Visualization of Simulated Urban Spaces: Inferring Parameterized Generation of Streets, Parcels, and Aerial Imagery

Carlos A. Vanegas, Daniel G. Aliaga,
Bedřich Beneš, *Member*, *IEEE Computer Society*, and Paul Waddell

Abstract—Urban simulation models and their visualization are used to help regional planning agencies evaluate alternative transportation investments, land use regulations, and environmental protection policies. Typical urban simulations provide spatially distributed data about a number of inhabitants, land prices, traffic, and other variables. In this article, we build on a synergy of urban simulation, urban visualization, and computer graphics to automatically infer an urban layout for any time step of the simulation sequence. In addition to standard visualization tools, our method gathers data of the original street network, parcels, and aerial imagery and uses the available simulation results to infer changes to the original urban layout. Our method produces a new and plausible layout for the simulation results. In contrast with previous work, our approach automatically updates the layout based on changes in the simulation data and, thus, can scale to a large simulation over many years. The method in this article offers a substantial step forward in building integrated visualization and behavioral simulation systems for use in community visioning, planning, and policy analysis. We demonstrate our method on several real cases using a 200-Gbyte database for a 16,300-km^2 area surrounding Seattle.

Index Terms—Computer graphics, information visualization, picture/image generation, simulation output analysis.

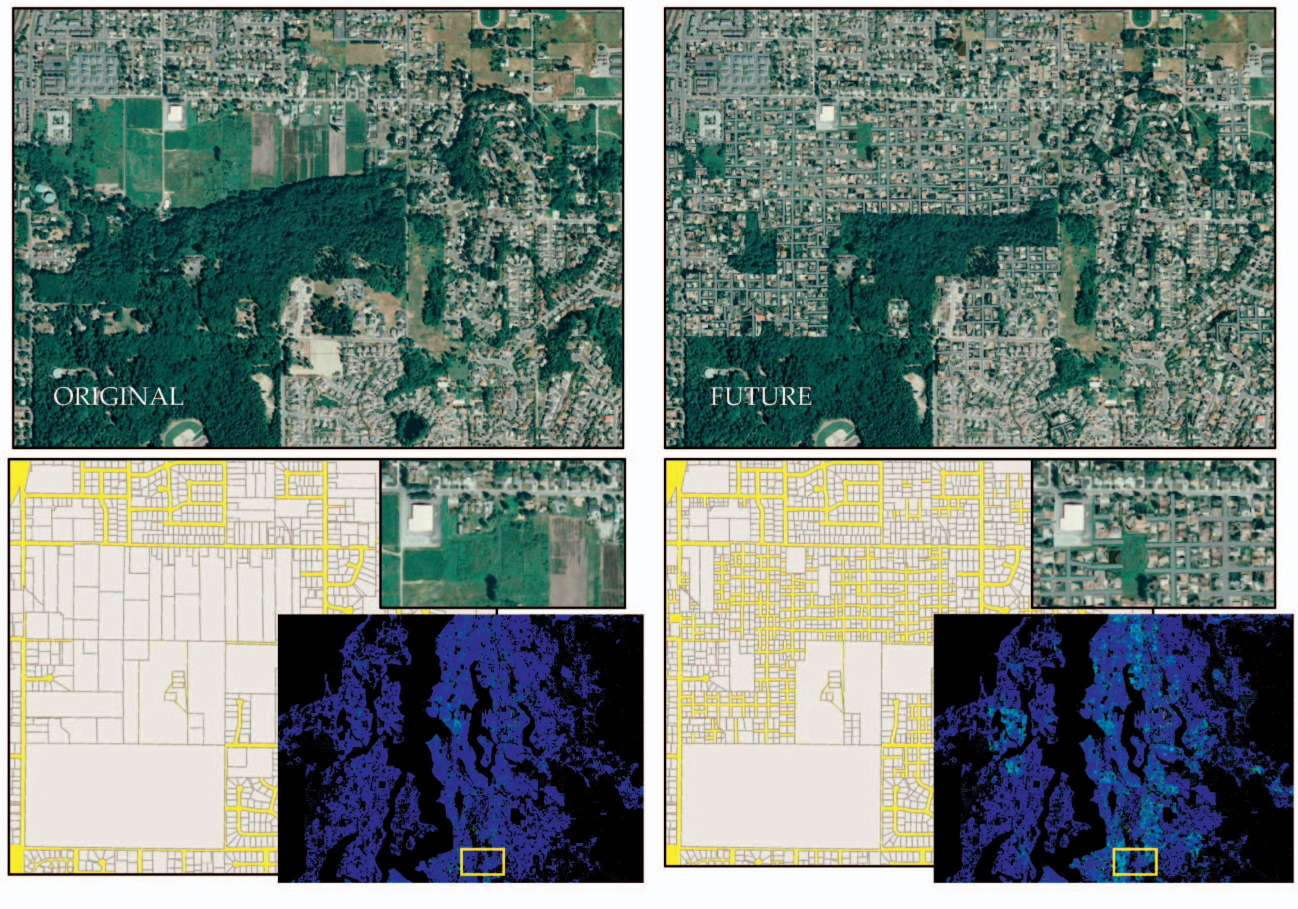
1 Introduction

Urban simulation models and the visualization of their results are increasingly being used in city, county, and regional planning agencies to assess alternative transportation investments, land use regulations, and environmental protection policies. The amount of data generated by such a simulation model over a long forecasting horizon and over a large scale (e.g., 10 to 30 years for a city of several million people) is overwhelming and, therefore, difficult to easily interpret for planners, policy makers, and the public and even for the modelers running the simulation. Visualization techniques are essential to extract useful information from the large mass of data generated by such simulations. However, to date, such simulation systems have been limited in their scope of visualization, in spite of providing very sophisticated economic and behavioral simulation engines. In this article and to the best of our knowledge, we propose the first method using the input data to an urban simulation, the output data of the simulation, and computer graphics techniques to automatically and interactively infer urban layouts (Fig. 1). A city can be represented by its urban layout, which we define to be the intricate collection of its man-made structures arranged into parcels, blocks, streets, and neighborhoods (e.g., aerial images together with Geographical Information System (GIS) vector data such as that provided by Google Maps, MapQuest, etc.). Since urban layouts are difficult to model by hand because they are very complex, large, and widespread, we use a simulation and an automatic inference approach to create a layout of an existing or a future urban space, therefore enabling multiple forms of visualizations for the aforementioned applications.

Our work builds on a synergy of efforts in urban simulation, urban visualization, and computer graphics (Fig. 2a). An urban simulation attempts to model and predict the complex socioeconomic interactions that govern the growth and development of an urban area. A typical output of such simulation is predictions of real estate development, prices, and location choices of households and firms at fine-grained levels of geography such as grid cells or parcels, over entire metropolitan areas and over planning horizons of several decades. Due to the magnitude and granularity of these simulations, it is difficult to adequately estimate all parameters, to automatically determine an exact city configuration at all stages of the simulation, and to intuitively visualize the significance of computed results. Urban visualization systems have been used to assist such simulations, but a typical scenario is that a manual postprocessing of simulation results needs to be done by a technical user. For instance, the user extracts summary indicators from the results, exports them from the simulation environment into a GIS, establishes relations of the indicators to existing GIS layers, and then chooses one of more thematic or choroplethic maps to display the spatial variation in the resulting indicators. Within the areas of computer graphics and visualization, significant effort has been devoted to the

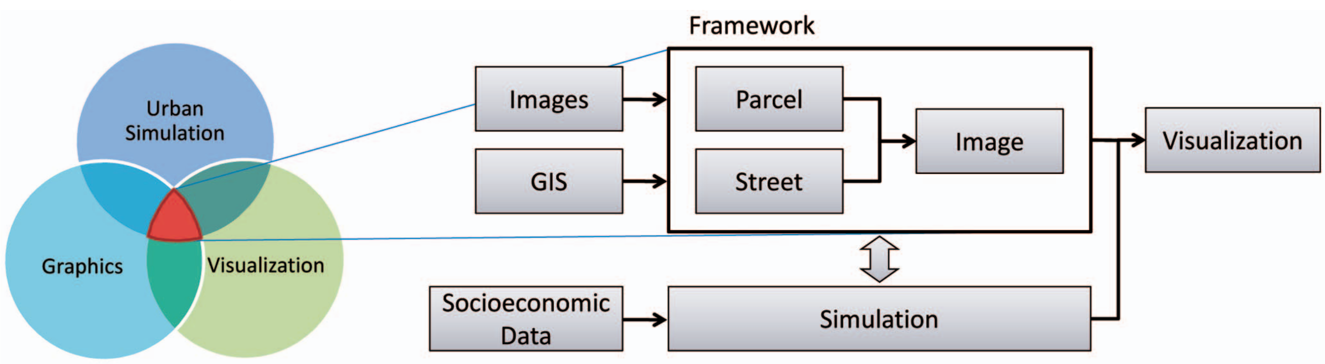
- *C.A. Vanegas and D.G. Aliaga are with the Department of Computer Science, Purdue University, 305 N. University Street, West Lafayette, IN 47907-2107. E-mail: {cvanegas, aliaga}@cs.purdue.edu.*
- *B. Beneš is with the Department of Computer Graphics Technology, Purdue University, 401 N. Grant Street, West Lafayette, IN 47907-2021. E-mail: bbenes@purdue.edu.*
- *P. Waddell is with the Evans School of Public Affairs, University of Washington, Rm 209A Parrington Hall, Box 353055, Seattle, WA 98195-3055. E-mail: pwaddell@u.washington.edu.*

Manuscript received 10 July 2008; revised 2 Oct. 2008; accepted 21 Oct. 2008; published online 29 Oct. 2008.
Recommended for acceptance by A. MacEachren.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2008-07-0094.
Digital Object Identifier no. 10.1109/TVCG.2008.193.



(a) (b)

Fig. 1. **Example visualizations.** In the top half are visualizations of the household/building growth of a large urban area spanning over 16,000 km^2 as per an urban simulation. In the bottom half, we focus on a particular area of the urban region and show how our algorithms have automatically and procedurally generated detailed urban layouts for augmenting the visualization. On (a), we show the initial state of the urban region, and on (b), we show the result after a 30-year simulation.



(a) (b)

Fig. 2. **Synergy and pipeline.** (a) Our work builds upon on a synergy of urban simulation, urban visualization, and computer graphics. (b) Our processing pipeline completes an urban simulation with visuals: we show a summary of the major components of our additional processing pipeline. Images and GIS data produce an initial urban layout that is, together with the simulation results, an input of our parcel, street, and image generator. This data is then visualized.

geometric modeling of nearly photorealistic urban structures (e.g., facades, buildings, and synthetic road networks). However, we have not found any work that attempts to infer the structures from an urban simulation and use them to help interpret results of the simulation. Although many works use manual input and thus could be made to follow the results of an urban simulation, the missing automaticity makes scaling such an approach to large areas challenging, especially for simulations spanning over many years.

The aforementioned processes have many limitations, not the least of which is the level of effort. As a result, too little visualization is actually done in practice, and this leads to diminished access to the simulation results, to reduced diagnostic capacity, and to reduced functionality as well.

While some automation to produce thematic maps and to incorporate them into the simulation platform has been performed [30], most users do not find these kinds of visualizations intuitive.

Our key observation is that by gathering data from the original aerial images together with GIS vector data and with urban simulation results, we can infer a new plausible urban layout for any time instance during the simulation. Therefore, automatically generated layouts can be used to enhance the already existing visualizations. An urban layout can be displayed in many different ways. Visualizing it as a map is probably the most intuitive way, as can be seen from the success of GIS and Internet-based mapping systems. This notion exploits that model users are comfortable with understanding and gathering intuition from aerial views of the urban landscape, whether from viewing aerial photographs or satellite images or from observing the cities they have flown over. Aerial views are holistic representations that allow the viewer to perceive various aspects of a place in a coherent way. Even without inspecting up close and in detail, the overall pattern of streets, the size of yards, the amount of vegetation, the mixture of nonresidential buildings, and the amount of open space all contribute to an intuition about the nature of the place. Similarly, parcel and street maps convey a sense of concreteness about an urban landscape and the nature of the development. By using an automatic generation of streets, parcels, and aerial imagery, the method in this article offers a substantial step forward in building integrated visualization and behavioral simulation systems for use in community visioning, planning, and policy analysis.

Our approach is to build upon existing visualization techniques of simulated urban spaces and to extend them by using new automatic inference algorithms for generating specialized content. Our method first selects important urban simulation variables and then uses them to produce visualizations. The input variables of urban simulators include values for the number of buildings, the number of households per parcel, the population within zones, etc. This data is traditionally displayed as color maps, graphs, diagrams, etc. The information that we generate from the output of such an urban simulator includes street network, parcels, and aerial images of buildings. Therefore, in addition to the aforementioned traditional visualization techniques, generated layouts can be used to enhance the already existing visualizations, exploiting the fact that people are efficient in understanding intuitive and content-rich images, such as vector-based maps and aerial imagery.

Our algorithms exploit the typical organization of an urban layout into streets, blocks, and parcels. These units are clustered and partitioned according to the uniqueness of their simulation parameter values. Then, as the simulation progresses, the method can appropriately extend or reuse a portion of the original urban layout, producing both plausible street and parcel network and plausible aerial images of the city. The effectiveness of this approach is improved as the size of the urban layout and its simulation grows (both spatially and temporally). To perform the urban simulation and without loss of generality, we use the publicly available UrbanSim simulation software [31], [34]. We have applied our techniques to a 30-year simulation of a 16,300-km^2 area surrounding Seattle using 200 Gbytes of data including high-resolution aerial imagery, per-year simulation data, and GIS data defining 1.4 million parcels.

Our major contributions include the following:

1. A methodology for enhancing the visualization of the results of urban simulations by using inferred higher level structural information.
2. A set of automatic and interactive algorithms for generating a visually plausible urban layout from the data produced by an urban simulation. Results have the full topology and connectivity of a parcel and street layout, can be analyzed and displayed, can be stored as images, can serve as input to algorithms that populate layouts with virtual buildings, and can be used in other applications.
3. The demonstration of real cases of visualization using about 200 Gbytes and 30 years of simulation results from the Seattle area.

2 Related Work

Our work builds upon current urban simulation models, urban visualization, and techniques in computer graphics. Although the term urban simulation has been used by some to describe 3D rendering of urban landscapes (e.g., [12], [19], [20], [21], [23], and [28]), in this article, the term corresponds to the use of behavioral or process modeling of the dynamic changes in urban activities and landscapes. When simulating changes of an existing urban space over time, a challenge is to adequately model and predict the complex interactions that govern its growth. The future structure of a city is dominated by both deterministic rules (e.g., population capacity of the city must grow) and organic rules (e.g., social, cultural, and economic interactions strongly influence how a city grows). Effectively and intuitively visualizing the results of such simulations is important for a variety of users. For instance, urban planning personnel need visualizations of changes to newly proposed neighborhoods or to large subsets of a city, emergency management professionals create models to train emergency response personnel in current and speculative urban layouts, and rapid prototyping and content creation technicians use partial information about urban layouts to generate plausible configurations of urban locations.

The most dominant paradigms to create urban simulations can be grouped into the following:

- *Early models attempting to represent emergent dynamics that adopted cellular automata as the modeling framework.* One of the most widely known is the Urban Growth Model [17] that has been applied to long-term changes in land cover patterns classified from remote sensing data [3].
- *Agent-based models that focus on examining cities as self-organizing complex systems.* Solutions have been designed to explore the emergent properties of agents with relatively simple behavioral rules embedded by a modeler [25].
- *Models exploiting a combination of urban economic analysis with statistical modeling of choices made by agents in the urban environment, such as households*

choosing residential locations. These models build on the pioneering work of McFadden on Random Utility Theory [18] and the further development of Discrete Choice Models [33].

Visualization has played an integral part in the development and use of urban simulations of all the aforementioned types. Batty [5] introduced various approaches that relate urban modeling, GIS, and computer graphics. The same author later described the impact of virtual reality and 3D visualization to GIS, and he has demonstrated this on a variety of complex examples [6]. Visualization has also been used for the purposes of education, exploration, explanation, and engagement [4]. A widely used urban visualization technique is cartograms, which use map shape warping to visualize relationships and values of urban and geospatial data sets (e.g., [10], [13], [22], and [24]). Alternatively, Chang et al. [7] use the multiple dimensions in their information visualization of urban relationships to produce a 3D visualization. Dykes and Brunsdon [11] introduced geowigs, a series of geographically weighted interactive graphics, to provide large-scale geographical environment visualization. In general, these and other approaches make use of visualization techniques including choroplethic maps 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, explicitly and automatically generating the geometry of urban layouts resulting from complex urban simulations is largely unexplored, for either explanatory or diagnostic purposes.

Computer graphics has been successfully used in several areas to extract implicit information from data and create improved visuals. In particular, significant effort has been devoted to the geometric modeling of urban structures per se. Many ideas of such urban modeling originated in visual models of plants [27], [29] and in building blocks of architectural structures (e.g., shape grammars [32]). One of the first techniques describing 3D city generation, via procedural modeling, was introduced by Parish and Mueller [23]. Merrell [19] proposes an example-based model synthesis using global search to resolve positional conflicts in urban 3D model generation. Methods have also been extended to the synthetic generation of buildings [20], [36], to the creation of buildings and façades imitating real-world structures [1], [21], and to generate synthetic road networks [8]. Watson and Mueller [35] provide a summary of such methods. These approaches focus on generating 3D content and do not automatically follow hints or rules as suggested by an urban simulation.

In contrast to previous work, in this paper, we focus on automatically inferring geometry and image content from urban simulation results. Hertzmann et al. [13] describe a framework for processing images by example and include modifying aerial photographs of cities. But they do not maintain or produce any underlying street or parcel information. In our own previous work [2], we describe a constraint-based system for modifying urban layouts (e.g., parcels and streets are interconnected such that moving the street changes the parcel and vice versa). While this method is aware of the structure of an urban layout, it provides no tools to synthesize a layout from urban simulation variables and no method to synthesize the imagery for new layouts. Instead, we analyze and interpret the results of urban simulations and then generate appropriate content.

3 Simulation of Urban Spaces

In this article, we simulate urban spaces and provide algorithms to visualize the intermediate and final steps of an urban simulation that models urban growth and development. Our visualization methodology is designed to work complementary to existing visualization methods and only requires data from simulations that model changes of population (e.g., households, buildings, etc.) per spatial unit of the simulation (e.g., grid cell, parcel, etc.). Our overall processing pipeline is shown in Fig. 2b. The pipeline takes as input high-resolution georeferenced aerial imagery, corresponding GIS street and parcel information (e.g., Shapefiles from ArcGIS), and the socioeconomic data of an urban area. Using this information, the simulation engine predicts the future value of several state variables of the urban space. Our visualization tools then enable generating detailed urban layouts at any point in the simulation and displaying results using various visual forms. Altogether, we provide an enhanced visualization of the data resulting from the urban simulation over a period of many years.

While our framework can be applied to various urban simulation packages (e.g., [8], [15], and [26]), we in particular use the publicly available UrbanSim simulation software (http://www.urbansim.org) developed by one of the authors as previous work [31], [34]. UrbanSim has become a standard tool for metropolitan land use and transportation planning and is currently used or is in the process of implementation for the simulation of numerous metropolitan areas in the US and Europe, for example, in the US by Detroit, Durham (NC), Honolulu, Houston (Texas), Phoenix, Salt Lake City, San Francisco, and Seattle and in Europe by Amsterdam, Brussels, Lyon, Paris, Rome, Tel Aviv, Turin, and Zurich. UrbanSim simulates the choices of individual households, businesses, and parcel landowners and developers, interacting in urban real estate markets and connected by a multimodal transportation system. This approach works with individual agents as is done in agent-based modeling and with very small cells as in the cellular automata approach or even buildings and parcels. But it differs from these approaches by drawing together choice theory, a simulation of real estate markets, and statistical methods to estimate model parameters and to calibrate uncertainty in the model system. It has been adapted and applied to numerous urban regions and extended to address issues such as how constraints on the availability of alternatives influence choices.

The simulation uses annual time steps, within which new residential and nonresidential buildings are virtually constructed by developers based on expected returns on investment and subject to development regulations, some of these by redeveloping existing buildings and most on vacant land. Households and jobs relocate and choose vacant space in existing buildings, and the interaction of agents competing for space drives real estate prices up or down. To visualize some state variables used in the simulation, it is necessary to compute such variables as indicators and assign them to a spatial unit of analysis. For example, households are updated by the simulation, but these are represented as individual records, one per household, and are therefore not conveniently visualized directly. Rather, the density of households or change in households, by some geography such as parcel or, more likely, a higher level of geography such as a zone or a neighborhood, is easier to interpret. On the other hand, some quantities such as the number of residential units constructed on a parcel can be more directly visualized and interpreted. Derived measures such as the predicted poverty rate in each neighborhood can also be readily computed and displayed visually. UrbanSim does not predict changes in the geometry of streets or parcels, however, so the process of subdivision of parcels must be inferred from other quantities that are predicted, such as the quantity of new real estate development on existing parcels. One of the key benefits of the visualization methods developed in this paper is the visual representation of the changes in street and parcel patterns. Although the visualization uses simulation results from this urban simulation engine, it can be applied to any other that also generates spatially detailed population changes.

```
Parcel and Street Generation (iterative form)
Let K be the desired number of parcels per block
Let N be the total number of new dwellings in B
FOR each city-block B to be partitioned
   Let P be the initial list of parcels in B
   Set P = {B} initially
   WHILE not enough parcels in B to hold the
          simulated population growth
      Get the parcel p' in P of largest area
      Compute the axes of the OBB of p' and choose one
          of them (A) such that Egress is maintained
      IF (N/|P|)> K
         Generate street geometry along A
         Add the geometry to the list of streets
      ENDIF
      Split p' along A, into p0 and p1
      Let P = P U {p0} U {p1} - p'
   ENDWHILE
ENDFOR

Image Generation
Let P be a new empty parcel
FOR each parcel Q in the original city
    Calculate the similarity f(P,Q) between P and Q
ENDFOR
Let Q=Q* be the parcel that maximizes f(P,Q)
Verify for P and for Q that their projection onto the
   longest axis of their OBB is monotonic
Define for P and for Q* a set SL of s scan lines that
   are uniformly separated, and perpendicular to their
   respective main axes.
For P and Q*, find the intersection points p0, p1,
   between each polygon and each scanline in the set
   SL associated to each of them
Generate n sample points along each scanline,
   uniformly separated between p0 and p1
Use the correspondence between the grids to warp the
   texture of Q* to P.
```

Fig. 3. Pseudocode summary of our algorithms.

4 Layout Generation

A variety of spatial transformations occurring to the parcels and to the buildings that occupy them can be inferred from the values of the state variables of the urban simulation system. The transformations that frequently take place in the layout include the replacement of buildings with newer and maybe larger structures (e.g., an empty parcel obtains a building or household structure), the division of a (large) parcel into smaller parcels, and the creation of streets (e.g., street growth through previously low-population-density areas). Our system detects changes in selected state variables and translates such changes into transformations to the initial street and parcel information and to the fragments of aerial imagery associated to each transformed parcel.

The aforementioned operations are grouped into (Section 4.1) parcel generation, (Section 4.2) street generation, and (Section 4.3) image generation. Fig. 3 provides a pseudocode summary of our methods that are individually explained in more detail in the remainder of this section.

4.1 Parcel Generation

We consider an existing parcel to change when the population assigned to it changes. To determine how to partition the parcel, we follow partitioning rules that satisfy both a set of desired properties (observed in real-world parcels) and the simulation data. The desired properties considered are the following: 1) parcels generally have egress (i.e., access to street), and thus, we let the user decide whether this property should be enforced, 2) city blocks are usually formed by rows of one or two parcels (e.g., parcels share a backyard with the parcels on the other side of the block), and hence, the subdivision should produce a similar arrangement, and 3) the contour of a parcel is most often a quadrilateral and often nearly rectangular.

Considering these properties, we devise an initial recursive algorithm that partitions an existing city block or a large parcel. Using the simulation data of a given urban area, we mark every parcel whose population has changed (e.g., the number of households has changed by at least one unit) as a candidate for subdivision. Subdivision is then attempted starting with the parcel that is largest in area. Most often the largest parcels to be subdivided correspond to empty lots that occupy an entire city block. Whether a parcel occupies an entire city block is determined from the set of parcels and the street network. The block-size parcel is then split into two parcels, each of which is recursively subdivided until reaching either the number of households specified by the simulation or a minimum parcel area. The minimum parcel area is computed automatically as the mean parcel area of all parcels within a chosen distance threshold. The number of households per parcel is updated during the subdivision by uniformly distributing household units between the new parcels. If egress is desired (first property), a split is made only if the resulting parcels will have access to streets. A general consequence of this constraint is that up to two rows of parcels occupying the original area are generated (second property).

Since our method performs a recursive binary division on an initial parcel, the contours of most of the newly formed parcels are nearly rectangular. An initial parcel with

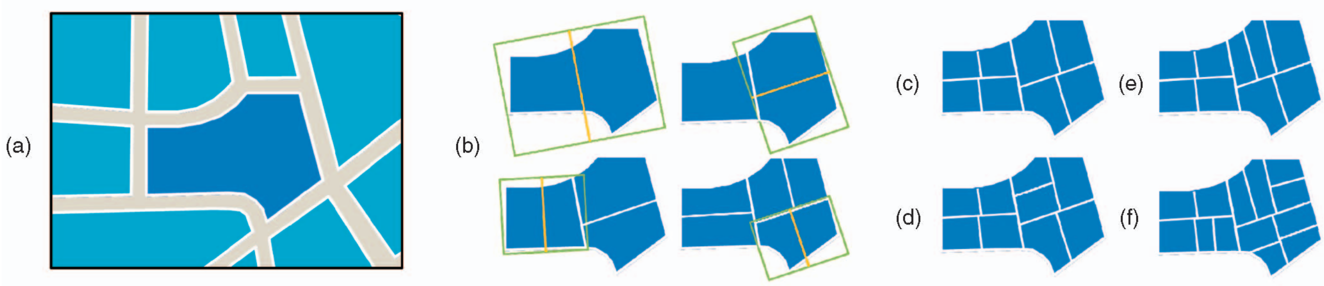


Fig. 4. **Parcels.** (a) A candidate large parcel corresponding to a city block is divided into smaller parcels (beige = street, light blue = parcels/city-blocks, and dark-blue = parcel to subdivide). (b) and (c) First four subdivisions of the original parcel yield to parcels with egress (green = OBB). (d) A subdivision that yields parcels without egress. (e) An alternative subdivision that yields parcels with egress. (f) The intermediate solution that results from continuing the recursive subdivision until the desired number of parcels is reached.

a more complex contour will be split by a line segment, and two new parcels with more simple contours will be generated. After a few subdivisions, most new parcels will be nearly rectangular. Also, the main axis of each parcel will be approximately orthogonal to the street to which it has access. As a result, this approach creates reasonable block divisions not only for rectangular initial blocks but also for blocks in a street network with organic (e.g., cul-de-sac) patterns. The end result is a set of parcels with an assigned number of households each: more than one household per parcel implies that the area contains a multifamily dwelling such as a townhouse or apartment complex. This information will be used later to find a plausible image fragment to populate the parcel.

Fig. 4 demonstrates a didactic example of this recursive process. In the middle of Fig. 4a, we observe the initial parcel geometry, which is then subdivided in several steps (Figs. 4b and 4c). In Fig. 4d, we observe a partitioning that yields a parcel without egress, and in Fig. 4e, we observe a partitioning that yields parcels with egress. Finally, Fig. 4f shows the intermediate results of continually subdividing the original region for parcels with egress. In addition to the above-described algorithm, streets must also be generated; otherwise, the egress rule will cause very long and narrow parcels shapes. The street generation is explained in the following section.

As a side note, real estate development that occurs in older inner-city neighborhoods often requires consolidation of multiple parcels in order to provide a sufficiently large site for a redevelopment project. Simulation of the consolidation of parcels will benefit from the geometric representation of parcels and their adjacencies, by allowing both economic factors and physical contiguity to be used to identify candidate parcels to consolidate. While our approach to generating parcel geometry supports such operations, the simulation of parcel consolidation remains to be implemented in future research.

4.2 Street Generation

In large real estate developments, new streets are built throughout the development zone so that every new parcel resulting from subdivision has egress. Our method pursues maintaining egress by generating new streets in parallel with parcel subdivision. Parcel subdivision generates a subdivision line for each step. For street generation, we in addition determine whether the new subdivision line corresponds to a new street segment (in order to generate new city blocks) or to a new parcel boundary segment (in order to generate more parcels within the same city block). To choose what type of subdivision line should be used, we consider the number of parcels per city block, a quantity that is given either by the simulation data or by a user-specified value. Furthermore, when the subdivision segment is selected to be a street segment, we generate city blocks whose aspect ratio is similar to that of neighboring city blocks. This is achieved by choosing among the possible division axes the one that would generate new blocks with an aspect ratio closer to the target aspect ratio. As a result, the new city blocks will resemble the shape of those in the nearby areas.

The number of additional parcels to be generated is a trigger to create new streets. When an original parcel is subdivided into two parcels, the ratio between the number of future parcels inside the block (given by the urban simulation) and the current number of parcels (which increases due to the parcel subdivision algorithm) is calculated. If such a ratio exceeds a specified value, then a street is generated between the two parcels.

The geometry for a new street is obtained by perturbing the initial subdivision line segment. We represent the initial subdivision segment as a polyline with a certain number of inner control points (e.g., two to five). Each of these points is deviated from the initial line in order to make the new street segment more similar to the streets in the nearby area. Attributes such as the average turning angle and the average length of close-by polyline segments are used to suggest how much each control point should be moved.

The street generation process subdivides each initial large parcel independently. Further processing is necessary in order to prevent streets with frequent dead ends or T-intersections. To avoid these cases, we detect street segments whose endpoints are relatively close to each other. Then, if the turning angle that would result from joining the two segments is not too sharp, the street segments are joined to form a single longer street.

The same stopping condition as with the initial algorithm, namely, either enforcing a smallest parcel area or reaching the targeted number of households, is used. The full recursive process creates a new network of parcels and streets of similar style to its surroundings that can be output in vector format.

Fig. 5 shows the recursive process of generating parcels and streets for the example in Fig. 4. The target number of

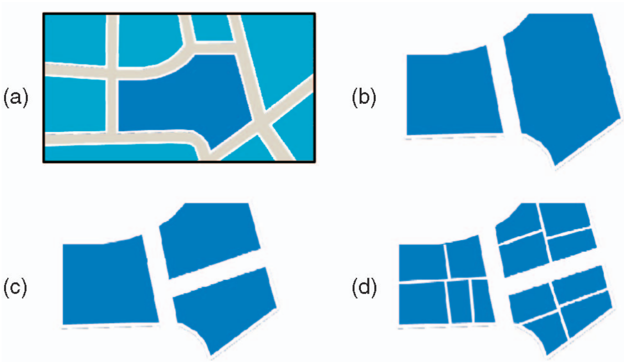


Fig. 5. **Parcel and streets.** (a) An initial parcel (from Fig. 4). (b) and (c) The first two subdivisions are chosen to be streets. (d) In further iterations, only more parcels per city block are produced.

additional parcels is 12, and the chosen approximate number of parcels per city block is four. During the first two iterations (Figs. 5a, 5b, and 5c), streets are produced, and the remaining number of parcels needed per city block is calculated (Fig. 5d).

Similar to parcel consolidation, streets can theoretically be removed from a layout. However, in our experience with urban simulation and examination of observed changes in street networks, this is a very rare phenomenon and, thus, we do not handle explicitly removing streets from the layout.

4.3 Parcel-Content Generation

Once all the streets and parcel geometries are created, we generate a plausible image content for each new parcel. As there is no information available for the specific details of the future structures, reusing existing aerial image fragments of parcels containing similar characteristics seems a plausible way of obtaining image samples to be viewed from similar aerial distances. While building styles may change significantly over a long time period (e.g., 100 years), for the typical simulation length that UrbanSim and related packages use (e.g., 10-30 years), we may assume that the building styles remain unchanged. Generating all structures inside each parcel is an alternative option, but this only changes the problem to that of inferring smaller structures.

To find the most appropriate existing (source) parcel image fragment to populate a new (destination) parcel, we define a similarity metric as the weighted sum of the similarity of simulation state values and the similarity of geometric shape. For simulation state values, we use all or some of households per parcel, year of the construction for buildings, and zoning classification (e.g., we desire a structure of the same number of households, of the same zoning, and closest in age to the newly created parcel). For geometric shape similarity, we compare their oriented bounding boxes (OBBs). The shapes with OBB pairing that is most similar in area and aspect ratio and respects the same side facing the street (assuming that information is available) are chosen.

Since the number of vertices of the source and destination parcels does not necessarily match, we must warp an a-sided polygon to a b-sided polygon. The selected source image fragment is texture-mapped onto the destination parcel geometry and rendered together with the rest of the layout.

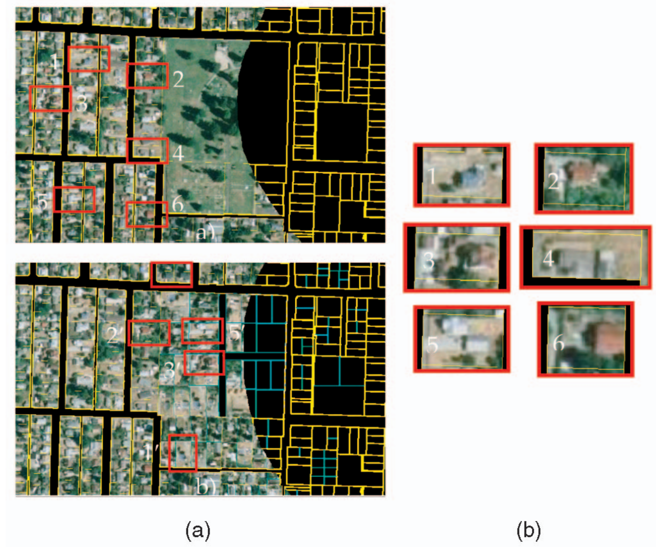


Fig. 6. **Images.** (a) Close-up of an original urban area from where parcel image fragments are extracted. (b) Close-up of newly generated area and a highlighting of some of their (reused) image content.

Our image-warping algorithm generates a correspondence between two arbitrary concave or convex polygons as long as their projection onto the central axis of their OBB is monotonic. Both OBBs are partitioned into scan lines perpendicular to their longest axis. A fixed number of points are regularly spaced, and the points are generated along both scan lines and culled to be inside each respective polygon. These points correspond one to one. The process is repeated for each scan line. The destination parcel is rendered using 1) the coordinates of the points to define a quadrilateral mesh and 2) the coordinates of the corresponding source parcel quadrilateral mesh to make texture coordinates. The effect is that the source texture is warped onto the destination parcel. A limitation of this warping approach is that there is no guarantee that the lengths of two corresponding scan lines are similar, and thus, visible image stretching could take place. However, our similarity metric has proved to avoid significant stretching almost completely when a large number of parcels of different shapes and sizes are present in the original city.

Fig. 6 shows an example of a parcel-content generation. Fig. 6a shows a small and restricted group of parcels from where image fragments are extracted and used to fill new parcels (Fig. 6b). As can be seen, the generated aerial image is similar to the original image and is its logical extension. This automatically generated layout is impossible to create using previous techniques.

5 Implementation Details

To perform our visualizations, our implementation reads GIS and simulation data directly from their respective sources, generates new content in vector form, and displays results using one of several rendering methods. In this section, we provide an overview of our data input processing, rendering methods, and user interactions.

Fig. 7. **Geometry generation.** (a) Original area within the test data set. (b) Using the data from the urban simulation and GIS input, we automatically compute a procedural subdivision of the area into new parcels and streets. (c) In order to enforce the egress rule, additional streets are generated.

5.1 Data Input Processing

The input to our method is a set of parcel and street geometries obtained from a GIS database, an aerial imagery of the targeted area, and the results of the urban simulation over the simulated time period. Our implementation directly reads the parcel and street geometries stored as standard Shape Files output by ArcGIS and the aerial images in single-resolution or multiresolution GeoTIFF files containing either state-plane coordinates (SPCs) or Universal Transverse Mercator (UTM) coordinates per image.

The data needed from the simulation, at a minimum, is a spatial grid with population changes specified per spatial unit. Our use of UrbanSim is done on a per-parcel basis, implying that we know the change of households per (original) parcel. Our program directly reads the binary output of UrbanSim.

5.2 Rendering

Our application enables the visualization of the simulation results using a variety of rendering methods. Choroplethic maps are used to display one of several simulation variables, either their absolute value or the change of value over time. In particular, the per-parcel UrbanSim variables of interest are the following: households, buildings, distance to large streets (e.g., arteries), distance to highways, land value, and year built of the constructions. Values can be displayed using linear or logarithmic color spectrum scales. To display the urban layouts generated by our algorithms, we can use either a map-style rendering (similar to Google Maps) or a map and image rendering providing plausible future aerial images of the simulated urban area. Examples of these rendering methods are in Section 6.

To enable the user to explore the data set, the program dynamically loads the data within the current field of view or within a chosen georeferenced bounding box. The most demanding rendering resource is the image database. Our implementation supports multiresolution imagery and dynamic loading. The user can choose to operate at one of several resolution levels, and our implementation dynamically loads/unloads the images needed to render the current view of the urban space and to produce the desired visualizations of future urban spaces. For example, our test database has GeoTIFF images of $5{,}000 \times 5{,}000$ pixels and of 50 Mbytes each. We can maintain interactive performance on a standard desktop PC for the entire data set by 1) downsampling all images to 64×64 (effectively yielding a single image of approximately $4{,}000 \times 4{,}000$ pixels covering the entire urban space) or 2) using a current few-tile close-up at the maximum resolution of $5{,}000 \times 5{,}000$ per image.

5.3 User Interaction

Our implementation can be used in a fully automatic fashion, requiring only minimal navigation through the urban space, selection of the visualization method, and optional specification of the maximum parcels per block and typical street width. This automaticity simplifies the visualization task and is friendly to using our approach in a fully Web-enabled viewing program.

6 Results

We have applied our approach to the exploration and visualization of a 30-year simulation of Seattle and the surrounding four-county area (Puget Sound). The geographical area contains more than 16,300 km^2, 1.4 million parcels, and 3,872 aerial images of $5{,}000 \times 5{,}000$ pixels of spatial resolution at 0.45 m/pixel. Results of our implementation prototype are shown in this section and in the accompanying video, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2008.193.

Fig. 7 shows a close-up of generated geometry (i.e., parcels and streets) for an exemplary area in our data set. Fig. 7a contains the original geometry as per GIS data. Fig. 7b has the region subdivided into the desired number of parcels, as per the simulation, but without enforcing egress. Fig. 7c contains the same area but where the enforcement of egress caused a slightly different arrangement of parcels and streets. It is worth noting that egress is not always guaranteed in the input GIS data, nor is the GIS data free of duplicated points and parcel polygons with non-self-intersecting boundaries. This causes some visual artifacts such as the thin and sharp parcels. Thus, we can produce egress for most (generated) areas but not necessarily for all existing areas.

Fig. 8 contains urban visualization examples demonstrating both traditional choroplethic maps and simulation interpretations that use our new automatically generated content. Figs. 8a and 8b contain choroplethic views using the Indicator System [30] and using a rainbow color map generated by our system. The rainbow color map depicts the changes in household values obtained via the simulation and displayed on top of the original layout; this value serves as input to our content generation process. The color of each parcel corresponds to the change in the number of

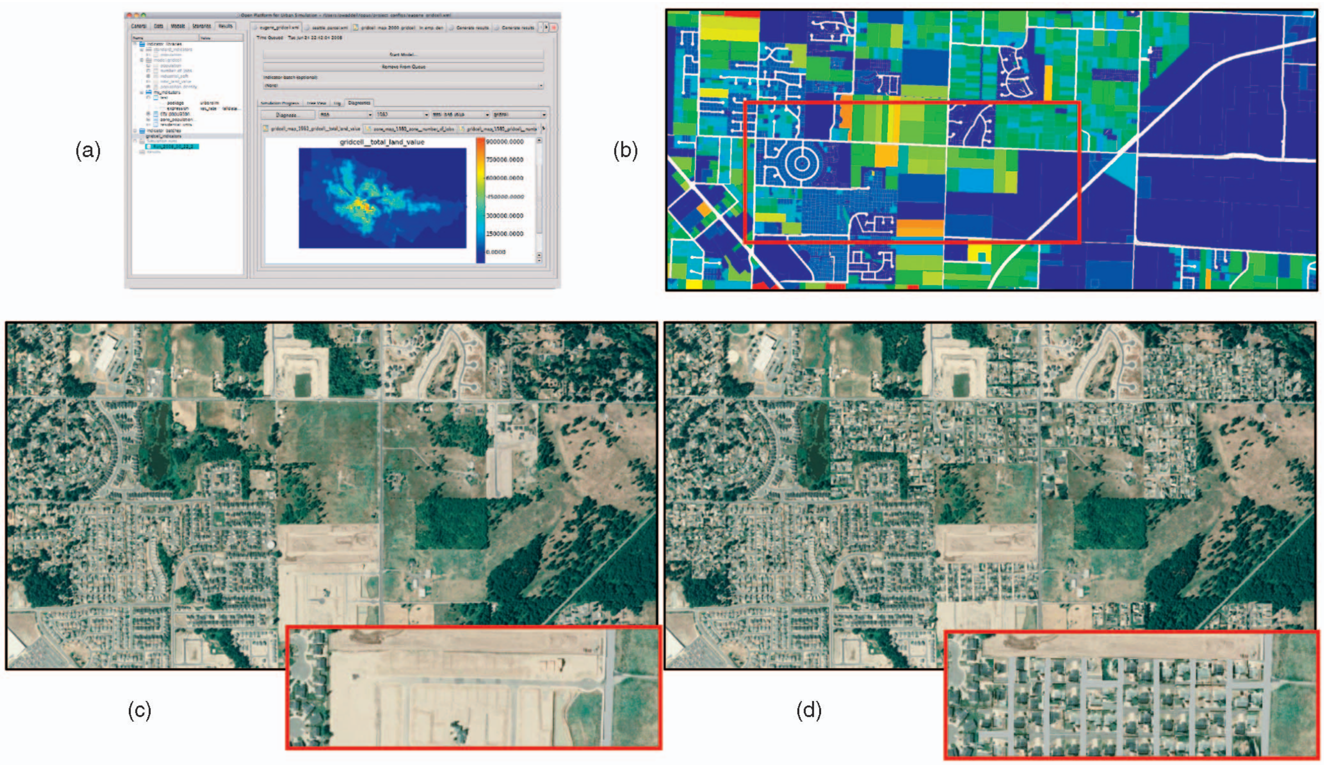


Fig. 8. **Visualizing urban simulation data.** We show exemplary choroplethic visualizations of urban simulation data and views using our proposed method. (a) A screen snapshot from the Indicator system supported by UrbanSim. (b) A rainbow color map used to represent changes in households/buildings for the initial set of parcels; blue corresponds to no change, and green/yellow/red corresponds to a small/medium/large increase in the number of households. (c) The original layout of an urban area before the simulation. (d) Our generated layout after the simulation. For both (c) and (d), close-ups are also included.

contained households. Figs. 8c and 8d show an aerial image before the simulation period and generated images representing the same area after the simulation.

The visualization of results from our simulation also provides a valuable means to examine results and supplies useful diagnostic capacity to determine where there may be anomalies in the simulation. Fig. 9 shows several specific examples of simulated future changes in an urban area. Fig. 9b shows an overview of an urban area in which considerable residential development has been simulated between 2000 and 2030. Our implementation renders new parcels in cyan and original parcels in yellow. The area (Area No. 1) is on the southern side of the city of Tacoma and reflects the pressures on residential development caused by high housing prices in Seattle and the growing demand for more affordable housing (it is a superset of the area shown in Fig. 8). This area is relatively well served by transportation access, and moderate land prices make residential development attractive. The presence of an urban growth boundary also contributes to the development of pockets such as this, since the boundary is intended to promote more compact development patterns.

Figs. 9c and 9d highlight several areas to the north and east of the city of Seattle that contain significant residential development in the simulation results (Area No. 2). Some of these patterns show infill development (i.e., the construction of housing on small remaining vacant parcels in an area that is mostly developed), for example, the area to the north of Kirkland on the eastern side of Lake Washington. This location is attracting development of new relatively expensive housing in an area that has been developing in a consistent way and is close to shopping and recreational amenities. The newly developed regions are similar to the existing areas and are visually plausible.

However, there are other pockets of development in the simulation results that are not always so plausible, such as the area on the west side of Seattle, overlooking Puget Sound (Figs. 9e and 9f). This is in the city of Shoreline and contains expensive housing on large steep lots (Area No. 3). These parcels are unlikely to be further subdivided on the scale the simulation results suggest. It is more likely that the input data for the simulation does not reflect realistic density constraints in this expensive neighborhood or that the development model algorithm needs to be modified to include a means of distinguishing expensive parcels built at a much lower density than regulations would allow and are unlikely to be subdivided because of neighborhood resistance and the amenity value of the large lot housing. Thus, the figures demonstrate the use of visualization for both explanatory and diagnostic purposes.

Fig. 10 contains an example of an effective redevelopment by taking adjacent parcels and constructing a new infrastructure of parcels and streets on top of them. Figs. 10a and 10b depict a real example of urban redevelopment, obtained from the county urban planning office. Figs. 10c and 10d show a similar phenomena using our data set and our content generation methods. Although the simulation model did not explicitly simulate a consolidation and repartitioning of the parcels, our approach

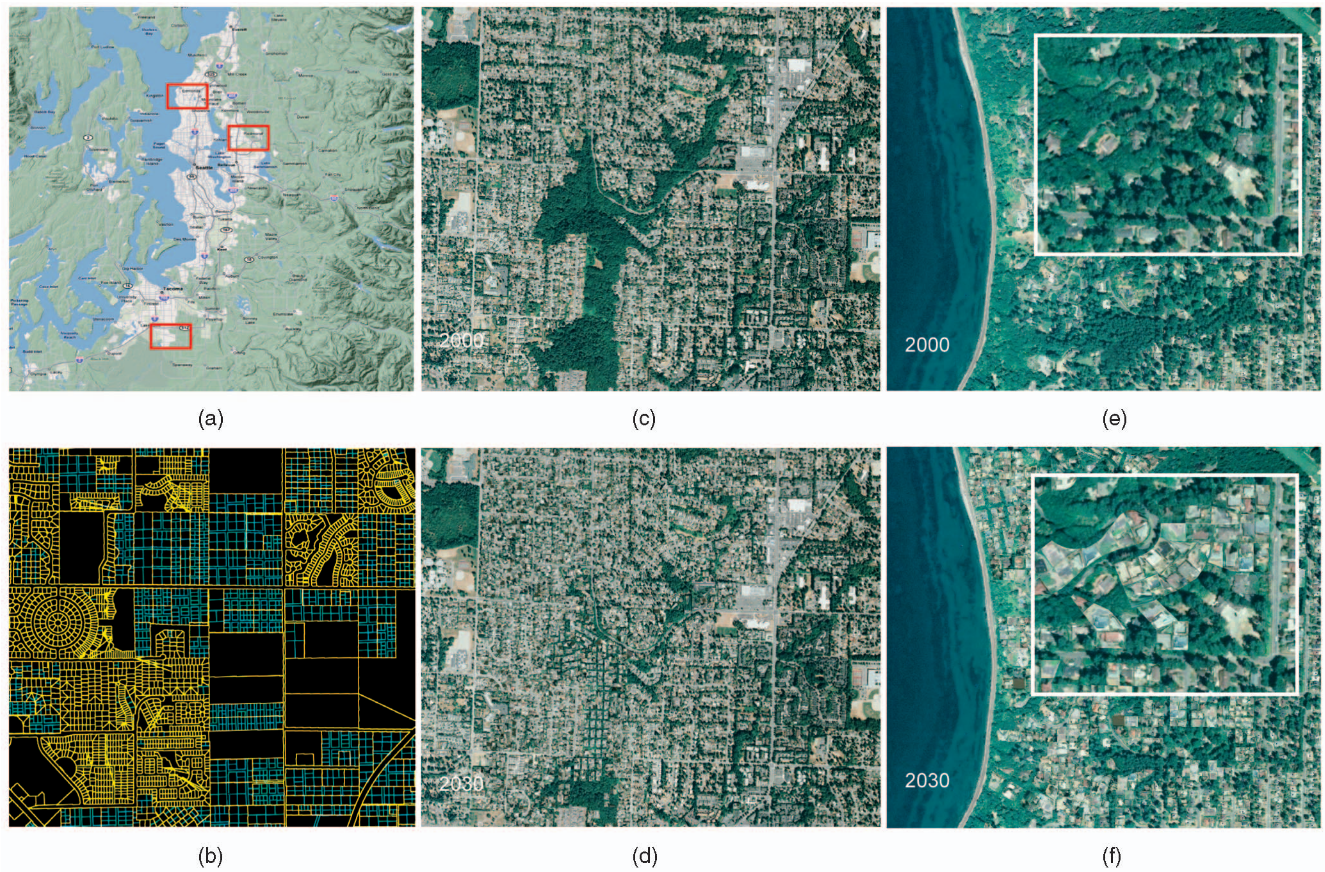


Fig. 9. **Simulation examples.** (a) An overview of the simulated area (borrowed from Google Maps). (b) Rendering of the parcel changes for Area No. 1 (new parcels in cyan). (c) and (d) Before and after view of simulated urban region for Area No. 2. (d) and (e) Before and after view of simulated urban region for Area No. 3.

Fig. 10. **Redevelopment.** (a) An area before a process of redevelopment occurred (year 2001). (b) The area in (a) after adjacent parcels have been merged and are in the process of redevelopment (year 2005). This information was obtained from the county urban planning office. (c) Another significantly larger region before the simulation (year 2001). (d) Simulated imagery and street/parcel data of the area in (c) after several adjacent parcels have been redeveloped and streets have been created (year 2030).

implicitly provides the adjacency information and is able to join adjacent parcels, merge streets, and create a new development replacing another one.

7 Conclusions and Future Work

We have shown a tool for displaying results from urban simulations. Our work builds upon existing visualization techniques of urban simulations and extends them by automatically inferring new urban layouts. To perform the simulation and without loss of generality, we use the publicly available UrbanSim package. Our inference algorithms gather stochastic data of the original urban layout and use the simulation state values to obtain a plausible urban layout, consisting of new parcels, streets, and imagery (e.g., vector and image data). Altogether, our approach allows for traditional visualizations, as well as that of new content. We have applied our method to visualize a 16,300-km^2 urban area.

We are pursuing several avenues of future work. First, we would like to investigate producing synthetic 3D building content for each of the parcels. In similar spirit to one of our previous works [1], one option is to produce scripts for a procedural modeling language (e.g., [20]). Second, we seek to provide visualization support for cases of zoning change (e.g., residential to commercial or vice versa), as well as larger scale changes such as highway alterations, bridges, waterway changes, and so forth. Third, we are interested in further integrating the visualization methods developed in this article with the simulation model in order to refine the simulation in UrbanSim to directly use the generated parcels and streets and, potentially, the synthesized imagery. For example, updating the geometry of the parcels that have been subdivided would allow the simulation to update density calculations and proximity measures that are key predictors of residential location choices and property values. An additional possibility is to use the content generation process to be able to explicitly suggest parcel aggregation, thus simulating the process of redevelopment in older areas undergoing transformation. In general, having the ability to generate plausible parcels, streets, and imagery for future simulated spaces provides significant new information that until now was not available to urban simulation models and can lead to considerable improvements in urban simulation and visualization.

Acknowledgments

This work was supported by US National Science Foundation (NSF) CCF 0434398 and by a Purdue Research Foundation grant.

References

[1] D. Aliaga, P. Rosen, and D. Bekins, "Style Grammars for Interactive Visualization of Architecture," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 4, pp. 786-797, July/Aug. 2007.
[2] D. Aliaga, B. Beneš, C. Vanegas, and N. Andrysco, "Interactive Reconfiguration of Urban Layouts," *IEEE Computer Graphics and Applications,* vol. 28, no. 3, pp. 38-47, May/June 2008.
[3] S. Alkheder, J. Wang, and J. Shan, "Fuzzy Inference Guided Cellular Automata Urban Growth Modeling Using Multi-Temporal Satellite Images," *Int'l J. Geographical Information Science,* 2007.
[4] M. Batty, P. Steadman, and Y. Xie, *Visualization in Spatial Modeling. Center for Applied Spatial Analysis,* Working Paper 79, Univ. College London, 2004.
[5] M. Batty, "Urban Modeling in Computer-Graphic and Geographic Information System Environments," *Environment and Planning B: Planning and Design,* vol. 19, no. 6, pp. 663-688, 1992.
[6] M. Batty, "Virtual Geography," *Time and Space Geographics Perspectives on the Future,* vol. 29, nos. 4/5, pp. 337-352, 1997.
[7] R. Chang, G. Wessel, R. Kosara, E. Sauda, and W. Ribarsky, "Legible Cities: Focus-Dependent Multi-Resolution Visualization of Urban Relationships," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1169-1175, Nov./Dec. 2007.
[8] G. Chen, G. Esch, P. Wonka, P. Mueller, and E. Zhang, "Interactive Procedural Street Modeling," *ACM Trans. Graphics,* vol. 27, no. 3, 2008.
[9] T. de la Barra, "Integrated Land Use and Transport Modeling: The TRANUS Experience," *Planning Support Systems: Integrating Geographic Information Systems, Models and Visualization Tools,* R.K. Brail and R.E. Klosterman, eds., pp. 129-156, ESRI Press, 2001.
[10] D. Dorling, A. Barford, and M. Newman, "Worldmapper: The World as You've Never Seen It Before," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 757-764, Sept./Oct. 2006.
[11] J. Dykes and C. Brunsdon, "Geographically Weighted Visualization: Interactive Graphics for Scale-Varying Exploratory Analysis," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1161-1168, Nov./Dec. 2007.
[12] C. Frueh and A. Zakhor, "Constructing 3D City Models by Merging Aerial and Ground Views," *IEEE Computer Graphics and Applications,* vol. 23, no. 6, pp. 52-61, Nov./Dec. 2003.
[13] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin, "Image Analogies," *Proc. ACM SIGGRAPH '01,* pp. 327-340, 2001.
[14] D.A. Keim, S.C. North, and C. Panse, "CartoDraw: A Fast Algorithm for Generating Contiguous Cartograms," *IEEE Trans. Visualization and Computer Graphics,* vol. 10, no. 1, pp. 95-110, Jan./Feb. 2004.
[15] R.E. Klosterman, "The What If? Planning Support System," *Planning Support Systems: Integrating Geographic Information Systems, Models and Visualization Tools,* R.K. Brail and R.E. Klosterman, eds., pp. 263-284, ESRI Press, 2001.
[16] J. Landis, "The California Urban Futures Model: A New Generation of Metropolitan Simulation Models," *Environment and Planning B,* vol. 21, pp. 399-420, 1994.
[17] K. Leonard, C. Clarke, and J. Gaydos, "Loose-Coupling a Cellular Automaton Model and GIS: Long-Term Urban Growth Prediction for San Francisco and Washington/Baltimore," *Int'l J. Geographical Information Science,* vol. 12, no. 7, pp. 699-714, 1998.
[18] D. McFadden, "Conditional Logic Analysis of Qualitative Choice Behavior," *Frontiers in Econometrics,* P. Zarembka, ed., Academic Press, 1973.
[19] P. Merrell, "Example-Based Model Synthesis," *Proc. ACM Symp. Interactive 3D Graphics and Games (I3D '07),* pp. 105-112, 2007.
[20] P. Mueller, P. Wonka, S. Haegler, A. Ulmer, and L.V. Gool, "Procedural Modeling of Buildings," *ACM Trans. Graphics,* vol. 25, no. 3, pp. 614-623, 2006.
[21] P. Mueller, G. Zeng, P. Wonka, and L.V. Gool, "Image-Based Procedural Modeling of Façades," *ACM Trans. Graphics,* vol. 26, no. 3, p. 85, 2007.
[22] C. Panse, M. Sips, D. Keim, and S. North, "Visualization of Geo-Spatial Point Sets via Global Shape Transformation and Local Pixel Placement," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 749-756, Sept./Oct. 2006.
[23] Y.I.H. Parish and P. Mueller, "Procedural Modeling of Cities," *Proc. ACM SIGGRAPH '01,* pp. 301-308, 2001.
[24] L.D. Pinnel, M. Dockrey, A.J.B. Brush, and A. Borning, "Design of Visualizations for Urban Modeling," *Proc. Joint Eurographics-IEEE TVCG Symp. Visualization (VisSym),* 2000.
[25] J. Portugali, *Self-Organization and the City.* Springer, 2000.
[26] S.H. Putman and C. Shih-Liang, "The METROPILUS Planning Support System: Urban Models and GIS," *Planning Support Systems: Integrating Geographic Information Systems, Models and Visualization Tools,* R.K. Brail, R.E. Klosterman, eds., pp. 99-128, ESRI Press, 2001.
[27] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants.* Springer, 1996.
[28] W. Ribarsky, T. Wasilewski, and N. Faust, "From Urban Terrain Models to Visible Cities," *IEEE Computer Graphics and Applications,* vol. 22, no. 4, pp. 10-15, July/Aug. 2002.
[29] A. Runions, M. Fuhrer, B. Lane, P. Federl, A.-G. Rollandlagan, and P. Prusinkiewicz, "Modeling and Visualization of Leaf Venation Patterns," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 702-711, 2005.
[30] Y. Schwartzman and A. Borning, "The Indicator Browser: A Web-Based Interface for Visualizing UrbanSim Simulation Results," *Proc. 40th Hawaii Int'l Conf. System Sciences (HICSS),* 2007.
[31] H. Ševčíková, A. Raftery, and P. Waddell, "Assessing Uncertainty in Urban Simulations Using Bayesian Melding," *Transportation Research Part B: Methodology,* vol. 41, no. 6, pp. 652-659, 2007.
[32] G. Stiny, *Pictorial and Formal Aspects of Shape and Shape Grammars.* Birkhauser, 1975.
[33] K. Train, *Discrete Choice Models with Simulation.* Cambridge Univ. Press, 2003.
[34] P. Waddell, G. Ulfarsson, J. Franklin, and J. Lobb, "Incorporating Land Use in Metropolitan Transportation Planning," *Transportation Research Part A: Policy and Practice,* vol. 41, pp. 382-410, 2007.
[35] B. Watson and P. Mueller, "Urban Design and Procedural Modeling," *ACM SIGGRAPH Course #14 Notes,* 2007.
[36] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, "Instant Architecture," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 669-677, 2003.

Carlos A. Vanegas received the BS degree in applied mathematics from EAFIT University, Colombia, in 2007. He is a PhD student in the Department of Computer Science, Purdue University, West Lafayette, Indiana. His research interest is in computer graphics, particularly in urban modeling and visualization, acquisition, and reconstruction.

Daniel G. Aliaga received the PhD degree from the University of North Carolina (UNC). He is an assistant professor of computer the Department of Computer Science, Purdue University, West Lafayette, Indiana. He is a researcher in computer graphics and in particular acquiring, modeling, and rendering large environments. He has authored about 40 papers in his field. He has served on numerous program committees and on several US National Science Foundation (NSF) panels. He has served as a journal editor and as the conference chair.

Bedřich Beneš received the PhD degree from the Czech Technical University. He is an assistant professor in the Department of Computer Graphics Technology, Purdue University, West Lafayette, Indiana. His research interests include procedural modeling, artificial life, real-time rendering, and global illumination. He is a member of ACM SIGGRAPH, Eurographics, and the IEEE Computer Society.

Paul Waddell is a professor in the Evans School of Public Affairs, University of Washington, Seattle. He is the director of the Center for Urban Simulation and Policy Analysis, where he leads the UrbanSim project. His research focuses on land use, transportation and environmental policy analysis, and urban simulation. He has been a PI or co-PI on more than million in federal research grants, primarily from the US National Science Foundation (NSF).