

Review Article

3D Design and Modeling of Smart Cities from a Computer Graphics Perspective

Daniel G. Aliaga

Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

Correspondence should be addressed to Daniel G. Aliaga, aliaga@purdue.edu

Received 30 September 2012; Accepted 22 October 2012

Academic Editors: U. Gudukbay, G. Lavoué, and G. Patanè

Copyright © 2012 Daniel G. Aliaga. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modeling cities, and urban spaces in general, is a daring task for computer graphics, computer vision, and visualization. Understanding, describing, and modeling the geometry and behavior of cities are significant challenges that ultimately benefit urban planning and simulation, mapping and visualization, emergency response, and entertainment. In this paper, we have collected and organized research which addresses this multidisciplinary challenge. In particular, we divide research in modeling cities and urban spaces into the areas of geometrical modeling and of behavioral modeling. The first area overlaps significantly with computer graphics and computer vision—our focus is on algorithms that produce intricate geometry quickly from a compact set of specifications (i.e., procedural modeling). The second area of behavioral modeling centers on understanding the underlying socioeconomic, meteorological, and resource consumption/waste production processes occurring within an urban space. Research in urban modeling, even from a computer graphics perspective, must tie the two areas of geometric and behavioral modeling together in order to ensure that useful 3D modeling techniques are developed and are placed within their needed context. In addition, we discuss the growing trend of inverse procedural modeling and some sample urban applications.

1. Introduction

1.1. Motivation. Design and modeling of cities, or more generally urban spaces (e.g., a neighborhood, borough, town, city, or dense region), has a long history in graphics with its root in flight simulators (1960s and 1970s) and architecture (1970s). Using computer graphics and interactive modeling and rendering systems to assist in the design, visualization, and simulation of complex 3D city models is of significant interest to a wide variety of stakeholders (e.g., ranging from scientists to engineers, politicians, and concerned citizens). An urban design process can take from months to years, employing many designers from a multitude of backgrounds. Any effort to reduce the length of the design cycle or improve the model quality is a profitable investment.

Properly managing the growth of existing cities and the design of future cities is a vital issue that will only continue to increase in importance during the twenty-first century. We are now in the first century of the urban civilization. Since 2008, and for the first time in history, more than half of the

world population lives in cities. In the year 1900, approximately 14 percent of the world's population of 1.6 billion people lived in cities. Today, more than half of the world's 7 billion people live in cities. According to the United Nations State of World Population 2007, (further) urbanization is inevitable; although cities embody environmental damage, the potential benefits of urbanization far outweigh the disadvantages—the challenge is in learning how to exploit its possibilities. Harvard professor and author of “The Triumph of the City” Edward Glaeser affirms that “cities magnify humanity's strengths...and spur innovation.” TED, the non-profit foundation that organizes conferences worldwide disseminating “ideas worth spreading,” has recently awarded its annual prize to the idea of “The City 2.0,” which essentially seeks to connect people around the globe to share ideas on how to address the ultimate design challenge: urbanization.

Enabling large-scale city design and modeling would be beneficial to numerous highly sought after applications including the following.

- (i) Entertainment. Quickly generating detailed digital content for populating urban areas in video games and in movies.
- (ii) Urban Planning and Simulation. Visualizing and predicting the future effect of population changes, economic/employment changes, and of adopted urban policies within cities and countries; showing the potential effect of socio-, economic-, and climatological aspects on urban development; providing road planners, solar energy planners, and urban energy/resource management with visualization tools; and allowing architects to see the results of using common buildings blocks.
- (iii) Mapping and Visualization. Providing maps and navigation services; using incomplete information to generate plausible models of urban locations (e.g., partial data obtained from aerial views of enemy territory); correlating urban and architectural styles with changes over time.
- (iv) 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 communication networks, resources, policing, and large-scale military deployments.

1.2. Challenges. Modeling cities, and urban spaces in general, is a daring task for computer graphics, computer vision, and visualization. Understanding, describing, and modeling the geometry (e.g., creating 2D/3D geometric models) and behavior (e.g., simulating urban development over time) of cities are of significant challenge that ultimately benefits the aforementioned applications.

From a geometric standpoint, cities and urban spaces are an intricate collection of manmade structures arranged into interconnected buildings, parcels, blocks, and streets distributed over a terrain. Dense urban environments are particularly difficult to model geometrically because they are very large and widespread, spanning from a few to hundreds of square kilometers. Traditionally, 3D modeling of cities 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 geometric modeling.

From a semantic and behavioral standpoint, cities are very large and complex ecosystems of socioeconomic entities which provide concentrated living, working, education, and entertainment options to its inhabitants. Designing resilient, sustainable, and livable cities is far from being a simple task. Using the classification of Weaver's [12] historical address to the Rockefeller foundation, city (and urban) modeling is an instance of a large problem of organized complexity: the observed phenomenon is not ad hoc and random, but rather organized, interrelated in a complicated way and exhibiting emergent behaviors. Further, city design is an example of a natural and built engineering process, whereby its modeling

and simulation is needed to be done over disparate scales and many aspects must be analyzed, optimized, controlled, and integrated.

In the past 10 years, there has been a huge explosion in methodologies for designing, modeling, and reconstructing the geometry of urban spaces due, greatly, to the rapid growth and availability of geographical information systems and georegistered aerial imagery and satellite data (e.g., remote sensing). For example, very recently the Internet (e.g., Google Maps, MapQuest, LiveEarth, Google Earth, etc.) has provided public and widespread access to a very large number of top-down-view aerial images and oblique-view aerial images of the entire country and world, in addition to geographical information system (GIS) data (e.g., Open Street Map). While previously such information, at a smaller scale, was available, it was not provided to the public. Today, the resolution and coverage of such data is rapidly and continually improving. For example, very soon the raw data for the United States will easily reach and exceed petabyte size image databases (e.g., the United States' 10 M km² at 6-inch per pixel resolution already amounts to petabyte size databases). Furthermore, a huge number of ground-level imagery also exist that is being captured by Google, Microsoft, Navtec and other companies and by private citizens (e.g., tourists). This last source of data has already fomented research to produce animations and 3D reconstructions of distant urban locations using large photo collections extracted from the web (e.g., 100,000+ pictures can be found of a single popular location) [13–15].

1.3. State of the Art. Research in modeling cities and urban spaces can be divided into the areas of geometrical modeling and of behavioral modeling. The first area overlaps significantly with computer graphics and computer vision (e.g., [16]). The focus is either algorithms that produce intricate geometry quickly from a compact set of specifications or algorithms that reconstruct geometry from aerial and terrestrial images, LIDAR, and other sensor modalities. The goal of behavioral modeling is understanding the underlying socioeconomic, meteorological, and resource consumption/waste production processes occurring within an urban space [17]. Research in urban modeling, even from a computer graphics perspective, must tie together the two areas of geometric and behavioral modeling in order to ensure that useful 3D modeling techniques are developed and are placed within their needed context. Their utility is to assist decision making of urban policies in current and future urban areas (e.g., [18, 19]).

2. Urban Geometry

A 3D urban space model is composed of a terrain, over which lies a road network containing city blocks divided into parcels (or lots), and each parcel contains one or more building structures. Further, models can include vegetation, water bodies, pedestrians and vehicles, and many more objects as well. With regards to the objectives of geometric modeling, we can divide them into two main groups:

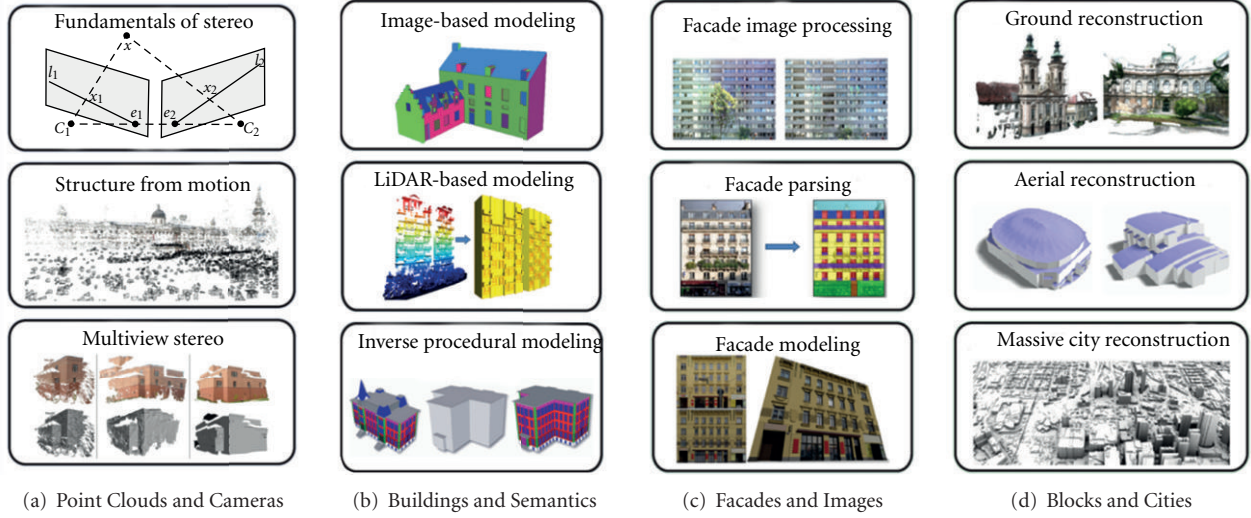


FIGURE 1: Urban reconstruction. A taxonomy of urban reconstruction work (image from [1]).

(i) the acquisition of sufficient information of an underlying large real-world city environment (e.g., [1] and Figure 1) and (ii) the support of compact and succinct editing of an underlying 3D model. While some applications seek exact physical replicas of the city (e.g., navigation), others can tolerate geometric approximations but need to be able to efficiently make speculative and extensive changes in order to run desired simulations under new conditions (e.g., urban planning and simulation). In this section, we first provide a summary of procedural modeling, a strategy that has become very popular for modeling urban spaces, as well as in other domains (e.g., [20]). Then, we emphasize a set of methods for adding building geometry. Afterwards, we focus on methods which produce urban layouts, consisting mostly of road networks and parcel geometries (e.g., although the geometric content may be spread over a digital elevation map, the content is mostly 2D geometric structures). Finally, we focus on the creation of city-scale geometric models.

2.1. Procedural Modeling. Urban structures, and other man-made objects, often exhibit a significant amount of symmetry and repetition on a global scale and individuality at a local scale. This structured redundancy can be captured using procedural modeling. Procedural methods have the advantage of exhibiting a high degree of detail amplification, for example, from only a small number of parameters significant details can be synthetically generated. Even though a small change in the parameter values can cause massive changes in the resulting model, promising results have been shown in several restricted domains.

Procedural modeling can use one of several production systems such as semi-Thue processes [21], Chomsky grammars [22], graph grammars [23], shape grammars [24], attributed grammars [25], L-systems [26], or set grammars [27]. Relevant to 3D modeling, one important aspect is the expressiveness of the chosen production system. On one side of the spectrum, a turing or C++ machine would provide

very high levels of expressiveness but at the cost of compactness. On the other side, using regular expressions yields a very compact notation but has limited expressiveness. Another very relevant aspect is how the intuitive or “easy to use” is the chosen methodology? For instance, it is intuitive to think of designing the facade of a building by recursively subdividing it—this leads to a split grammar, for example. However, it is less intuitive to think of modeling a tree by a set of splitting operations; instead, a system that supports “extending/growing” segments and “branching” is more natural—this brings us to L-systems. Thus, the procedural modeling methodology must be well chosen otherwise the intuitiveness, and ease of use, of the modeling tool will quickly be lost.

Several fundamental methodologies for procedural plant and architecture modeling have been proposed. For example, grammar-based systems have been used to model trees and other plants. Prusinkiewicz and Lindenmayer [26] proposed L-systems for procedurally modeling plants. Later work continued improving the expressivity of these methods in a variety of directions (e.g., [2, 28, 29] and Figure 2). Such procedural modeling methods have appeared in computer graphics since the 1980s (e.g., [30]).

In architecture, several approaches have been presented that exploit and represent repetition using patterns and grammars. Stiny and Mitchell [24] proposed the notion of using a shape grammar to construct and analyze architectural spaces. Alexander et al. [31] defined desirable patterns for configuring entire urban spaces. Other works have focused on implementing procedural methods for selected and well-known architectural styles (e.g., Le Corbusier-style buildings [32]). Marshall [33] summarizes a variety of styles and street patterns in a form suitable for procedural modeling.

Some works have combined plant modeling with architectural structures. For example, Wissen-Hayek et al. [34] encode design guidelines for urban green spaces into shape grammar rules and produce plausible 3D urban models.



FIGURE 2: Procedural modeling. An example of procedural ecosystem modeling (image from [2]).

Beneš et al. [35] provide an interactive, intuitive, and efficient modeling process for virtual plants and plant ecosystems that responds to surrounding shape and geometry cues. Vegetation can be made to interactively grow, in a plausible way, around existing architectural and geometrical structures, thus populating a model with vegetation. Beneš et al. [36] extended the scale to entire cities and supports managed and unmanaged growth of a simple ecosystem over an entire city in order to explore city designs and vegetation management rules that can yield a desired outcome. Recently, Pirk et al. [37] describe a dynamic tree modeling and representation technique that allows complex tree models to interact with their environment using biologically motivated transformations.

In the following, we describe several forward procedural modeling approaches that stem from the study of plants and architecture and has grown into building and city modeling. The term forward procedural modeling refers to a modeling process whereby the user specifies procedural rules and their parameter values. This is in contrast to inverse procedural modeling, whereby the output is given but the rules and/or parameters are unknown techniques along this line of research will be discussed later in this paper.

2.2. Buildings and Facades. Several works have focused their attention on the procedural modeling of structures varying from individual buildings to entire cities. Parish and Müller's [16] seminal work on city modeling describes an entire system that models streets using an L-system, divides the space in between streets into parcels, and populates the parcels with simple building models, altogether yielding a pioneering approach to rapid city model creation.

We can loosely divide such previous-related efforts into urban reconstruction efforts and urban modeling efforts. The first group attempts to accurately reconstruct existing urban spaces from ground-based and aerial-based photographs and/or from sensor data, such as LIDAR. Musialski et al. [1] provide a good overview of urban reconstruction efforts that overlap with the disciplines of computer vision, photogrammetry, and computer graphics. While these approaches have made significant progress in terms of accuracy and completeness, the research does not focus on

generating novel models. The second group has devoted significant effort to the geometric and/or procedural modeling of urban structures such as a buildings, facades, roads and layouts, and entire cities and is described in the rest of this section.

Methods for modeling buildings have exploited the repetitive pattern many buildings exhibit. Wonka et al. [27] enable designing new buildings using split grammars and introduces an attribute matching system and a separate control grammar. Collectively, these tools provide the flexibility to model a large variety of building styles. Later, Mueller et al. [3] proposed a novel shape grammar-based procedural method, CGA to construct detailed building models (Figure 3). Context sensitive shape rules allow the user to specify interactions between different parts of a hierarchical description of a model. It was demonstrated in a large reconstruction effort of Pompeii. Recently, Kelly and Wonka [38] describe a method to interactively sketch buildings by procedural extrusions of building footprints. Their approach can model difficult structures such as curved roofs, overhanging roofs, roof constructions with vertical walls, dormer windows, bay windows, buttresses, chimneys, and intricate columns. Merrell and Manocha [39, 40] propose an alternative scheme based on example-based model synthesis. The algorithm exploits the connectivity between the adjacent boundary features of the input examples and computes an output model that has similar connected features while still resembling the input.

Methods for facade generation have addressed such from a variety of perspectives. The work of Müller et al. [41] generates detailed geometric facades by proposing a set of split operations for an aerial view of an actual building. The method uses a form of synchronization to exploit the repetition of facades features, such as windows, to increase robustness and uses template matching to obtain the final detailed geometry. Subsequently, Xiao et al. [42] use images captured along streets, structure from motion, and top-down recursive decomposition method, followed by a bottom-up merging which detects architectural bilateral symmetry and repetitive patterns to recover a model of facades. Later, the work was extended to an image-based street side view model, including the labeling of specific object classes, such as building, sky, ground, vegetation, and car. Several streets in several cities were automatically modeled. In other work, instead of focusing on precisely modeling existing facades, Haegler et al. [43] centers on a new lightweight grammar representation that serves to compactly encode facade structures and allows fast computer graphics rendering.

Other work has concentrated on different aspects of building design. For example, Birch et al. [44] describe how to interactively edit a procedural building model and Lipp et al. [45] focus on a visual form of interactive grammar editing, thus avoiding having to explicitly write grammar rules using a text editor. Legakis et al. [46] describe an approach to procedurally generate building surface details, such as brick and stone cellular textures. The design of self-supporting building structures [47], several free-form building designs [48–51], and circular arc structures [52] has received



FIGURE 3: Procedural building modeling. An example of procedurally generating and altering 3D building models (image from [3]).



FIGURE 4: Urban layout modeling. (a) On the left side, we see a fragment of a real city containing street and aerial image data. The user can then interactively “grow” the street network in a similar style. (b) The generated new streets can be populated with a best fit of fragments from the aerial imagery in the source fragment (image from [4, 5]).

particular attention requiring a custom set of methodologies and concerns (e.g., [48, 49]).

It is worth noting that several of the above works have been transferred to commercial products, such as into CityEngine, a commercially available software platform for building-scale and city-scale procedural modeling (<http://www.esri.com/software/cityengine>).

2.3. Roads and Parcels. The modeling of roads and parcel geometries has received particular attention and a diverse set of approaches. Collectively, this subset of an urban model is sometimes referred to as “urban layouts.” Often, the modeling includes the treatment of aerial imagery and builds off the concept of texture or image synthesis (e.g., [53]) but in addition to the synthesis of pixel data, associated vector data is also synthesized.

A well-known treatment of texture synthesis was defined by Efros and Leung [54]. In this seminal paper, the texture synthesis process grows a new image outward from an initial seed. A Markov random field model is assumed, and the conditional distribution of the pixels is estimated by querying the sample image and finding all similar neighborhoods. In later work, Efros and Freeman [55] present an image-based method for generating a novel visual appearance by stitching together small patches of existing images. Kwatra et al. [56] extend the concept to images and video using graph cuts. Hertzmann et al. [57] describe a similar framework for processing images by example and applied such to aerial urban images. However, they do not maintain or produce any underlying street or parcel information. Thus, at a

distance, the texture synthesis methodology produces apparently plausible aerial views. However, upon close inspection, it becomes clear that the lack of explicit treatment of roads and parcels produces meaningless results.

Several works have added concepts to the image-synthesis process to make it suitable for urban aerial views. For example, Aliaga et al. [4] describe an editor for the interactive reconfiguration of city layouts, which provides tools to expand, scale, replace, and move parcels and blocks, while efficiently maintaining their connectivity and the distribution of urban zoning. In Aliaga et al. [5], an interactive system synthesizes urban layouts by example (Figure 4). This method 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. The approach provides several high-level operations to easily and 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 concerned about low-level structural details—collectively the tools provide a powerful way to generate new synthetic urban content. Groenewegen et al. [58] procedurally create layouts of structurally plausible cities from high-level, intuitive user input such as city size, location, and historic background. The resulting layouts consist of different kinds of city districts which are arranged using constraints derived from established models of urban land use. Lipp et al. [59] present an interactive modeling solution for city layouts that combines the power of procedural modeling with the flexibility of manual modeling. They describe

transformation and merging operators for both topology-preserving and topology-changing transformations based on graph cuts. Together with a layering system, it allows intuitive manipulation using operations such as drag and drop, translation, and rotation. Krecklau et al. [60] propose a novel approach for the temporal interpolation of city maps. Given a sparse set of historical city maps, the output is a fast forward animation of the city map development where roads and buildings are constructed and destroyed over time in order to match the sparse historical facts and to look plausible where no precise facts are available. Then, based on the city map animation they can create a procedural city model in order to render a 3D animation of the city development over decades. The method does, however, require significant manual input to specify when/where buildings and roads where constructed or destroyed.

The modeling of road networks has also received special attention in computer graphics. Parish and Müller [16] are amongst the first to center on road network generation as key to producing a 3D urban model. Their solution is based on using a L-system to grow a road network, similar to growing the branches of a plant. Starting from a seed segment, additional road segments are grown. A challenging aspect is of course controlling the growth process so as to produce a desired final road network. Chen et al. [61] provide an alternative methodology that supports both local and global control. They exploit the visual similarity between flow/tensor fields and road networks to interactively model large road networks. A user can create a street network from scratch or modify an existing street network. The framework is intuitive because it uses tensor fields to guide the generation of a street network with both global and local modeling operations such as brush strokes, smoothing, constraints, noise, and rotation fields. The aforementioned work of Aliaga et al. [5], while performing an example-based synthesis of an urban layout, also uses a stochastic and by-example methodology to reproduce existing, or generate new, road networks. The new roads can be generated in a style similar to an existing network or styles can be “interpolated” to yield new road network style and distributions. Galin et al. [62] propose an automatic method for generating roads based on a weighted anisotropic shortest path algorithm. Given a scene, they generate a road trajectory that minimizes a weighted cost function of terrain slope and natural obstacles, such as rivers, lakes, mountains, and forests. The road is generated by excavating the terrain along the path and instantiating generic parameterized models. In subsequent work, Galin et al. [63] describe a procedural method for generating hierarchical road networks connecting cities, towns, and villages over large terrains. This approach relies on a graph generation algorithm combined with a path merging algorithm that creates junctions between the different types of roads. High-level user input manipulates the density and the pattern of the network.

The subdivision of blocks and generation of parcels has been addressed in several previous works. Parish and Müller [16] include simple treatment of block subdivision which has later been improved upon. The desired goals for an automatic block subdivision method include providing egress

(i.e., ensuring a parcel has street access), yielding only plausible parcel shapes, and supporting a variety of subdivision styles. Within urban design and planning research, the focus of parcel generation is often satisfying the interests of real-estate investors and complying with zoning and building law regulations [64]. Blocks are frequently partitioned by hand and according to desired patterns (e.g., [65, 66]). With an interactive computer graphics editing framework, changing an urban layout can cause unexpected changes in a block’s subdivision, leading to difficult shape control and editing—even more so for large models. In the recent work of Vanegas et al. [67], they focus on providing an umbrella method for subdividing blocks into parcels and supporting the multiple aforementioned goals. The approach performs a partitioning of the interior of city blocks using user-specified subdivision attributes and style parameters. Moreover, the method is both robust and persistent in the sense of being able to map individual parcels from before an edit operation to after an edit operation—a critical task for interactive editing of city models.

2.4. City-Scale Geometry. Designing entire cities, or large portions thereof, as a 3D model is of considerable interest to our motivational applications. Modeling at such scale is often done by hand and/or by direct extrapolation from a given set of GIS data (such as in Parish and Müller [16]). Recently, however several works have exploited notions from urban simulation and GIS (to be described in more detail in Section 3) to automatically, or semiautomatically, construct large-scale detailed 3D city models. In this section, we review several of these methodologies.

Honda et al. [68] propose an approach to generating virtual cities that vary dynamically over time. They focus on the layout of buildings, roads, and traffic lights. The approach generates virtual cities by deleting and locating buildings from two points of view: building layouts using environment vectors, which can represent features, themselves to generate cityscapes, and time-varying using changes of vectors to vary cityscapes. They demonstrate how their method can represent features and generate various cityscapes that change over time.

Kelly and McCabe [69] present Citygen which aims to automate the procedural generation of the urban geometry typical of a modern city. Procedural modeling generates road networks over a supplied terrain. Building footprints and buildings are automatically extracted (and can be changed by hand). The user is given an interface to directly manipulate geometric elements such as road intersection nodes and to directly control and specify many aspects of the procedural generation.

To reduce the amount of manual input that is required, and to ensure plausible city geometries are produced, geometrical city modeling is tightly integrated with urban simulation. As opposed to urban visualization which takes the output of a simulation and visualizes it, this methodology is tightly integrated and iterates between simulation computations and geometry generation. Weber et al. [70] present a simulation system that does not rely on a priori



FIGURE 5: City modeling. Using data from a real city (c), a behavioral-geometrical simulator can generate a plausible 3D city model that resembles the actual city (f). Closeups of a dense urban region (a) and a more forestry region (b) can be compared to their quantitatively similar examples (d) and (e). For (d–f), a complete 3D model is actually produced, but seen from an aerial viewpoint in this example (image from [6]).

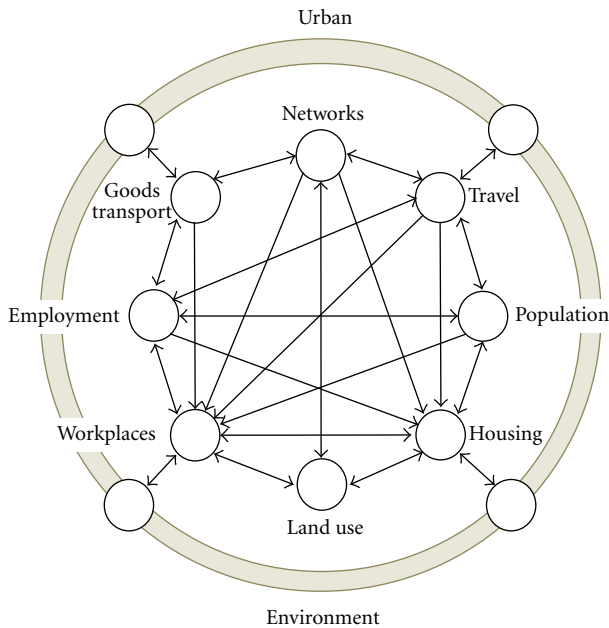


FIGURE 6: Urban simulation. A prototypical graph of the dependencies between different aspects of an urban simulation (based on image in [7], courtesy Carlos Vanegas).

land-use simulation on a regular grid, but instead builds a complete and inherently geometric simulation that includes exact parcel boundaries, streets of arbitrary orientation, street widths, 3D street geometry, building footprints, and 3D building envelopes. The system includes a fast urban simulation and supports user interaction at interactive speed, producing new geometries at the rate of about 1 second per time step. Their interactive simulator focuses on urban growth though without supporting feedback (e.g., a change of building heights does not modify the distribution of population and jobs).

Vanegas et al. [6] further incorporate an urban simulation yielding continual feedback from simulation to

geometry and vice versa (Figure 5). Their system builds off of the well-known UrbanSim simulation platform [19, 71] to create city models. In Vanegas et al. [72], the output of a previous urban simulation is used to generate a plausible 2D urban layout. Their system creates 3D urban models from scratch and simulates the interdependency of several aspects of a city, thus enabling fast design of very realistic city models, shown for regions spanning over 200 square kilometers. The proposed design process uses an iterative dynamical system for reaching equilibrium: a state where the demands of behavioral modeling match those of geometrical modeling. 3D models are generated in a few seconds and conform to both plausible urban behavior and plausible urban geometry. The framework includes an interactive agent-based behavioral modeling system as well as adaptive geometry generation algorithms.

3. Urban Simulation

Urban simulation has become very relevant to 3D computer graphics modeling as a way to generate large, plausible city models. The relationship between urban simulation and urban 3D modeling is bidirectional in the sense that 3D modeling is being used to improve the precision and abilities of urban simulation and urban simulation is being used to produce detailed 3D city models. An urban simulation, in this context, refers to the use of behavioral or process modeling of the spatial patterns of urban economic agents and objects such as jobs, population, housing, and land use [7] (Figure 6). The simulation captures the interaction among household and business agents that are located 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. Urban simulation models can be loosely divided into the following three dominant paradigms.

3.1. Cellular Automata. Early models attempting to represent emergent dynamics adopted cellular automata (CA) as the

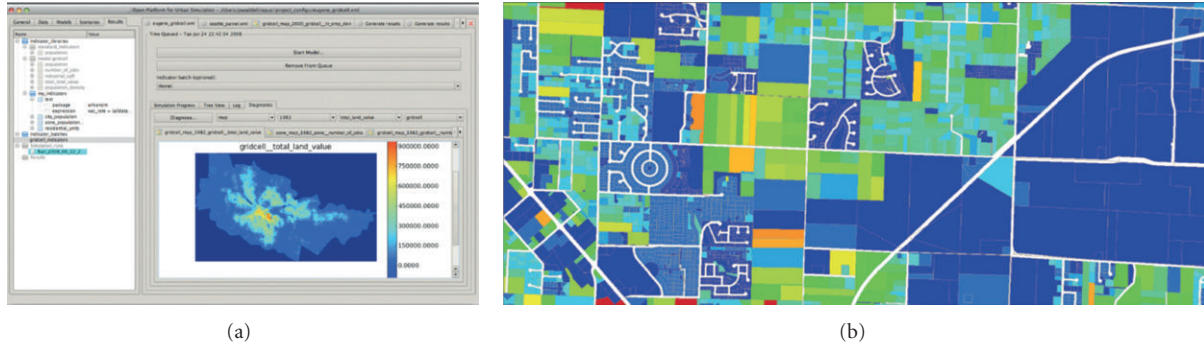


FIGURE 7: Urban visualization. An example of an urban visualization using a web-based indicator system and choroplethic maps (image from [8]).

modeling framework [73]. The essential principles of CA include the following four concepts:

- (i) cells—the object space (e.g., city) is divided into a discrete set of cells often arranged in a grid; with each cell, a variety of data fields or variable values (e.g., job counts, population, housing details, etc.) can be stored,
- (ii) state—each cell can take on exactly one state selected from a predetermined set of possible states (e.g., land, water, dense urban, suburban, forest, etc.),
- (iii) neighborhood—given a set of cells, we can easily establish a spatially compact set of adjacent and/or proximate cells, and
- (iv) transition rules—for a given CA, we can define a set of transition rules that given a current state of a cell and its neighbors, we can alter the state of the cell. Typically, the rules are uniformly tested for applicability to all cells.

One of the most widely known CA methods is the Urban Growth Model [74]. A CA-based method, using approximately 500×700 cells, was calibrated using historical digital maps of an urban area. Then, it was applied to two very different urban areas (i.e., San Francisco and Washington/Baltimore, both in the United States) and produced reasonable predictions as compared to other methods. In other work, it has been applied to long-term changes in land cover patterns classified from remote sensing data [18]. Unfortunately, the CA modeling approach only simulates the conversion of land use based on the characteristics of cells and their immediate spatial context. Hence there is no action generated at a distance. This limitation of a CA method prohibits modeling the effect of changes to the built environment or its occupants based on more global entities, such as travel networks and large-scale economic dependencies.

3.2. Agent-Based Models. Agent-based models (ABM) have extended the CA framework to include mobile and interacting agents in a spatially large urban context. Agent-based models mainly differ from cellular automata in that the

used agents are objects without a fixed location. Agents can interact with each other as well as the environment in a very flexible manner and are usually regarded as acting autonomously. Succinctly, an ABM has the following characteristics:

- (i) agents are explicitly designed to represent a particular mobile object (e.g., a person); there may be more than one agent type in a single simulation thus the agents are implicitly distributed throughout the environment, and
- (ii) agents can sense and act within their environment in one of several ways; their behavior can be reactive (e.g., they behave based solely on their surroundings) or deliberative (e.g., they have their own agenda or goal which drives their behavior); clearly, the design of the agents sensing and acting is critical to a simulation.

Altogether, agents exhibit a form of autonomous behavior and thus lend themselves to a variety of simulated behaviors including emergent patterns. Within urban simulation, many works have 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 [75]. However, relatively little attention has been paid to issues of validating models using observed data or trends, and, as with CA models, traditional ABM urban simulations have behaviors influenced only by a localized context.

3.3. Urban-Economic Discrete-Choice Models. An 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 integrates agent-based models with the pioneering work of McFadden on Random Utility Theory [76] and the development of discrete choice models, for which he recently won the year 2000 Nobel Prize in Economics. In classical continuous modeling, calculus methods are used to determine an optimum configuration of a model (e.g., regression analysis). In contrast, discrete choice models use statistical procedures to

describe choices (e.g., as made by people) among a finite set of alternatives and thus is more amenable to simulation urban environments. Research using this approach diverges on the

- (i) level of aggregation—aggregate (discrete choice) models represent agents by grouping them based on types and by grouping large zones based on locations; microsimulation models represent individual agents, such as households and jobs, and objects, such as buildings and parcels, and
- (ii) time representation—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 [77] and equilibrium discrete choice models [78, 79].

3.4. UrbanSim. An example of a recent discrete-choice approach is UrbanSim [19, 80, 81], which simulates the choices of individual households, businesses, and parcel landowners and developers interacting in urban real-estate markets. 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 [82], an explicit representation of real-estate markets, and statistical methods to estimate model parameters and to calibrate uncertainty in the model system [83].

The core of UrbanSim consists of a series of models that run in a specified order once per a simulated year, using individual agents and entities, as a microsimulation. A simulation is usually performed for a period of several years—up to 30 years for long-term project evaluation. Data for the model systems are derived from large datasets obtained from a variety of sources, including local tax assessor files of parcels and buildings, employment data from unemployment insurance records or other administrative sources, census data, household surveys, spatial data on streets, terrain and other environmental features, and land use and transportation plans. Parameters of the price models are obtained by using the hedonic regression framework [84, 85], estimated using observed real estate data and estimated by Ordinary Least Squares, or alternatively Maximum Likelihood Estimation (MLE) [86]. Parameters of the location choice models are obtained by using observed data and MLE of Multinomial Logit Models based on a Random Utility Maximization framework [76, 82, 87, 88]. Most of the models are stochastic and involve Monte Carlo sampling of choice outcomes conditional on a probability generated from a Multinomial Logit Model.

The robustness of the UrbanSim system to date has been assessed in several ways. One measure of success is its increasingly widespread adoption by public agencies responsible for regional planning of multibillion dollar transportation plans for which robust models are essential. UrbanSim has been the

subject of a favorable peer review process to assist in resolving a lawsuit over the environmental impacts of a highway project in Salt Lake City [89]. The accuracy of the model has been validated by simulating relatively long periods for which observed data were available, and the model system was found capable of reproducing key spatial dynamics in households and jobs. Finally, methods compatible with UrbanSim have been developed to calibrate the uncertainty in the model system in order to be able to make principled inferences about the uncertainty embedded in the model predictions [83].

3.5. Urban Visualization. Visualization engines have been connected to urban simulation systems, though they do not typically generate the 3D geometry of urban spaces resulting from complex urban simulations. Batty [90] introduced various approaches that relate urban modeling, GIS, and computer graphics. The same author later described the potential (positive) impact of virtual reality and 3D visualization to GIS and demonstrated this on a variety of complex examples [91]. Visualization has also been used for the purposes of education, exploration, explanation, and engagement [92]. 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., [93–96]). Alternatively, Chang et al. [97] use the multiple dimensions in their information visualization of urban relationships to produce a 3D visualization. Dykes and Brunsdon [98] introduced “geowigs,” a series of geographically weighted interactive graphics, to provide large-scale geographical environment visualization. Vanegas et al. [72] describe a visualization system that takes the output of an urban simulation, and the associated aerial imagery and street vector data of the targeted region, and produces a plausible modification to the aerial imagery and street vector data based on the output of the simulation. Thus, a top-down (e.g., aerial) view of a city reflecting the changes in the distribution and density of population and/or job counts can be visually seen.

In general, these and other approaches make use of visualization techniques including choroplethic maps (e.g., [8] and Figure 7) generated by exporting simulation results, summarized by a zonal geography, to a 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. However, to our knowledge, generating the geometry automatically from the simulation data is not heavily explored in the urban simulation community (instead, we refer the reader back to Section 2.4).

4. Inverse Procedural Modeling

A more recent, and alternative, form of procedural modeling reverses the computational direction to yield an “inverse

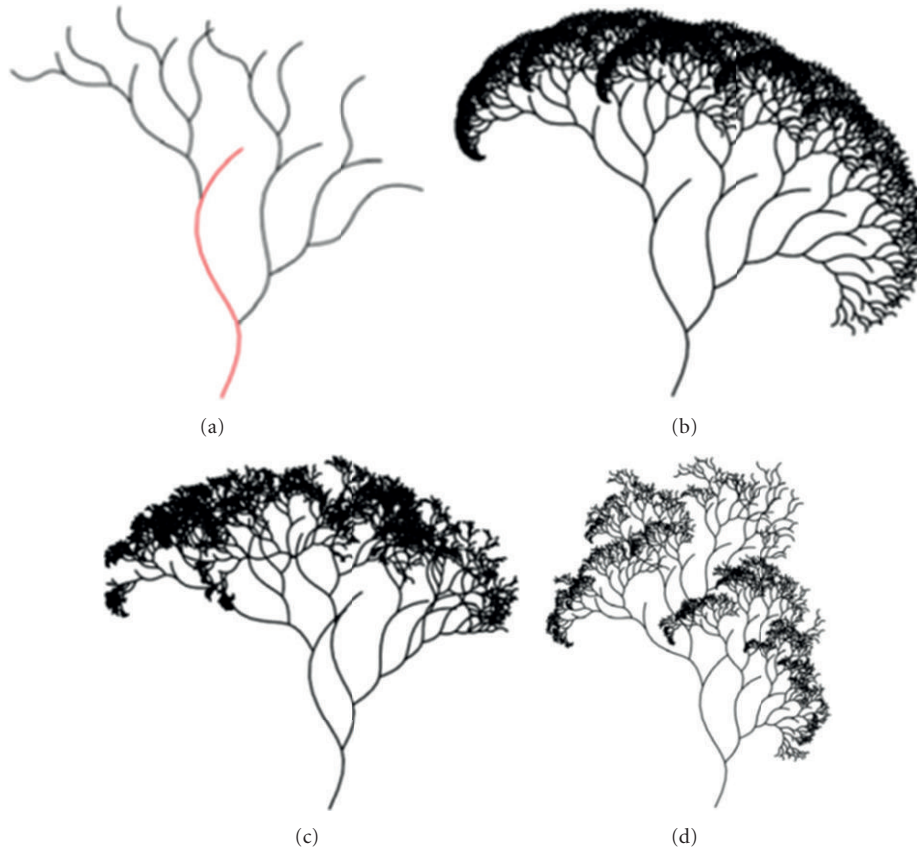


FIGURE 8: Inverse 2D plant modeling. An example of inverse procedural modeling of rules and parameter values for arbitrary 2D vector content. (a) Input vector data (e.g., curved segments). (b–d) Various recreations using the computed rules and altered parameter values (image from [9]).

modeling” framework. While procedural modeling has its clear advantages, it may often be the case that (i) the procedural model is known but the parameter values that yield a desired outcome are not, or (ii) neither the procedural model nor the parameter values is known. The former case is arguably simpler and occurs when the underlying structure is well understood and can be modeled but the parameter values for a particular instance are not available. The latter occurs when a geometric model can be observed (by pictures or by laser scanning) but the underlying procedural model is not known—having knowledge of the procedural model would then enable the same powerful and succinct editing ability of forward procedural modeling.

The notion of inverse modeling has been exploited before. Many classical inverse methods are based on regularization theory (e.g., [99]). These methods attempt to “regularize” ill-posed inverse problems, for example, the inverse problem can be defined as trying to find the underlying weights relating input parameters to target indicators. While standard regularization can improve the ability to compute a plausible solution to a generic inverse problem (e.g., Tikhonov-based regularization/regression), the typical emphasis is noise removal and/or an assumed unknown formation process of the observed indicators—neither is the case necessarily in urban modeling work.

4.1. Inverse Procedure and Parameter Estimation. The general task of discovering both the procedural rule set and the parameter values is a significant and exciting challenge. While assuming no priors at all makes the task exceedingly difficult, several works have focused on exploiting a small set of domain-specific fundamental assumptions. Stava et al. [9] proposed inverse procedural modeling of rules and parameter values for arbitrary 2D vector content (Figure 8). They assume terminal symbols are known and generate context-free rules for linear structures using L-systems. The L-system rules efficiently code the regular structures and the parameters represent the properties of the structure transformations. The input terminal symbols’ position and orientation are pairwise compared and the transformations are stored as points in multiple 4D transformation spaces. By careful analysis of the clusters in these transformation spaces, they detect sequences of elements and code them as L-system rules. This process is repeated iteratively ultimately yielding a hierarchy of grouped elements. The extracted procedural model can then be used to alter, symmetrize (e.g., [100]) or substitute parts of the original input.

Bokeloh et al. [101] address the problem of inverse procedural modeling: given a 3D model example, they seek to find rules that describe objects similar to the example. They find explicit shape modification rules that guarantee strict local



FIGURE 9: Inverse building modeling. An example of inverse building modeling of known rules but unknown parameter values. (a) One of several input photographs. (b) Manually labeled building model from which rule application and parameter values are automatically inferred. (c) Regeneration of the same initial building model. (d) Generation of a different building in the same style. (e) Nonphotorealistic rendering possible by altering only the terminal symbols of the inferred procedural model (image from [10]).

similarity by looking at partial symmetries in the object. The object is cut into pieces along curves within symmetric areas. Then, they systematically collect such cut operations and analyze their dependency to build a shape grammar. They discuss how to extract general rewriting systems, context free hierarchical rules and grid-based rules. All of this information is derived automatically. The extracted rules are then used to implement tools for semiautomatic shape modeling demonstrated on a number of different example data sets.

Park et al. [102] present a new approach to motion rearrangement that preserves the syntactic structures of an input motion by automatically learning a context-free grammar from the motion data. They reduce an input motion into a string of terminal symbols by segmenting the motion into a series of subsequences, and then associating a group of similar subsequences with the same symbol. Next, they search for an optimal segmentation such that a large number of subsequences can be clustered into groups with little error. Once the input motion is encoded as a string, a grammar induction algorithm is employed to build up a context-free grammar. This grammar can reconstruct the original string and produce novel, though similar, new strings (or motions).

Talton et al. [103] describe an algorithmic method for learning design patterns directly from data using techniques from natural language processing and structured concept learning. Given a set of labeled, hierarchical designs as input, they induce a probabilistic formal grammar over exemplars. Once learned, this grammar encodes a set of generative rules for the class of designs, which can be sampled to synthesize

novel artifacts. They demonstrate the method on geometric models and Web pages and discuss how the learned patterns can drive new interaction mechanisms for content creators.

Finally, from a very different urban perspective, Beirão et al. [104, 105] describe a “city induction” process where from geospatial data and design principles a suitable set of urban design patterns are inducted (i.e., the rules and patterns are discovered). The result is a grammar for generating plausible models of an urban space. The methodology uses a discursive grammar which includes a shape grammar, a description grammar, and a set of heuristics. The shape grammar provides the geometric rules. The description grammar describes the design from other relevant (nongeometric) viewpoints. The set of heuristics is used to guide the generation of designs by comparing the description of the evolving design with the description of the desired house. The generation process is fast and deterministic—thus suitable for a web-based implementation.

4.2. Inverse Parameter Estimation. Several inverse procedural modeling papers have described a set of assumed building and facade structural priors and focus on estimating the parameter values for a particular instantiation. In the following, we describe several exemplary works at the building and at the facade levels. For instance, Bekins and Aliaga [106] and Aliaga et al. [10] propose a general grammar-based structure for an arbitrary building (Figure 9). The building is assumed to have a unique ground floor, followed by several intermediate floors, and a unique top floor. Each

floor is then further divided into facades, windows, and doors. Given a new building model reconstructed using an interactive photogrammetry, for example, its components (e.g., windows, doors, floors, facades, etc.) are labeled by the user. The system then determines the parameter values for a particular instantiation of the general building grammar. The user can then interactively, and automatically, apply the grammar to any new building structure resulting in a building of similar “style” as the original building.

Vanegas et al. [107] present a passive image-based method that exploits existing mapping and navigation databases in order to automatically create 3D building models. The method defines an a priori parameterized grammar for representing changes in building geometry that approximately follow the Manhattan-world assumption. This assumption states that there is a predominance of three mutually orthogonal directions in the scene [108]. This work extends previous Manhattan-world methods to robustly produce a single, coherent, complete geometric model of a building with partial textures. It discovers the parameter values for the known grammar so that it produces the same set of facade orientation changes observed in the captured images. Vanegas et al. [109] expand the methodology to LIDAR data.

For facades, Müller et al. [41] use the idea of splitting rules to infer the procedural structure of a facade from a single orthographic image as input. The general structure of the facade is assumed to follow a predetermined heuristic and the method finds important symmetry information. Global optimization is used as well to find adequate splitting lines to define rows and columns within the facade. Then, a library of detailed shape grammar rules are used to obtain the final model. Xiao et al. [42] propose a similar facade method that decomposes a photograph of a facade using top-down recursive subdivision, followed by a bottom-up merging with the detection with the detection of architectural bilateral symmetry and repetitive patterns. Teboul et al. [110] uses an assumed procedural shape prior (i.e., style) and a random exploration of the grammar space to optimize the sequence of derivation rules towards a best semantic-geometric interpretation of the observations. In later work, they apply machine learning techniques to refine the inference process for a focused style (i.e., the Parisian Haussmannian architecture) Teboul et al. [111].

4.3. Controlling Procedural Modeling. A key benefit of procedural modeling is its detail amplification—from a succinct set of rules and parameters, a seemingly very complex and organized output can be produced. However, this detail amplification is also its Achilles’ heel: controlling procedural modeling is very challenging and an open problem, especially in the context of large and complex 3D urban models.

Benes et al. [112] generalize the concept of the environment of a 3D model and creates a set of guides. Each guide defines a region in which a specific procedural model operates. These guides are connected by a set of links that “pass messages” between the guides. The guides, procedural models, and connectivity graph serve to improve the local

and global control of the overall modeling process. Shown examples include procedural structures such as ornamental patterns, streets networks, bridges, trees, and physics-based mass-spring systems.

Another group of methods attempts to control procedural modeling by using Monte Carlo Markov Chains (MCMC) [113]. MCMC/Metropolis-based [114] methods have been used in many vision and robotics applications [115]), in light transport [116, 117], and in rigid-body simulations [118]. More closely related to urban and architectural modeling, Alegre and Dellaert [119] present a probabilistic image-based approach to the semantic interpretation of building facades. Their method is based on constructing a Bayesian generative model from stochastic context-free grammars that encode knowledge about facades. This model combines low-level segmentation and high-level hierarchical labelling so that the levels reinforce each other and produce a detailed hierarchical partition of the depicted facade into structural blocks. MCMC sampling is used to approximate the posterior over partitions of a given image. Merrell et al. [120] (and also Yu et al. [121]) propose methods for searching in a parameterized interior design space for architectural models. Their interactive furniture layout system assists users by suggesting furniture arrangements that are based on interior design guidelines. The system quantifies layout guidelines using a weighted sum of scores and rapidly samples the underlying function using a hardware-accelerated Monte Carlo sampler. The results demonstrate an increase in the quality of furniture arrangements produced by participants with no prior training in interior design. A related data-driven method for architectural programming automatically generates building layouts for computer graphics applications [122]. Given a set of high-level requirements, an architectural program is synthesized using a Bayesian network trained on real-world data. The architectural program is realized in a set of floor plans, obtained through stochastic optimization. The floor plans are used to construct a 3D building with internal structure. While not strictly performing an inverse procedural modeling, it does follow a similar data-driven learning approach.

Talton et al. [103] take as input a stochastic context-free grammar and explore the possible strings generated by the grammar to find a production sequence that yields a target 3D shape. The production is generated by performing maximum a posteriori estimation over the space of possible productions from the grammar. The algorithm supports specifications of many forms, including geometric shapes and analytical objectives. It has been demonstrated on procedural models of trees, cities, buildings, and Mondrian paintings. The methodology renders/voxelizes the model to evaluate a cost function and yields, for example, a multi-building example result in about 14 minutes. Thus, although not interactive, interesting urban model shape control is enabled.

Vanegas et al. [123] propose a framework for adding intuitive high-level control to an existing urban procedural model. In particular, they add to the modeling arsenal an inverse modeling tool whereby users, unaware of the rules of the underlying urban procedural model, can alternatively specify arbitrary target indicators to control the modeling

process. The system itself will discover how to alter the parameters of the urban procedural model so as to produce the desired 3D output. In their paper, this process is called “inverse design.”

4.4. Shape Editing. A related set of methods are those of Bokeloh et al. [124, 125]. In Bokeloh et al. [124], they introduce a new structure-aware shape deformation technique. The key idea is to detect continuous and discrete regular patterns and ensure that these patterns are preserved during freeform deformation. They propose a variational deformation model that preserves these structures, and a discrete algorithm that adaptively inserts or removes repeated elements in regular patterns to minimize distortion. Later, in Bokeloh et al. [125], they present an approach to high-level shape editing that adapts the structure of the shape while maintaining its global characteristics. The main contribution is a new algebraic model of shape structure that characterizes shapes in terms of linked regular translational patterns and is parameterized by a small set of numerical parameters bounded by a set of linear constraints. These methods although they do not strictly extract procedural rules or parameters, they are able to edit a model, potentially created by a procedure, in a semantically sensible fashion.

5. Computer Graphics for Urban Design and Planning

As outlined in the introduction, there are numerous applications to city-scale 3D modeling. The key is providing both an intuitive and fast generation mechanism. In this section, we provide an insight into the use of computer graphics modeling for urban content generation in the domain of urban design and planning. Cities only occupy 2% of the Earth’s surface. Nevertheless, the concentrated presence of manmade structures, the high-resource consumption, and production of residuals and waste severely affect the environmental wellbeing. Improper urban planning and lack of awareness of the effect shape and building materials have on energy consumption and on the local weather systems can cause strong and unwanted impact. Moreover, cities are warmer than the surrounding area, due to “urban heat islands,” thus the aforementioned effects may be disproportionately large causing a host of ecological, economic, and social implications including inefficient energy use, unexpected health and mortality, economic losses, air quality deterioration, and reduced livability.

Yet, moving beyond the basic physics of the problem and into integrating such concepts with actual city planning and designing requires the development of a computational framework that can start pulling the heterogeneous interactions into a geometric generation and behavioral simulation platform. One key aspect for such a tool is enabling the rapid and flexible design of 3D models of cities that can be used in the simulations and can be shown to stakeholders. Such a task includes designing, simulating, and visualizing future cities for regional planning agencies so that they can evaluate building smart codes, sustainability plans, future

architectural designs, alternative transportation investments, land use regulations, and environmental protection policies.

The use of 3D computer graphics to build and show city models builds upon the notion that the targeted set of heterogeneous users are comfortable with understanding and gathering intuition from ground level and aerial renderings and visualizations of the 3D urban landscape. Such views are holistic representations that allow the viewer to perceive various aspects of a place in a coherent way. User studies have shown the receptiveness of visualization and interactive participatory collaborations as a way to simultaneously communicate results to the large variety of stakeholders often present in city designing or modeling meetings [126, 127]. However, creating such models requires considering the relationships between many parameters and effects, spanning multiple disciplines. In particular, we focus on the interplay of the city itself (e.g., its socioeconomic aspects) with the surrounding environment (e.g., its meteorological aspects) while also considering greening and the ecosystem. For example, building heights and interbuilding distances affect urban weather, but also population density, vegetation, and energy consumption, as well as suitability for solar energy. The interaction between these aspects must be addressed collectively and as an integrated system. In isolation, each of these aspects is understood to a limited degree. However, understanding their interaction is a significant and major intellectual challenge.

Recently, we have seen a proliferation of methods that start to blur the boundary between geometric modeling and scientific simulations, especially pertaining to smart city design. In the following, we provide a sampling of such activities in an effort to make the reader aware of this cross-disciplinary endeavor and to foment more such research and development. The below references are by no means complete but attempt to help bridge the gap between computer graphics and other domains for which cross-disciplinary activities are emerging.

5.1. Urban Weather. Meteorological simulations that explicitly consider the effects of the geometry of the urban landscape are significantly different from simulations over other natural land use land cover (LULC). This is primarily due to the unique physical property of manmade materials and has drawn increased research interest in recent years [128–131]. The Urban Heat Island is caused by the high heat capacity and low albedo of concrete buildings. Tall buildings also lead to the higher roughness and displacement length that alters regional surface winds and the atmospheric boundary layer convergence/divergence patterns. Moreover, urban landscape generally has lower evaporation/transpiration or latent heat flux as compared to the surrounding rural region. This creates spatial gradients in surface heat fluxes, which, combined with changes in boundary layer convergence, can create zones of preferential convection, and mesoscale weather patterns and climatic regimes (e.g., [132]). Most urban weather simulation methods assume either a very simple urban geometrical model or offload all modeling-to-time-consuming manual efforts [133].

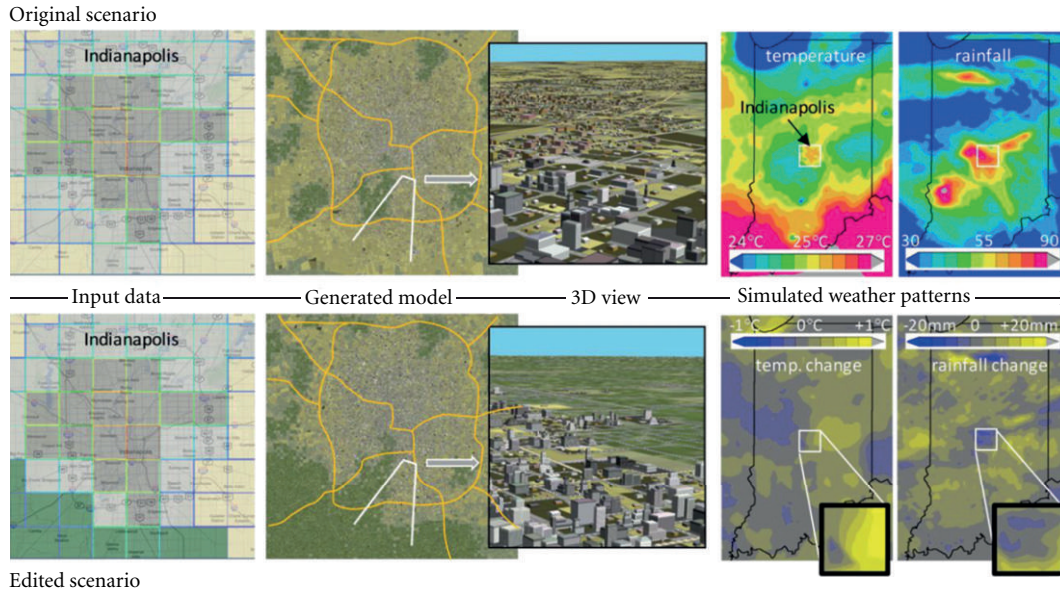


FIGURE 10: Future city meteorology. Top: original urban scenario for Indianapolis, Indiana (USA). Bottom: hypothetical (edited) urban scenario where the southwest corner became parks. Using land use/land cover data (left column), complemented by population and terrain data, this prototype automatically produces a plausible 3D city model (second and third columns) from which urban morphology parameters are extracted for a regional weather simulation over Indiana (fourth and rightmost columns). In this example, the proposed change increases city temperature and decreases city rainfall (image from [11]).

One effort for bridging the gap between meteorological simulations and procedural geometric design is described in Aliaga et al. [11] (Figure 10). In this work, they describe a visualization-based decision tool that enables exploring the link between urban land use and urban weather, in particular predicting and visualizing changes in urban temperature, precipitation, and humidity. The approach uses an interactive tool to quickly and automatically produce plausible detailed 3D city models by means of a hybrid computational simulation of urban behavior and procedural urban geometry. From the city model, urban morphology parameters are efficiently computed and used by our custom meteorological simulator which considers the influence of the urban landscape. The result is a compelling visualization ability for understanding the complex feedback between urban land use and the regional meteorology of current cities and of potential future cities with desired greening patterns.

5.2. Urban Ecosystem. Cities all over the world are challenged by increasing the quality of life of urban citizens in order to ensure sustainable urban development. One aspect of significant concern is designing and ensuring the sustainability of the urban ecosystem of plants and vegetation. However, it is quite challenging to devise successful policies for integrating environmental aspects. Within urban planning, Wissen-Hayek et al. [34] describe one representative approach to integrate ecological parameters into urban design using a procedural shape grammar driven modeling and visualization system. Design specifications and ecological goals are explored and given in the Masterplan of MASDAR City (Saudi Arabia) as an application example. They used the concept of ecosystem services to break down the ecological

process knowledge to design rules and meaningful, quantifiable spatial indicators. Results demonstrate the application of the proposed approach covering different planning scales (e.g., district and building level). The integrated model suits as an assessment tool that can be used to test urban design alternatives on the ecological functioning as a starting point for architects.

From a computer graphics perspective, Beneš et al. [36] explore designing a geometric city model that uses a user-guided simulation and procedural system for integrating plants into the interactive design of the 3D urban models. This approach uses as input 3D geometry of an urban layout from which it infers initial conditions and parameters of procedural rules. A spatially varying manageability level defines the amount of influence between the wild ecosystem simulation, where the plants compete for resources and seed freely, and the managed ecosystem, where nearly no seeding is allowed and the plants grow only under well-defined conditions. The wild ecosystems are handled by a simulation of plant competition for resources, whereas the procedural generation is based on an expandable set of behavioral rules of owners and typical plant management. The system provides a high degree of controllability and works tightly with an urban simulation system. Results show various examples, such as plant development over time in managed and unmanaged areas, effect of procedural rules on the plant distribution, and the effect of changing the level of manageability and the plant distribution.

If we focus more on the geometric modeling of the individual plants or trees, many papers have been published. Deussen and Lintermann [134] provide a summary of many techniques. One recent work is that of Pirk et al. [37], where

a dynamic tree modeling and representation technique allows complex tree models to interact with their environment. This method uses changes in the light distribution and proximity to solid obstacles and other trees as approximations of biologically motivated transformations on a skeletal representation of the tree's main branches and its procedurally generated foliage. Benes et al. [35] present a novel technique to create scenes populated by virtual plants that communicate actively with the environment and attempt to generate an optimal spatial distribution dynamically adapting to neighboring plants, user defined obstacles, light, and gravity.

5.3. Urban Radiation and Energy. In today's world of high energy consumption and waste production, addressing efficiency and sustainability of urban structures is of prime importance. With regards to general urban radiation and energy consumption, we highlight a demonstrative work that ties-in with computer graphics. Robinson et al. [135] describe a system called "CitySim" that focuses on simulating buildings' energy flows. The paper also describes work under way to model energy embodied in materials as well as the flows of water and waste and interrelationships between these flows; likewise their dependence on the urban climate. This tool, as well as SUNtool [136], exploits the use of a radiosity-based algorithm. Radiosity, though originating from engineering, also has a strong presence in computer graphics research (e.g., Greenberg et al. [137], Smits et al. [138], Lischinski et al. [139], and many more papers).

Two recent books [140, 141] provide a broad introduction to this field. Robinson's book [140] is the first book to directly address the physics of urban sustainability and how urban sustainability may be (geometrically) modeled and optimized. Comprehensive techniques for the modeling and optimization of the urban metabolism are also described, together with means for defining sustainability as the fitness function to be optimized. Beckers [141] proposes a multidisciplinary and systematic approach involving disciplines as different as meteorology, geography, computer science, architecture and urban engineering systems, all surrounding the essential problem of urban solar radiation. It collects the points of view of 18 specialists from around the world on the interaction between solar energy and constructions, combining territorial, urban, and architectural scales to better regulate energetic efficiency and light comfort for the sustainable city.

6. Conclusions

In this paper, we have collected and organized research which addresses the multidisciplinary challenge of designing realistic and plausible 3D models of buildings, urban spaces, and smart cities in general. Although this paper includes works mostly from a computer graphics perspective, the challenges involved require knowledge from several disciplines. Meeting these challenges is of interest to a wide audience spanning from computer scientists, to urban planners, to emergency management, and to visualization efforts. The proliferation of data via the Internet as well as the rapid

increase in urbanization has both fomented work along this line of research and as enabled it. As simulation models and methods are becoming increasingly sophisticated and better able to represent the complex processes occurring in urban spaces, we must also develop the 3D modeling and design engines to produce the necessary geometric input. Moreover, we should exploit the high performance of today's computing and graphics abilities to yield interactive design tools for improving decision making.

The design and simulation of future "smart cities" is very useful to regional planning agencies to evaluate sustainability, architectural designs for future growth, alternative transportation investments, land use regulations, and environmental protection policies. Sustainable city modeling is an effort of worldwide concern covering many different aspects and a large collection of very diverse cities. These activities span large distances, many cultural backgrounds, and numerous professional, scientific, and governmental communities with varying levels of involvement. Across the globe, a variety of future city and sustainability-related research activities are occurring. We hope to continue the awareness and relevancy of methods pertaining to computer graphics.

References

- [1] P. Musialski, P. Wonka, D. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer, "A Survey of Urban Reconstruction," in *Proceedings of European Association for Computer Graphics (EUROGRAPHICS '12)*, p. 27, 2012, State-of-the-Art-Report.
- [2] O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr, and P. Prusinkiewicz, "Realistic modeling and rendering of plant ecosystems," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, pp. 275–286, July 1998.
- [3] P. Mueller, P. Wonka, S. Haegler, A. Ulmer, and L. van Gool, "Procedural modeling of buildings," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 614–623, 2006.
- [4] D. G. Aliaga, B. Beněš, C. A. Vanegas, and N. Andryscio, "Interactive reconfiguration of urban layouts," *IEEE Computer Graphics and Applications*, vol. 28, no. 3, pp. 38–47, 2008.
- [5] D. G. Aliaga, C. A. Vanegas, and B. Beneš, "Interactive example-based urban layout synthesis," *ACM Transactions on Graphics*, vol. 27, no. 5, article 160, 2008, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques in Asia (ACM SIGGRAPH '08).
- [6] C. A. Vanegas, D. G. Aliaga, B. Beneš, and P. Waddell, "Visualization of simulated urban spaces: Inferring parameterized generation of streets, parcels, and aerial imagery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 424–435, 2009.
- [7] M. Wegener, "Overview of land use transport models," in *Handbook of Transport Geography and Spatial Systems*, pp. 127–146, 2004.
- [8] Y. Schwartzman and A. Borning, "The indicator browser: a web-based interface for visualizing UrbanSim simulation results," in *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, January 2007.

- [9] O. Stava, B. Beneš, R. Měch, D. G. Aliaga, and P. Krištof, "Inverse procedural modeling by automatic generation of L-systems," *Computer Graphics Forum*, vol. 29, no. 2, pp. 665–674, 2010.
- [10] D. G. Aliaga, P. A. Rosen, and D. R. Bekins, "Style grammars for interactive visualization of architecture," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 786–797, 2007.
- [11] D. Aliaga, C. Vanegas, M. Lei, and D. Niyogi, "Visualization-based Decision Tool for Urban Meteorological Modeling," *Environment & Planning B*. In press.
- [12] W. Weaver, "Science and complexity," *American Scientist*, vol. 36, pp. 536–544, 1948.
- [13] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, October 2007.
- [14] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 72–79, October 2009.
- [15] J. M. Frahm, P. Georgel, D. Gallup et al., "Building Rome on a cloudless day," in *Proceedings of the European Conference on Computer Vision*, pp. 368–381, 2010.
- [16] Y. I. H. Parish and P. Müller, "Procedural modeling of cities," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH '01)*, pp. 301–308, August 2001.
- [17] M. Batty, *Cities and Complexity: Understanding Cities With Cellular Automata, Agent-Based Models, and Fractals*, The MIT Press, Cambridge, Mass, USA, 2007.
- [18] S. Al-Kheder, J. Wang, and J. Shan, "Fuzzy inference guided cellular automata urban-growth modelling using multi-temporal satellite images," *International Journal of Geographical Information Science*, vol. 22, no. 11-12, pp. 1271–1293, 2008.
- [19] P. Waddell, "UrbanSim: modeling urban development for land use, transportation, and environmental planning," *Journal of the American Planning Association*, vol. 68, no. 3, pp. 297–343, 2002.
- [20] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley, *Texturing and Modeling*, Academic Press, Orlando, Fla, USA, 3rd edition, 2003.
- [21] M. Davis, R. Sigal, E. J. Weyuker, and M. D. Davis, *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, Academic Press, New York, NY, USA, 1994.
- [22] M. Sipser, *Introduction to the Theory of Computation*, Course Technology, Boston, Mass, USA, 1996.
- [23] G. Ehrig, H. K. Engels, and G. Rozenberg, *Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages and Tools*, World Scientific Publishing, River Edge, NJ, USA, 1999.
- [24] G. Stiny and W. J. Mitchell, "The palladian grammar," *Environment and Planning B*, vol. 5, no. 1, pp. 5–18, 1978.
- [25] D. E. Knuth, "Semantics of context-free languages," *Mathematical Systems Theory*, vol. 2, no. 2, pp. 127–145, 1968.
- [26] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, Springer, New York, NY, USA, 1990.
- [27] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, "Instant architecture," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 669–677, 2003.
- [28] R. Mech and P. Prusinkiewicz, "Visual models of plants interacting with their environment," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 397–410, August 1996.
- [29] P. Prusinkiewicz, M. James, and R. Mech, "Synthetic topiary," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*, pp. 351–358, New York, NY, USA, 1994.
- [30] A. R. Smith, "Plants, fractals, and formal languages," *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3, pp. 1–10, 1984, *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*.
- [31] C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*, Cess Center for Environmental, 1977.
- [32] J. Halatsch, A. Kunze, and G. Schmitt, "Using shape grammars for master planning," in *Proceedings of the Design Computing and Cognition (DCC '08)*, pp. 655–673, 2008.
- [33] S. Marshall, *Streets and Patterns*, Routledge, New York, NY, USA, 2004.
- [34] U. Wissen-Hayek, N. Neuenschwander, J. Halatsch, and A. Grêt-Regamey, "Procedural modeling of urban green space pattern designs taking into account ecological parameters," in *Proceedings of the 28th Conference on Education in Computer Aided Architectural Design in Europe (eCAADe '10)*, 2010.
- [35] B. Beneš, N. Andryscio, and O. Stava, "Interactive modeling of virtual ecosystems," in *Proceedings of the 5th Eurographics Workshop on Natural Phenomena*, pp. 9–16, 2009.
- [36] B. Beneš, M. A. Massih, P. Jarvis, D. G. Aliaga, and C. A. Vanegas, "Urban ecosystem design," in *Proceedings of the 15th ACM Symposium on Interactive 3D Graphics and Games*, pp. 167–174, February 2011.
- [37] S. Pirk, O. Stava, J. U. Kratt et al., "Plastic trees: interactive self-adapting botanical tree models," *ACM Transactions on Graphics*, vol. 31, no. 4, article 50, 2012.
- [38] T. Kelly and P. Wonka, "Interactive architectural modeling with procedural extrusions," *ACM Transactions on Graphics*, vol. 30, no. 2, Article ID 14, pp. 1–15, 2011.
- [39] P. Merrell, "Example-based model synthesis," in *Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D '07)*, pp. 105–112, May 2007.
- [40] P. Merrell and D. Manocha, "Continuous model synthesis," *ACM Transactions on Graphics*, vol. 27, no. 5, article 158, 2008.
- [41] P. Müller, G. Zeng, P. Wonka, and L. van Gool, "Image-based procedural modeling of facades," *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 85, 2007.
- [42] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, and L. Quan, "Image-based façade modeling," *ACM Transactions on Graphics*, vol. 27, no. 5, article 161, 2008.
- [43] S. Haegler, P. Wonka, S. M. Arisona, L. van Gool, and P. Müller, "Grammar-based encoding of facades," *Computer Graphics Forum*, vol. 29, no. 4, pp. 1479–1487, 2010, *Proceedings of the Eurographics Symposium on Rendering*.
- [44] P. J. Birch, S. P. Browne, V. J. Jennings, A. M. Day, and D. B. Arnold, "Rapid procedural-modelling of architectural structures," in *Proceedings of the Virtual Reality, Archeology, and Cultural Heritage (VAST '01)*, pp. 187–196, November 2001.
- [45] M. Lipp, P. Wonka, and M. Wimmer, "Interactive visual editing of grammars for procedural architecture," *ACM Transactions on Graphics*, vol. 27, no. 3, article 102, 2008.
- [46] J. Legakis, J. Dorsey, and S. Gortler, "Feature-based cellular texturing for architectural models," in *Proceedings of the Computer Graphics Annual Conference and Interactive Techniques (ACM SIGGRAPH '01)*, pp. 309–316, August 2001.

- [47] E. Vouga, M. Höbinger, J. Wallner, and H. Pottmann, "Design of self-supporting surfaces," *ACM Transactions on Graphics*, vol. 31, no. 4, article 87, 2012, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '12).
- [48] H. Pottmann, Y. Liu, J. Wallner, A. Bobenko, and W. Wang, "Geometry of multi-layer freeform structures for architecture," *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 1276458, 2007, Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '07).
- [49] H. Pottmann, A. Asperl, M. Hofer, and A. Kilian, *Architectural Geometry*, Bentley Institute Press, 2007.
- [50] H. Pottmann, A. Schiftner, P. Bo et al., "Freeform surfaces from single curved panels," *ACM Transactions on Graphics*, vol. 27, no. 3, article 76, 2008, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '08).
- [51] M. Eigensatz, M. Kilian, A. Schiftner, N. J. Mitra, H. Pottmann, and M. Pauly, "Paneling architectural freeform surfaces," *ACM Transactions on Graphics*, vol. 29, no. 4, article 45, 2010, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '10).
- [52] P. Bo, H. Pottmann, M. Kilian, W. Wang, and J. Wallner, "Circular arc structures," *ACM Transactions on Graphics*, vol. 30, no. 101, pp. 1–11, 2011, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '11).
- [53] L. Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the Art in Example-based texture synthesis," *EUROGRAPHICS STAR*, 2009.
- [54] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 1033–1038, September 1999.
- [55] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 341–346, August 2001.
- [56] V. Kwatra, A. Schödl, I. A. Essa, G. Turk, and A. F. Bobick, "Graphcut textures: image and video synthesis using graph cuts," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 277–286, 2003.
- [57] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the Computer Graphics Annual Conference and Interactive Techniques (ACM SIGGRAPH '01)*, pp. 327–340, August 2001.
- [58] S. A. Groenewegen, R. M. Smelik, K. J. de Kraker, and R. Bidarra, "Procedural city layout generation based on urban land use models," in *Proceedings of the European Association for Computer Graphics (EUROGRAPHICS '09)*, 2009.
- [59] M. Lipp, D. Scherzer, P. Wonka, and M. Wimmer, "Interactive modeling of city layouts using layers of procedural content," *Computer Graphics Forum*, vol. 30, no. 2, pp. 345–354, 2011.
- [60] L. Krecklau, C. Manthei, and L. Kobbelt, "Procedural interpolation of historical city maps," *Computer Graphics Forum*, vol. 31, no. 2, pp. 691–700, 2012.
- [61] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang, "Interactive procedural street modeling," *ACM Transactions on Graphics*, vol. 27, no. 3, article 103, 2008.
- [62] E. Galin, A. Peytavie, N. Maréchal, and E. Guérin, "Procedural generation of roads," *Computer Graphics Forum*, vol. 29, no. 2, pp. 429–438, 2010, Proceedings of European Association for Computer Graphics.
- [63] E. Galin, A. Peytavie, E. Guérin, and B. Benes, "Authoring hierarchical road networks," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2021–2030, 2011, Proceedings of Pacific Graphics.
- [64] P. Panerai, J. Castex, J. Depaule, and I. Samuels, *Urban Forms: Death and Life of the Urban Block*, Architectural Press, 2004.
- [65] D. Thadani, L. Krier, and A. Duany, *The Language of Towns & Cities: A Visual Dictionary*, Rizzoli, 2010.
- [66] D. Parolek, K. Parolek, and P. Crawford, *Form-Based Codes: A Guide for Planners, Urban Designers, Municipalities, and Developers*, John Wiley & Sons, New York, NY, USA, 2008.
- [67] C. Vanegas, T. Kelly, B. Weber, J. Halatsch, D. Aliaga, and P. Mueller, "Procedural generation of parcels in urban modeling," *Computer Graphics Forum*, vol. 31, no. 2, pp. 681–690, 2012, Proceedings of the European Association for Computer Graphics.
- [68] M. Honda, K. Mizuno, Y. Fukui, and S. Nishihara, "Generating autonomous time-varying virtual cities," in *Proceedings of the International Conference on Cyberworlds*, pp. 45–52, November 2004.
- [69] G. Kelly and H. McCabe, "Citygen: an interactive system for procedural city generation," in *Proceedings of the 5th International Conference on Game Design and Technology*, pp. 8–16, 2007.
- [70] B. Weber, P. Müller, P. Wonka, and M. Gross, "Interactive geometric simulation of 4D cities," *Computer Graphics Forum*, vol. 28, no. 2, pp. 481–492, 2009.
- [71] P. Waddell, A. Borning, M. Noth, N. Freier, M. Becke, and G. Ulfarsson, "Microsimulation of urban development and location choices: design and implementation of UrbanSim," *Networks and Spatial Economics*, vol. 3, no. 5, pp. 43–67, 2003.
- [72] C. A. Vanegas, D. G. Aliaga, B. Beneš, and P. A. Waddell, "Interactive design of urban spaces using geometrical and behavioral modeling," *ACM Transactions on Graphics*, vol. 28, no. 5, article 111, 2009.
- [73] P. Torrens and D. O'Sullivan, "Cellular automata and urban simulation: Where do we go from here?" *Environment & Planning B*, vol. 28, no. 2, pp. 163–168, 2001.
- [74] K. C. Clarke and L. J. Gaydos, "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*, vol. 12, no. 7, pp. 699–714, 1998.
- [75] J. Portugali, *Self-Organization and the City*, Springer, Berlin, Germany, 2000.
- [76] D. McFadden, "Conditional logit analysis of qualitative choice behavior," in *Frontiers in Econometrics*, pp. 105–142, 1974.
- [77] S. Putman, *Integrated Urban Models: New Research and Applications of Optimization and Dynamics*, Pion, London, UK, 1991.
- [78] A. Anas and I. Kim, "General equilibrium models of polycentric urban land use with endogenous congestion and job agglomeration," *Journal of Urban Economics*, vol. 40, no. 2, pp. 232–256, 1996.
- [79] F. Martinez, "MUSSA: land use model for Santiago city," *Transportation Research Record*, no. 1552, pp. 126–134, 1996.
- [80] A. de Palma, N. Picard, and P. Waddell, "Discrete choice models with capacity constraints: an empirical analysis of the

- housing market of the greater Paris region,” *Journal of Urban Economics*, vol. 62, no. 2, pp. 204–230, 2007.
- [81] P. Waddell and F. Ulfarsson, “Introduction to urban simulation: design and development of operational models,” in *Transport Geography and Spatial Systems*, vol. 5 of *Handbook in Transport*, pp. 203–236, Pergamon Press, 2004.
- [82] K. Train, *Discrete Choice Methods with Simulation*, Cambridge University Press, New York, NY, USA, 2nd edition, 2009.
- [83] H. Ševčíková, A. E. Raftery, and P. A. Waddell, “Assessing uncertainty in urban simulations using Bayesian melding,” *Transportation Research B*, vol. 41, no. 6, pp. 652–669, 2007.
- [84] K. Rosen, “Hedonic prices and implicit markets: product differentiation in pure competition,” *Journal of Political Economy*, vol. 82, pp. 34–55, 1974.
- [85] P. Waddell, B. J. L. Berry, and I. Hoch, “Residential property values in a multinodal urban area: new evidence on the implicit price of location,” *The Journal of Real Estate Finance and Economics*, vol. 7, no. 2, pp. 117–141, 1993.
- [86] W. Greene, *Econometric Analysis*, Pearson Education, Upper Saddle River, NJ, USA, 5th edition, 2002.
- [87] D. McFadden, “Modeling the choice of residential location,” in *Spatial Interaction Theory and Planning Models*, A. Karlqvist, L. Lundqvist, F. Snickars, and J. W. Wiebull, Eds., pp. 75–96, North Holland, Amsterdam, The Netherlands, 1978.
- [88] D. McFadden, “Econometric models of probabilistic choice,” in *Structural Analysis of Discrete Data With Econometric Applications*, C. F. Manski and D. McFadden, Eds., pp. 198–272, The MIT Press, Cambridge, Mass, USA, 1981.
- [89] P. Waddell, G. F. Ulfarsson, J. P. Franklin, and J. Lobb, “Incorporating land use in metropolitan transportation planning,” *Transportation Research Part A*, vol. 41, no. 5, pp. 382–410, 2007.
- [90] M. Batty, “Urban modeling in computer-graphic and geographic information system environments,” *Environment & Planning B*, vol. 19, no. 6, pp. 663–688, 1992.
- [91] M. Batty and S. Cole, “Time and space: geographic perspectives on the future,” *Futures*, vol. 29, no. 4-5, pp. 277–289, 1997.
- [92] M. Batty, P. Steadman, and Y. Xie, “Visualization in spatial modeling, Visualization in spatial modeling,” in *Proceedings of the ES Lab International Workshop on The Study of Artificial Complex Environments*, Centre for Advanced Spatial Analysis (UCL), London, UK, 2004.
- [93] D. Dorling, A. Barford, and M. Newman, “Worldmapper: the world as you’ve never seen it before,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 757–764, 2006.
- [94] D. A. Keim, S. C. North, and C. Panse, “CartoDraw: a fast algorithm for generating contiguous cartograms,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 1, pp. 95–110, 2004.
- [95] C. Panse, M. Sips, D. A. Keim, and S. C. North, “Visualization of geo-spatial point sets via global shape transformation and local pixel placement,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 749–756, 2006.
- [96] L. D. Pinnel, M. Dockrey, A. J. B. Brush, and A. Borning, “Design of visualizations for urban modeling,” in *Proceedings of the Joint EUROGRAPHICS and IEEE TCVC Symposium on Visualization (VisSym ’00)*, 2000.
- [97] R. Chang, G. Wessel, R. Kosara, E. Sauda, and W. Ribarsky, “Legible cities: focus-dependent multi-resolution visualization of urban relationships,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1169–1175, 2007.
- [98] J. Dykes and C. Brunsdon, “Geographically weighted visualization: Interactive graphics for scale-varying exploratory analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1161–1168, 2007.
- [99] T. Poggio and C. Koch, “Ill-posed problems in early vision: from computational theory to analogue networks,” *Proceedings of the Royal Society of London B*, vol. 226, no. 1244, pp. 303–323, 1985.
- [100] N. Mitra, L. Guibas, and M. Pauly, “Symmetrization,” *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 63, 2007, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’07).
- [101] M. Bokeloh, M. Wand, and H. P. Seidel, “A connection between partial symmetry and inverse procedural modeling,” *ACM Transactions on Graphics*, vol. 29, no. 4, article 104, 2010.
- [102] J. P. Park, K. H. Lee, and J. Lee, “Finding syntactic structures from human motion data,” *Computer Graphics Forum*, vol. 80, no. 8, pp. 2183–2193, 2011.
- [103] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. MěCh, and V. Koltun, “Metropolis procedural modeling,” *ACM Transactions on Graphics*, vol. 30, no. 2, article 11, 2011.
- [104] J. N. Beirão and J. P. Duarte, “Urban grammars: towards flexible urban design,” in *Proceedings of the 23rd Conference on Education in Computer Aided Architectural Design in Europe (eCAADe ’05)*, 2005.
- [105] J. N. Beirão, P. Nourian, and B. Mashhoodi, “Parametric urban design: an interactive sketching system for shaping neighborhoods,” in *Proceedings of 29th Conference on Education in Computer Aided Architectural Design in Europe (eCAADe ’11)*, pp. 225–234, 2011.
- [106] D. Bekins and D. G. Aliaga, “Build-by-number: rearranging the real world to visualize novel architectural spaces,” in *Proceedings of the IEEE Visualization (VIS ’05)*, pp. 143–150, October 2005.
- [107] C. A. Vanegas, D. G. Aliaga, and B. Beneš, “Building reconstruction using manhattan-world grammars,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’10)*, pp. 358–365, June 2010.
- [108] J. M. Coughlan and A. L. Yuille, “Manhattan world: compass direction from a single image by Bayesian inference,” in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV ’99)*, pp. 941–947, September 1999.
- [109] C. A. Vanegas, D. G. Aliaga, and B. Benes, “Automatic extraction of Manhattan-World building masses from 3D laser range scans,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 10, pp. 1627–1637, 2012.
- [110] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios, “Segmentation of building facades using procedural shape priors,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’10)*, pp. 3105–3112, June 2010.
- [111] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, “Shape grammar parsing via Reinforcement Learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2273–2280, 2011.
- [112] B. Benes, O. Stava, R. Mech, and G. Miller, “Guided procedural modeling,” *Computer Graphics Forum*, vol. 21, no. 3, pp. 325–334, 2011.

- [113] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC, New York, NY, USA, 1995.
- [114] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [115] F. Dellaert, "A sample of monte carlo methods in robotics and vision," in *Proceedings of the ISM International Symposium on the Science of Modeling*, 2003.
- [116] E. Veach and L. J. Guibas, "Metropolis light transport," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*, pp. 65–76, August 1997.
- [117] E. Veach and L. J. Guibas, "Metropolis light transport," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*, pp. 65–76, ACM Press/Addison-Wesley Publishing, August 1997.
- [118] S. Chenney and D. A. Forsyth, "Sampling plausible solutions to multi-body constraint problems," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 219–228, ACM, July 2000.
- [119] F. Alegre and F. Dellaert, "A probabilistic approach to the semantic interpretation of building facades," in *Proceedings of the Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*, 2004.
- [120] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun, "Interactive furniture layout using interior design guidelines," *ACM Transactions on Graphics*, vol. 30, no. 4, Article ID 87, 2011.
- [121] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, "Make it home: automatic optimization of furniture arrangement," *ACM Transactions on Graphics*, vol. 30, no. 4, Article ID 86, 2011.
- [122] P. Merrell, E. Schkufza, and V. Koltun, "Computer-generated residential building layouts," *ACM Transactions on Graphics*, vol. 29, no. 6, Article ID 1866203, 2010, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '10).
- [123] C. Vanegas, D. Aliaga, B. Benes, and P. Waddell, "Inverse Design of Urban Procedural Models," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques in Asia (ACM SIGGRAPH '12)*, p. 11, 2012, ACM Transactions on Graphics.
- [124] M. Bokeloh, M. Wand, V. Koltun, and H. P. Seidel, "Pattern-aware shape deformation using sliding dockers," *Transactions on Graphics*, vol. 30, no. 6, Article ID 123, 2011, Proceedings of the SIGGRAPH Asia Conference.
- [125] M. Bokeloh, M. Wand, H. P. Seidel, and V. Koltun, "An algebraic model for parameterized shape editing," *ACM Transactions on Graphics*, vol. 31, no. 4, 2012, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '11).
- [126] R. Laing, A. M. Davies, D. Miller, A. Conniff, S. Scott, and J. Morrice, "The application of visual environmental economics in the study of public preference and urban green-space," *Environment and Planning B*, vol. 36, no. 2, pp. 355–375, 2009.
- [127] C. J. Pettit, C. M. Raymond, B. A. Bryan, and H. Lewis, "Identifying strengths and weaknesses of landscape visualisation for effective communication of future alternatives," *Landscape and Urban Planning*, vol. 100, no. 3, pp. 231–241, 2011.
- [128] H. Akbari, S. Menon, and A. Rosenfeld, "Global cooling: increasing world-wide urban albedos to offset CO₂," *Climatic Change*, vol. 94, no. 3–4, pp. 275–286, 2009.
- [129] M. P. McCarthy, M. J. Best, and R. A. Betts, "Climate change in cities due to global warming and urban effects," *Geophysical Research Letters*, vol. 37, no. 9, Article ID L09705, 2010.
- [130] V. Masson, "A physically-based scheme for the urban energy budget in atmospheric models," *Boundary-Layer Meteorology*, vol. 94, no. 3, pp. 357–397, 2000.
- [131] K. W. Oleson, G. B. Bonan, and J. Feddema, "Effects of white roofs on urban temperature in a global climate model," *Geophysical Research Letters*, vol. 37, no. 3, Article ID L03701, 2010.
- [132] D. Niyogi, C. Kishtawal, S. Tripathi, and R. S. Govindaraju, "Observational evidence that agricultural intensification and land use change may be reducing the Indian summer monsoon rainfall," *Water Resource Research*, vol. 46, Article ID W03533, 17 pages, 2010.
- [133] D. Niyogi, T. Holt, S. Zhong, P. C. Pyle, and J. Basara, "Urban and land surface effects on the 30 July 2003 mesoscale convective system event observed in the southern Great Plains," *Journal of Geophysical Research D*, vol. 111, no. 19, Article ID D19107, 2006.
- [134] O. Deussen and B. Lintermann, *Digital Design of Nature: Computer Generated Plants and Organics*, Springer, New York, NY, USA, 2005.
- [135] D. Robinson, F. Haldi, J. Kämpf et al., "Citysim: comprehensive micro-simulation of resource flows for sustainable urban planning," Proceedings of the 11th International Building Performance Simulation Association Conference (IBPSA '09), 2009.
- [136] D. Robinson, N. Campbell, W. Gaiser et al., "SUNtool—a new modelling paradigm for simulating and optimising urban sustainability," *Solar Energy*, vol. 81, no. 9, pp. 1196–1211, 2007.
- [137] D. P. Greenberg, M. F. Cohen, and K. E. Torrance, "Radiosity: a method for computing global illumination," *The Visual Computer*, vol. 2, no. 5, pp. 291–297, 1986.
- [138] B. E. Smits, J. R. Arvo, and D. H. Salesin, "Importance-driven radiosity algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 273–282, 1992, Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92).
- [139] D. Lischinski, F. Tampter, and D. P. Greenberg, "Combining hierarchical radiosity and discontinuity meshing," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*, pp. 199–208, August 1993.
- [140] D. Robinson, *Computer Modelling For Sustainable Urban Design: Physical Principles, Methods, and Applications*, Earthscan, 2011.
- [141] B. Beckers, *Solar Energy at Urban Scale*, ISTE Wiley, 2012.

