the simulation model, and the original street network, parcels, and aerial imagery of the simulated city (Figure 19). The inference algorithms gather stochastic data of the original urban layout and use the simulation state values to obtain a plausible urban layout, consisting of new parcels, streets, and imagery (e.g., vector and image data). Altogether, this approach allows for traditional visualizations as well as that of new content. It has been applied to visualize a 16,300 km^2 urban space.

Both of these works make the initial strides toward the ultimate goal of a more integrated approach of urban modeling, simulation, and visualization.

5. Conclusions, Challenges and Open Problems

Providing realistic and plausible models of dense urban spaces is a challenge that requires knowledge from several disciplines. The pursuit of the accurate modeling of urban spaces is of significant interest today to urban planners, to emergency management, and to visualization efforts. In recent years, the data collected via several forms of acquisition is available, through the Internet, to a widespread audience and has fomented significant activity and applications. Several geometric modeling methods have focused on urban spaces in order to improve their flexibility and efficiency. Simultaneously, simulation models are becoming increasingly sophisticated and better able to represent the complex processes occurring in urban spaces. These simulations serve to better understand urban spaces and to help guide 3D modeling and rendering efforts to produce more realistic and interactive imagery.

In this article, we have attempted to help guide future efforts in urban modeling to have a better understanding of the multiple aspects of this challenge. We look forward to more holistic approaches and to multi-disciplinary collaborations.

Acknowledgements

This work was funded in part by NSF awards IIS 0753116, CCF 0643822, IIS 0757623, and by the Purdue Research Foundation. We also thank Eric Bruneton, Bernard Frisher, Aaron Hertzmann, Paul Merrell, Tan Ping, the Corbis database and Procedural Inc. for their images, and the anonymous reviewers for their valuable feedback.

References

[ABVA08] Aliaga D. G., Beneš B., Vanegas C. A., Andrysco N.: Interactive reconfiguration of urban layouts. *IEEE Computer Graphics & Applications 28*, 3 (2008), 38–47.

[ACW*99] Aliaga D., Cohen J., Wilson A., Baker E., Zhang H., Erikson C., Hoff K., Hudson T., Stuerzlinger W., Bastos R., Whitton M., Brooks F., Manocha D.: MMR: an interactive massive model rendering system using geometric and image-based acceleration. In *ACM Symposium on Interactive 3D Graphics* (1999), pp. 199–206.

[AK96] Anas A., Kim I.: General equilibrium models of polycentric urban land use with endogenous congestion and job agglomeration. *Journal of Urban Economics 40*, 2 (1996), 232–256.

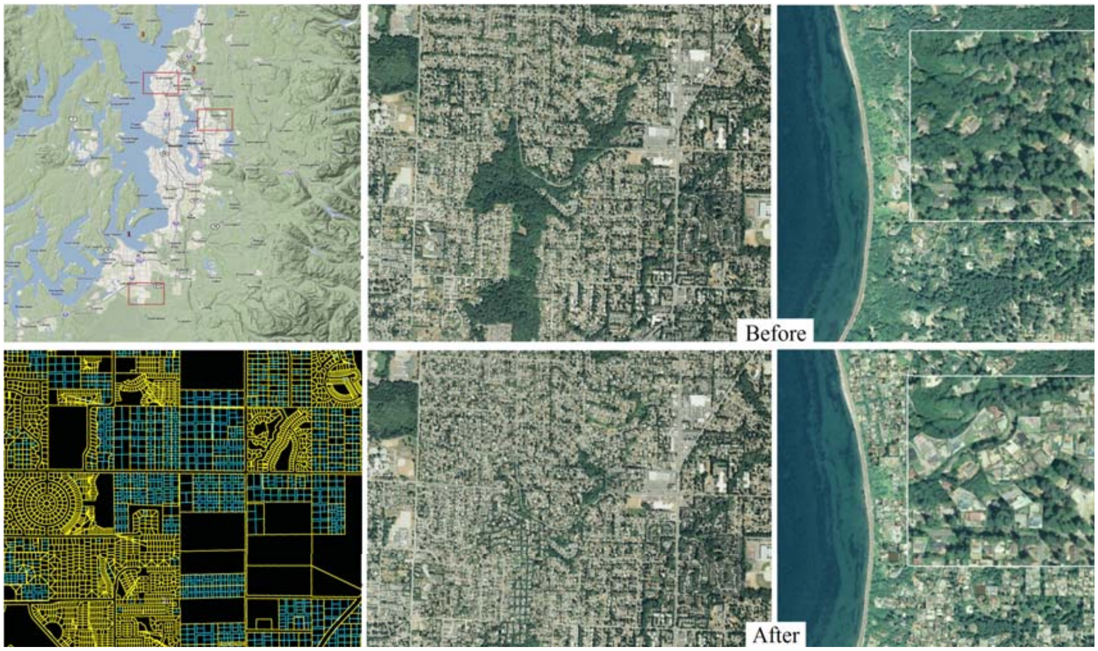

Figure 19: ***Visualization of Simulated Urban Spaces*** *[VABW09]. (Left column) An overview of the simulated region (borrowed from Google Maps) along with the parcel geometry inferred from the urban simulation data for a subset of that region. (Middle column) A close up of a part of the city where new developments are predicted by the simulation after a 30 year period, and their geometry inferred by our system. (Right column) A close up of a different part in which parcel subdivision was indicated by the simulation.*

[AkWS08] Al-kheder S., Wang J., Shan J.: Fuzzy inference guided cellular automata urban-growth modelling using multi-temporal satellite images. *International Journal of Geographic Information Science 22*, 11-12 (2008), 1271–1293.

[AL99] Aliaga D. G., Lastra A.: Automatic image placement to provide a guaranteed frame rate. In *Proceedings of ACM SIGGRAPH* (1999), pp. 307–316.

[ARB07] Aliaga D. G., Rosen P. A., Bekins D. R.: Style grammars for interactive visualization of architecture. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (2007), 786–797.

[AVB08] Aliaga D. G., Vanegas C. A., Beneš B.: Interactive example-based urban layout synthesis. *ACM Transactions on Graphics 27*, 5 (2008), 1–10.

[AYRW09] Ali S., Ye J., Razdan A., Wonka P.: Compressed facade displacement mapping. *IEEE Transactions on Visualization and Computer Graphics 15*, 2 (2009).

[Bat92] Batty M.: Urban modeling in computer-graphic and geographic information system environments. *Environment and Planning B: Planning and Design 19*, 6 (1992), 663–688.

[Bat07] Batty M.: *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals*. The MIT Press, September 2007.

[BBJ*01] Birch P., Browne S., Jennings V., Day A., Arnold D.: Rapid Procedural-Modelling of Architectural Structures. In *Virtual Reality, Archaeology and Cultural Heritage (VAST)* (2001), pp. 187–196.

[BC97] Batty M., Cole S.: Time and space : Geographic perspectives on the future. *Futures 29*, 4-5 (1997), 277 – 289.

[BN08] Bruneton E., Neyret F.: Real-time rendering and editing of vector-based terrains. In *Eurographics '08, April, 2008* (Hersonissos, Grèce, 2008), Drettakis G., Scopigno R., (Eds.), vol. 27.

[BWPP04] Bittner J., Wimmer M., Piringer H., Purgathofer W.: Coherent hierarchical culling: Hardware occlusion queries made useful. *Computer Graphics Forum (Proceedings Eurographics) 23*, 3 (2004), 615–624.

[CBZ*08] Chang R., Butkiewicz T., Ziemkiewicz C., Wartell Z., Pollard N., Ribarsky W.: Legible simplification of textured urban models. *IEEE Computer Graphics & Applications 28*, 3 (2008), 27–36.

[CDG*07] Cignoni P., Di Benedetto M., Ganovelli F., Gobbetti E., Marton F., Scopigno R.:

Ray-casted blockmaps for large urban models visualization. *Computer Graphics Forum (Proceedings of Eurographics) 26*, 3 (2007), 405–413.

[CEW*08] CHEN G., ESCH G., WONKA P., MÜLLER P., ZHANG E.: Interactive procedural street modeling. *ACM Transactions on Graphics 27*, 3 (2008), 1–10.

[Cla98] CLARKE K. C.: Loose-coupling a cellular automaton model and gis: long-term urban growth prediction for san francisco and washington/baltimore. *International Journal of Geographical Information Science* (1998), 699–714.

[CLDD09] CABRAL M., LEFEBVRE S., DACHSBACHER C., DRETTAKIS G.: Structure preserving reshape for textured architectural scenes. *Computer Graphics Forum (Proceedings of the Eurographics conference)* (2009).

[CWK*07] CHANG R., WESSEL G., KOSARA R., SAUDA E., RIBARSKY W.: Legible cities: Focus-dependent multi-resolution visualization of urban relationships. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1169–1175.

[DB07] DYKES J., BRUNSDON C.: Geographically weighted visualization: Interactive graphics for scale-varying exploratory analysis. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1161–1168.

[DDTP00] DURAND F., DRETTAKIS G., THOLLOT J., PUECH C.: Conservative visibility preprocessing using extended projections. In *Proceedings of ACM SIGGRAPH* (2000), pp. 239–248.

[DF81] DOWNING F., FLEMMING U.: The bungalows of buffalo. *Environment and Planning B 8* (1981), 269–293.

[dPPW07] DE PALMA A., PICARD N., WADDELL P.: Discrete choice models with capacity constraints: An empirical analysis of the housing market of the greater paris region. *Journal of Urban Economics 62*, 2 (2007), 204–230.

[DRRT07] DRETTAKIS G., ROUSSOU M., RECHE A., TSINGOS N.: Design and evaluation of a real-world virtual environment for architecture and urban planning. *Presence: Teleoper. Virtual Environ. 16*, 3 (2007), 318–332.

[DSWD94] DAVIS M., SIGAL R., WEYUKER E. J., DAVIS M. D.: *Computability, Complexity, and Languages : Fundamentals of Theoretical Computer Science*. Academic Press, 1994.

[Dua02] DUARTE J.: *Malagueira Grammar – Towards a Tool for Customizing Alvaro Siza's Mass Houses at Malagueira*. PhD thesis, MIT School of Architecture and Planning, 2002.

[Fin08] FINKENZELLER D.: Detailed building facades. *IEEE Computer Graphics & Applications 28*, 3 (2008), 58–66.

[Fle87] FLEMMING U.: More than the sum of its parts: the grammar of queen anne houses. *Environment and Planning B 14* (1987), 323–350.

[GBK06] GUTHE M., BALÁZS A., KLEIN R.: Near optimal hierarchical culling: Performance driven use of hardware occlusion queries. In *Eurographics Symposium on Rendering* (June 2006), pp. 207–214.

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of ACM SIGGRAPH* (1997), pp. 209–216.

[GKM93] GREENE N., KASS M., MILLER G.: Hierarchical z-buffer visibility. In *Proceedings of ACM SIGGRAPH* (1993), pp. 231–238.

[GMC*06] GOBBETTI E., MARTON F., CIGNONI P., BENEDETTO M. D., GANOVELLI F.: C-BDAM – compressed batched dynamic adaptive meshes for terrain rendering. *Computer Graphics Forum (Proceedings of Eurographics) 25*, 3 (2006), 333–342.

[GPSL03] GREUTER S., PARKER J., STEWART N., LEACH G.: Real-time procedural generation of 'pseudo infinite' cities. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia* (New York, NY, USA, 2003), ACM, pp. 87–ff.

[Hav05] HAVEMANN S.: *Generative Mesh Modeling*. PhD thesis, TU Braunschweig, 2005.

[HBW06] HAHN E., BOSE P., WHITEHEAD A.: Persistent realtime building interior generation. In *Sandbox: Proceedings of the ACM SIGGRAPH Symposium on Videogames* (2006), pp. 179–186.

[HER99] H. EHRIG G. ENGELS H. K., ROZENBERG G.: *Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages and Tools*. World Scientific Publishing Company, 1999.

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Proceedings of ACM SIGGRAPH* (2001), pp. 327–340.

[Hop96] HOPPE H.: Progressive meshes. In *Proceedings of ACM SIGGRAPH* (1996), pp. 99–108.

[Hop99] HOPPE H.: Optimization of mesh locality for transparent vertex caching. In *Proceedings of ACM SIGGRAPH* (1999), pp. 269–276.

[HYN03] HU J., YOU S., NEUMANN U.: Approaches to large-scale urban modeling. *IEEE Comput. Graph. Appl. 23*, 6 (2003), 62–69.

[JWSP05] JESCHKE S., WIMMER M., SCHUMANN H., PURGATHOFER W.: Automatic impostor placement for guaranteed frame rates and low memory requirements. In *Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games* (2005), pp. 103–110.

[KE81] KONING H., EIZENBERG J.: The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B 8* (1981), 295–323.

[KEBK05] KWATRA V., ESSA I., BOBICK A., KWATRA N.: Texture optimization for example-based synthesis. *ACM Transactions on Graphics 24*, 3 (2005), 795–802.

[KNP04] KEIM D. A., NORTH S. C., PANSE C.: Cartodraw: a fast algorithm for generating contiguous cartograms. *IEEE Transactions on Visualization and Computer Graphics 10*, 1 (2004), 95–110.

[Knu68] KNUTH D.: Semantics of context-free languages. *Mathematical Systems Theory 2*, 2 (1968), 127–145.

[LDG01] LEGAKIS J., DORSEY J., GORTLER S.: Feature-based cellular texturing for architectural models. In *Proceedings of ACM SIGGRAPH* (2001), pp. 309–316.

[LH04] LOSASSO F., HOPPE H.: Geometry clipmaps: Terrain rendering using nested regular ggrids. *ACM Transactions on Graphics 23*, 3 (2004), 769–776.

[LKR*96] LINDSTROM P., KOLLER D., RIBARSKY W., HODGES L. F., FAUST N., TURNER G. A.: Real-time continuous level of detail rendering of height fields. In *Proceedings of ACM SIGGRAPH* (1996), pp. 109–118.

[LPW*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics 25*, 3 (2006), 681–689.

[LRC*02] LUEBKE D., REDDY M., COHEN J. D., VARSHNEY A., WATSON B., HUEBNER R.: *Level of Detail for 3D Graphics*. Morgan Kaufmann, 2002.

[LSCO03] LEYVAND T., SORKINE O., COHEN-OR D.: Ray space factorization for from-region visibility. *ACM Transactions on Graphics 22*, 3 (2003), 595–604.

[LWW08] LIPP M., WONKA P., WIMMER M.: Interactive visual editing of grammars for procedural architecture. *ACM Transactions on Graphics 27*, 3 (2008), 1–10.

[Mar96] MARTINEZ F.: Mussa: A land-use model for santiago city. *Transportation Research Record 1552* (1996), 126–134.

[MBW08] MATTAUSCH O., BITTNER J., WIMMER M.: CHC++: Coherent hierarchical culling revisited. *Computer Graphics Forum (Proceedings Eurographics) 27*, 2 (2008), 221–230.

[Mcf74] MCFADDEN D.: Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics* (1974), 105–142.

[Mer07] MERRELL P.: Example-based model synthesis. In *ACM Symposium on Interactive 3D Graphics and Games* (2007), pp. 105–112.

[MH02] MÖLLER T., HAINES E.: *Real-Time Rendering, Second Edition*. A. K. Peters Limited, 2002.

[MM08] MERRELL P., MANOCHA D.: Continuous model synthesis. *ACM Transactions on Graphics 27*, 5 (2008), 1–7.

[MPB05] MARVIE J. E., PERRET J., BOUATOUCH K.: The FL-system: a functional L-system for procedural geometric modeling. *The Visual Computer 21*, 5 (2005), 329–339.

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Transactions on Graphics 25*, 3 (2006), 614–623.

[MZWG07] MÜLLER P., ZENG G., WONKA P., GOOL L. V.: Image-based procedural modeling of facades. *ACM Transactions on Graphics 24*, 3 (2007), 85.

[NS04] NIRENSTEIN S. GAIN J. B. E.: Aggressive visibility pre-processing with adaptive sampling. In *Rendering Techniques* (2004).

[PDBB00] PINNEL L. D., DOCKREY M., BRUSH A. J. B., BORNING A.: Design of visualizations for urban modelling. In *VisSym: Joint Eurographics and IEEE TCVC Symposium on Visualization* (2000).

[PL91] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants*. Springer Verlag, 1991.

[PLW*07] POTTMANN H., LIU Y., WALLNER J., BOBENKO A., WANG W.: Geometry of multi-layer freeform structures for architecture. *ACM Transactions on Graphics 26*, 3 (2007), 65.

[PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *Proceedings of ACM SIGGRAPH* (2001), pp. 301–308.

[POJ05] POLICARPO F., OLIVEIRA M. M., JO A. L. D. C.: Real-time relief mapping on arbitrary polygonal surfaces. In *ACM Symposium on Interactive 3D Graphics and Games* (2005), pp. 155–162.

[Por00] PORTUGALI J.: *Self-Organization and the City*. Berlin: Springer-Verlag, 2000.

[Pro08] PROCEDURAL: www.procedural.com, 2008.

[PSB*08] POTTMANN H., SCHIFTNER A., BO P., SCHMIEDHOFER H., WANG W., BALDASSINI N., WALLNER J.: Freeform surfaces from single curved panels. *ACM Transactions on Graphics 27*, 3 (2008), 1–10.

[Put91] PUTMAN S. H.: *Integrated Urban Models 2 - New Research and Applications of Optimization and Dynamics*. Pion, London, 1991.

[RGL04] ROMAN A., GARG G., LEVOY M.: Interactive design of multi-perspective images for visualizing urban landscapes. In *IEEE Visualization* (2004), pp. 537–544.

[SB07] SCHWARTZMAN Y., BORNING A.: The indicator browser: A web-based interface for visualizing urbansim simulation results. *Hawaii International Conference on System Sciences* (2007), 92a.

[SDDS00] SCHAUFLER G., DORSEY J., DECORET X., SILLION F. X.: Conservative volumetric visibility with occluder fusion. In *Proceedings of ACM SIGGRAPH* (2000), pp. 229–238.

[SHOW02] SMITH J., HODGINS J., OPPENHEIM I., WITKIN A.: Creating models of truss structures with optimization. In *Proceedings of ACM SIGGRAPH* (2002), pp. 295–301.

[Sip96] SIPSER M.: *Introduction to the Theory of Computation*. Course Technology, Boston, 1996.

[SLS*96] SHADE J., LISCHINSKI D., SALESIN D. H., DEROSE T., SNYDER J.: Hierarchical image caching for accelerated walkthroughs of complex environments. In *Proceedings of ACM SIGGRAPH* (1996), pp. 75–82.

[SM78] STINY G., MITCHELL W. J.: The palladian grammar. *Environment and Planning B 5* (1978), 5–18.

[SNB07] SANDER P. V., NEHAB D., BARCZAK J.: Fast triangle reordering for vertex locality and reduced overdraw. *ACM Transactions on Graphics 26*, 3 (2007).

[ŠRW07] ŠEVČÍKOVÁ H., RAFTERY A., WADDELL P.: Assessing uncertainty in urban simulations using Bayesian melding. *Transportation Research B 41* (2007), 652–669.

[SS96] SCHAUFLER G., STÜRZLINGER W.: A three dimensional image cache for virtual reality. In *Proceedings of Eurographics* (1996), pp. 227–236.

[SSGH01] SANDER P. V., SNYDER J., GORTLER S. J., HOPPE H.: Texture mapping progressive meshes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 409–416.

[Sti75] STINY G.: *Pictorial and Formal Aspects of Shape and Shape Grammars*. Birkhauser Verlag, Basel, 1975.

[Sti80] STINY G.: Introduction to shape and shape grammars. *Environment and Planning B 7* (1980), 343–361.

[TO01] TORRENS P., O'SULLIVAN D.: Cellular automata and urban simulation: where do we go from here? *Environment & Planning B: Planning and Design 28*, 2 (2001), 163–168.

[VABW09] VANEGAS C., ALIAGA D., BENES B., WADDELL P.: Visualization of simulated urban spaces: Inferring parameterized generation of streets, parcels, and aerial imagery. *IEEE Transactions on Visualization and Computer Graphics 15*, 4 (2009).

[Wad02] WADDELL P.: UrbanSim: Modeling urban development for land use, transportation, and environmental planning. *Journal of the American Planning Association 68*, 3 (2002), 297–314.

[WBN*03] WADDELL P., BORNING A., NOTH M., FREIER N., BECKE M., ULFARSSON G.: Microsimulation of urban development and location choices: Design and implementation of urbansim. *Networks and Spatial Economics 3*, 5 (2003), 43–67.

[Weg94] WEGENER M.: Operational urban models state of the art. *Journal of the American Planning Association 60* (1994), 17–29.

[Weg04] WEGENER M.: Overview of land use transport models. In *Handbook of Transport Geography and Spatial Systems*. 2004, pp. 127–146.

[WFM01] WATSON B., FRIEDMAN A., MCGAFFEY A.: Measuring and predicting visual fidelity. In *Proceedings of ACM SIGGRAPH* (2001), pp. 213–220.

[WLKT09] WEI L.-Y., LEFEBVRE S., KWATRA V., TURK G.: Eurographics STAR, State of the Art in Example-based texture synthesis, 2009.

[WMWG09] WEBER B., MÜLLER P., WONKA P., GROSS M.: Interactive geometric simulation of 4D cities. *Computer Graphics Forum (Proceedings of Eurographics) 28*, 2 (2009).

[WUFL07] WADDELL P., ULFARSSON G., FRANKLIN J., LOBB J.: Incorporating land use in metropolitan transportation planning. *Transportation Research Part A: Policy and Practice 41* (2007), 382–410.

[WWS00] WONKA P., WIMMER M., SCHMALSTIEG D.: Visibility preprocessing with occluder fusion for urban walkthroughs. In *Proceedings of the Eurographics Workshop on Rendering* (2000), pp. 71–82.

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM Transactions on Graphics 22*, 3 (2003), 669–677.

[WWZ*06] WONKA P., WIMMER M., ZHOU K., MAIERHOFER S., HESINA G., RESHETOV A.: Guided visibility sampling. *ACM Transactions on Graphics 25*, 3 (2006), 494–502.

[XFT*08] XIAO J., FANG T., TAN P., ZHAO P., OFEK E., QUAN L.: Image-based façade modeling. *ACM Transactions on Graphics 27*, 5 (2008), 1–10.

[YLPM05] YOON S.-E., LINDSTROM P., PASCUCCI V., MANOCHA D.: Cache-oblivious mesh layouts. *ACM Transactions on Graphics 24*, 3 (2005), 886–893.