

Virtual City Map Generation using Area Subdivision

Kerstin Müller Luc Heischbourg Christoph Fünfzig Sebastian Petsch Hans Hagen

TU Kaiserslautern, Computer Graphics Group, Germany

Abstract

In this paper we present a new approach to build typical street networks for urban and rural areas automatically. Most street networks have a characteristic appearance which is a mixture between a kind of a regular spreading of streets and various perturbations of the regular structure, e.g., caused by the terrain. Our City Map Generation (CMG) algorithm gets as input an height field including watercourses and restricted areas such as nature protection areas or mountain areas. Furthermore, the user can specify the urbanization by the number of city centers or by a density map. The CMG algorithm produces a street map for the given landscape based on rules, which the user can also modify to obtain an individual appearance. Additionally, the regular structure is varied by a set of rules incorporating randomness. The results can be used to generate models for city areas, e.g., in movies and commercials. Depending on the demands, our flexible approach can be adjusted to produce a variety of road maps automatically. We show some examples and demonstrate that our algorithm is easy to implement and to use.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation, I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Splines, I.6.3 [Computer Graphics]: Simulation and Modeling—Applications

1. Introduction

Urban spaces consist of buildings and neighborhoods connected by a street network. Their modeling is a complex challenge as they populate large areas, spanning from a few to hundreds of square kilometers, with patterns of a varying density. In reality, street networks have grown, and different sections have been changed hundreds of times in the past. But their construction follows practical considerations and a hierarchy given by the importance of streets [Mar05].

Traditionally, the generation of city maps has been a manual modeling process, which requires a lot of resources. For projects of a certain size in commercials or movies, such an effort is invested but tools are also well appreciated. Similarly in simulation games, street maps are required for large spaces and here the creation becomes outrageous even when terrain maps are available. We consider the problem of automatic generation of city layouts that are realistic in appearance for such applications based on a small set of data sources (terrain map, land utilization map, optional street density map). For easy use, the automatic initial layout step using default parameters produces already realistic results.

According to [VAW*09], city layout construction is the

central component in city modeling as the city shape is largely determined by it and street and house modeling follow. We give a short overview of the approaches, which have evolved over time. Parish et al. [PM01] started the work on procedural techniques to generate road networks and city layouts. In their CityEngine project, road networks are generated in a fully automatic way by utilizing L-shape grammars. The grammar grows the road shape out of an initial road trunk similar to plant growth. Designing a shape grammar to achieve a specific road network shape is a difficult task as noted by several authors [SYBG02], [CEW*08]. While procedural solutions originate from the creation of natural phenomena, [dSM06], [KM07] apply similar methods to fill city lots with a user-selected pattern of secondary roads after the primary road network has been created manually. Sun et al. [SYBG02] use similar techniques, but extract primary roads from digital maps and complete the road network based on them by tiling with templates. Chen et al [CEW*08] utilize two-dimensional real, symmetric tensor fields for user specification and editing of the road network structure. The tensor field defines local coordinate systems in each point of the city map, and they can be edited in an image-based way by the user. Their approach is ex-

tremely flexible but actually allows the creation of impossible road networks in reality. Half-automatic or manual editing approaches for road networks are the most common in practice. In [FWB*01], an integrated, graphical editor with a specific user interface for city models consisting of roads and buildings is presented. Aliaga et al. [AVB08] present an interactive approach for synthesizing urban layouts, street network patterns as well as image textures, using example fragments extracted from aerial-view imagery. New layouts are created by operations out of fragments such as join, expand, and blend.

In this paper, we present a road map generation algorithm based on a terrain map, a land utilization map, and an optional density map. Different from previous work, we solve the problem by area subdivision respecting a number of connection points to incorporate into the street network. Our algorithm has two phases. In an initial layout step, the grid or radial/ring road structure is generated for an initial set of city areas. Next, the yet undeveloped areas are further processed by subdivision into city lots using further major roads. A pattern based on a fixed number of parameters can be used for each area to achieve variability.

The major contributions are

- Area subdivision of a polygonal mesh is used to represent and modify the planning area. The mesh structure represents areas as faces together with streets as edges of the mesh data structure. Large-scale patterns are realized by placing the city center locations, and radial and ring roads around the city center areas as well as parallel roads with respect to terrain features. Small-scale details result from the use of procedural patterns originating from planning patterns in reality [Mar05].
- Given an area to subdivide, and a start and an end point of the road on its border, we present a local algorithm to design the road course. For this purpose, we are using cubic B-splines for the road's representation stored with an edge of the polygonal mesh data structure.
- We focus on a small number of parameters for the street direction and the street connections for a fast and easy to use generator.

2. Overview CMG algorithm

Our algorithm gets as input a terrain model in the form of a regularly sampled height field and a terrain texture image. Annotations like watercourses and restricted areas (e.g., nature protection areas) are defined by B-spline curves. Additionally, a user defined parameter list gives the CMG algorithm information about the city centers and the appearance of the cities, that means a specification of how the algorithm should build the streets by subdivision. The user can place city centers on the terrain to known or desired positions by coordinates or by using a density map. Also in an automatic generation mode, the user presets the number of

centers, and the city centers are placed by the system in the non-restricted and water-free areas of the terrain.

In a first step, the algorithm builds streets along watercourses and restricted areas as well as along the border of the processed terrain in parts or completely. In the next step, the algorithm connects the neighbored city centers, makes crossroads at intersecting streets and generate bridges if required by watercourses. Circular roads are built around the city centers in a variable number and shape. After that, we have a planar graph where the vertices are defined by intersections of streets or rivers, the edges are defined by the streets or rivers and the faces are the uncovered area surrounded by streets or rivers respectively. The roads which are established up to this stage are called first major roads.

The second major roads are generated in a further step: For each uncovered and water-free face, a subdivision process starts and insert streets, called second major roads, in the face. To obtain a connection of smaller streets over major roads or rivers, the subdivision process is continued over rivers or major roads at neighbored faces. Also smaller streets, which run into one major road from two neighbored faces, are connected if they are next to each other. The subdivision algorithm generates varying face sizes depending on the density map or the distance to the city center (if no density map is available). In this way, small faces are generated at the city core, and the faces get larger towards outside areas. We obtain varying street densities in dependence of the distance to the city center. In a last step, the algorithm places small streets with dead ends and loops, called minor roads, in the residential and back roads area. Finally in a cleanup process, undesired streets can be deleted. The output of our CMG algorithm is a road map with the desired street density and number of city centers spread out on the given terrain model.

3. Preliminaries

The city map is represented by a polygonal mesh with center lines of streets or center lines of rivers as edges. The intersections of them are vertices. Faces are the areas surrounded by streets or rivers. A face can be restricted, covered by water or free. A restricted face lies in a restricted area like a nature protection area. A face that is covered by water is part of a river, lake, or the sea. For both, the restricted and the water covered faces, a construction of streets or buildings is not allowed. On the contrary, free faces permit the generation of streets and buildings.

For the map representation during the street construction process, we use OpenMesh [BSBK02]. OpenMesh is a generic and efficient data structure for representing and manipulating polygonal meshes [Bot]. Additional information can be stored with the vertices, faces, and edges easily. For example, an edge gets the control points of a B-spline curve to describe the course of a river or a street. An edge can also

have the tag hidden for its street. A hidden street is a street that exists in the planar graph as an edge but is not visualized. A face obtains the tag if it is restricted, water area or free. Furthermore, we need to change the mesh by topological operations, e.g., split face for inserting a street into a free face. These operations are supported by the OpenMesh library. The resulting city map can be transferred to an arbitrary internal or external scene format.

3.1. Terrain Information

Our algorithm requires as input data a terrain model in the form of a regularly sampled height field with a terrain texture image. For the watercourses, restricted areas like nature protection areas, and mountain areas, we need their borders defined by cubic B-spline curves on the terrain height field. Note that each restricted area belongs to exactly one connected land area.

3.2. Parameter List

A user defined parameter list provides the necessary information for the algorithm to build the streets. The algorithm generates the streets by building centers, connecting them and subdividing the land parcels. The user chooses the number of city centers, and the centers are set by the algorithm in the non-restricted and water-free areas of the terrain. Alternatively, the user can set the locations of the centers. The street density of the cities can be defined procedurally and rotationally invariant as a function of the distance to the associated city center (for example, $1/r^2$ with the distance r). Likewise, the algorithm can use a density map to generate the desired street density according to the population density. A density map is a scalar map which evaluates the density for a point. For each city the type of its center can be chosen by a parameter, e.g., a center with a main junction in the middle or a center enclosed by a ring road. The number and shape of ring roads around a center can also be changed by a parameter. Additionally, the distance of the ring roads to the center can be modified. The user can influence the subdivision process for the street generation by giving probabilities for the different rules: e.g., the probability for subdivision by longest street, subdivide orthogonal to nearest city center, subdivide in direction to nearest city center, choose random street for subdivision. Furthermore, the user can implement new rules and set their probabilities. The streets are defined by B-spline curves. The number of control points per kilometer for major and minor roads can also be specified by user as well as some random perturbation and the angle range for a new street junction. Neighbored junctions can be merged if they are close to each other using a user defined distance parameter.

4. Constructing a street

A street in our approach has always a start point and an end point. The course of a street is defined by a cubic B-spline

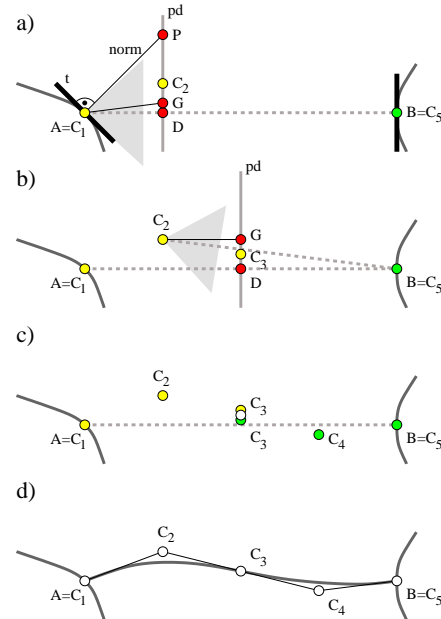


Figure 1: Construction of a street.

curve. The number of control points per kilometer for major and minor roads is a user-defined parameter. As default, highways are smoother and have less bends than small roads in a residential area. A new street can only be inserted in a free face, where its start point A and end point B must lie inside the free face. For constructing a new street s we do the following steps. In dependence of the type of street, the distance d from A to B , and a user defined parameter which defines the number of control points per kilometer for each street type, we get the number of control points n for the B-spline curve that defines the street. Start point A is the first control point C_1 and end point B is the last one, called C_n . To construct the control points $C_2, \dots, C_{\lfloor n/2 \rfloor}$ we proceed from start point A . Similarly, we proceed from end point B for the control points $C_{n-1}, \dots, C_{\lceil n/2 \rceil}$. In case n is odd, control point $C_{(n-1)/2+1}$ is built from both ends and we average the two resulting points. To get control point C_2 , three additional points P, D, G must be computed which provide the desired point C_2 by averaging. Point D lies on $(n-i)/(n-1) \cdot C_{i-1} + (i-1)/(n-1) \cdot B$ with $i=2$ for control point C_2 . Point G and P are on the plane perpendicular to $\overline{C_{i-1}B}$ crossing D , called pd . The intersection between the plane pd and the direction of the smallest inclination of the terrain in the cone (marked in gray in Figure 1) to end point B is the point G . A larger cone angle allows a greater influence of the terrain on the street geometry. If start point A is part of an existing street then let t be the tangent of A and let $norm$ be the normal at A lying in the terrain. Point P is obtained by intersecting $norm$ with plane pd . The average of D, G , and P only in the case start point A is part of

an existing street, provides the new control point C_2 . The remaining control points $C_3, \dots, C_{\lfloor n/2 \rfloor}$, and $C_{n-1/2+1}$, n odd, are computed in a similar way without the additional point P . Starting from the end point B going in direction of start point A , we do the same in reverse orientation to get the control points $C_{n-1}, \dots, C_{\lceil n/2 \rceil}$, and $C_{n-1/2+1}$, n odd. For the resulting control points C_1, \dots, C_n , we check if the street resides in the interior of the face. If the curve runs outside the face, we move the affected control points towards the center of the face till the curve remains in the interior and does not cross another street.

5. CMG Algorithm

We use a polygonal mesh to store the planar graph containing all information for the street network. In every step of the algorithm we insert new edges to the graph and compute the street geometry which is stored with the new edge.

5.1. First Major Roads

We generate the first major roads in three steps as depicted in Figure 2. The algorithm starts with a planar graph consisting of a start face with four vertices and four edges, which stands for the processed terrain and its four surrounding hidden streets. In the first step, the additional terrain information must be integrated into the planar graph. Watercourses are added by inserting edges into the start face (blue lines in Figure 2, left). Edges following the right and left side of the rivers or landside of the coastlines are included. The control points for the streets following the right and left side of the rivers or landside of the coastlines are computed via the control points of the center line of the river. Because not all of these streets are necessary to construct a useful street map, however, their edges are necessary for a consistent data structure, we can set some of these streets as hidden (marked in gray in Figure 2, left). The decision hidden or not is done in dependence of the desired street density given by the density map or distance to the center if no density map is available.

As further terrain information restricted areas like nature protection areas, and mountain areas, need to be added next. If the restricted area lies completely in a face, some additional edges must be inserted to connect the restricted area with its surrounding face (see the edges marked in red in Fig. 2, left). We achieve this by simply adding edges to the north, west, south or east side randomly. The streets around the restricted areas can be calculated via the control points of its border curve and are assigned to the associated edges of the face for the restricted area. For the edges connecting the restricted area with the remaining street network, a new curve must be generated. The start point lies on the street around the restricted area and the end point lies on the street curve of an edge of the surrounding face.

In the second step, the city centers and its connecting

roads must be set. Our approach offers two possibilities for setting the city centers. In a manual mode, the user can mark the positions for the city centers on the terrain by coordinates or provide a density map with city centers. In an automatic generation mode, the locations for the city centers are chosen randomly. Their number is a user defined parameter. After choosing city center coordinates, automatically or manually, each city center is added to the free face to which its coordinates belong, called its associated face. If we have more than one city center per associated face, the associated face is split so that we obtain for each center an associated face. To integrate a city center into the planar graph, a vertex or a face is inserted in its associated face. Whether a vertex or a face is chosen as center depends on which type of city is desired according to the parameter settings.

After that, neighbored city centers need to be connected in the planar graph. Two city centers are neighbored, if they are in two neighbored faces which are connected by a shared edge or connected by two water-covered faces symbolizing a river (see Fig. 2, middle). We do the following for each city center with its associated face: If an edge of the associated face is connected with a face having a city center, directly or via two water-covered river faces, then we split this edge. A new edge is inserted into the associated face between the split point on the edge and the city center vertex or face respectively. If two neighbored city centers are connected via two water-covered faces which represents a river, we must generate a bridge. In this case the two water-covered faces are split as illustrated in Fig. 2, middle, by two new edges symbolizing the bridge. The connection roads are constructed with a city center as start point and a randomly chosen point on the road of the shared edge or river street as end point.

In the third step, the ring roads are generated. Urban road networks often contain fractional or full ring roads. So, in dependence of the desired type of city, we must generate the necessary edges for the ring roads in the planar graph. The incoming edges to the city center (face or vertex) are split and edges are inserted between these points in the faces next to the city centers (see Fig. 2, right). The number of rings and the completeness of these rings can be controlled by parameter or randomly. The curves for the ring roads are constructed via start- and end points on the incoming street curves of the city centers using the method described in Section 4.

5.2. Second Major Roads

At this point, we have all first major roads and continue with a subdivision process to get the second major roads. For each free face F of the polygonal mesh, we start a subdivision process: Firstly, we check if a subdivision of Face F is necessary according to the density map. If no density map is available, we test if the area of F is bigger than a user defined parameter A permits or if any vertex of F is near enough to the city center. So, if a further subdivision is indicated or an edge of

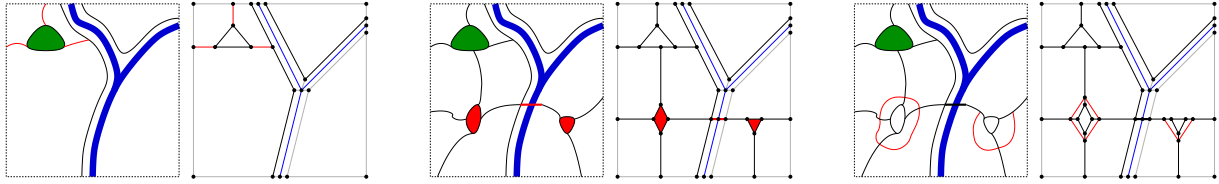


Figure 2: Street network (left) and planar graph (right) in each image group respectively. Edges with hidden streets are marked in gray. Left: Terrain with a river and streets along the river as well as a restricted area (green). Middle: Two city centers are included (marked in red). Right: Ring roads are integrated around the city centers (marked in red).

F is part of a first major street and the center of F is near enough to the center we start subdivision. The predicate *near enough* is given by user defined parameters. The splitting of face F can be done in various ways controlled by user parameters, e.g., the split is done in direction to the nearest city center, orthogonal to it, aligned with the points of the compass, or divided at the longest edges, or completely randomly. For all the new faces obtained from this splitting, a recursive call is done. To avoid dead ends and to continue streets over rivers, we proceed the subdivision on its neighbored face if a hidden street or, e.g., a river is reached. To compute the streets for the edges provided by the subdivision process, more parameters must be considered. The start and end point is defined by the criteria of the subdivision method, e.g., where the intersection is placed on the street of the start- and end point. Again, we use the construction method of Section 4 with an additional noise parameter for second major streets to perturb their regular structure.

5.3. Minor Roads

To get the minor streets, each face obtains the additional information about its minor street pattern. The minor streets are not integrated in the planar graph, we store only the type of the minor street pattern per face. The pattern of the minor street is drawn into its face by scaling and rotating the pattern so that it fits completely into its face and does not cross or contact the surrounding streets of its face. The spread out of minor streets is controlled by user defined parameters depending on the size of the face and its distance to the city center. Finally, edges of streets with less traffic can be set as hidden if the user wants to thin out the planar graph or if, e.g., only a partial urbanization of the processed terrain is desired. After this process, we have the planar graph stored in a polygonal control mesh of a fully street network from a given terrain. Also, we have the geometric routes of our city map stored with them.

6. Results

In this section, we show some significant examples of road networks generated by our algorithm. Naturally given areas like mountains are surrounded by streets, and rivers are fractional seamed by shore-line streets as shown in Figure 3, top

left. The urbanization of an area can be configured by the distribution of city centers, as they determine the area subdivision. A small number of city centers generates the typical European city grown from a small number of suburbs, which have merged into a single metropolitan area (Figure 3, top right). Two types of city centers are shown in Figure 3, bottom. In the bottom right picture, the highway merges into a ring road, which encircles the city center. In the bottom left picture, the highways meet in one city center point. From the parameter settings, the density of the major and minor streets in the ring remains the same. This changes at the exterior of the ring road. Here, street density decreases with larger distance from the ring and is only retained along the main streets. The fragmentation and the bending of these major roads can be varied. The remaining free faces are filled with a local network pattern.

7. Conclusion

We provide a flexible approach for the automatic generation of realistic street networks for urban and rural areas. Our City Map Generation (CMG) algorithm is able to generate street networks on a given terrain with existing watercourses and restricted areas such as nature protection areas or mountain areas. The approach is based on area subdivision guided by city center points distributed over the free area and radial and ring roads connecting them. Their generation is influenced by user parameters. The areas created by the first major roads are processed by subdivision resulting in second major roads. Finally, depending on the parameter settings, a variety of minor road patterns can be used to fill desired areas. In terms of data structure, we use a standard polygonal mesh for storing the planar subdivision of the planning area. Streets are represented by the edges of this data structure. For the geometric instantiation of streets, we employ the cubic B-spline form: It allows to generate smooth roads connecting a start to an end point. An additional crossroad can be added to the street by generating a triple knot for an interior control point. Our routing algorithm for streets is simple and local, and it is steered by a direction cone towards the curve's end point. The cone angle as a parameter controls jointly the curvature and the allowed inclination of

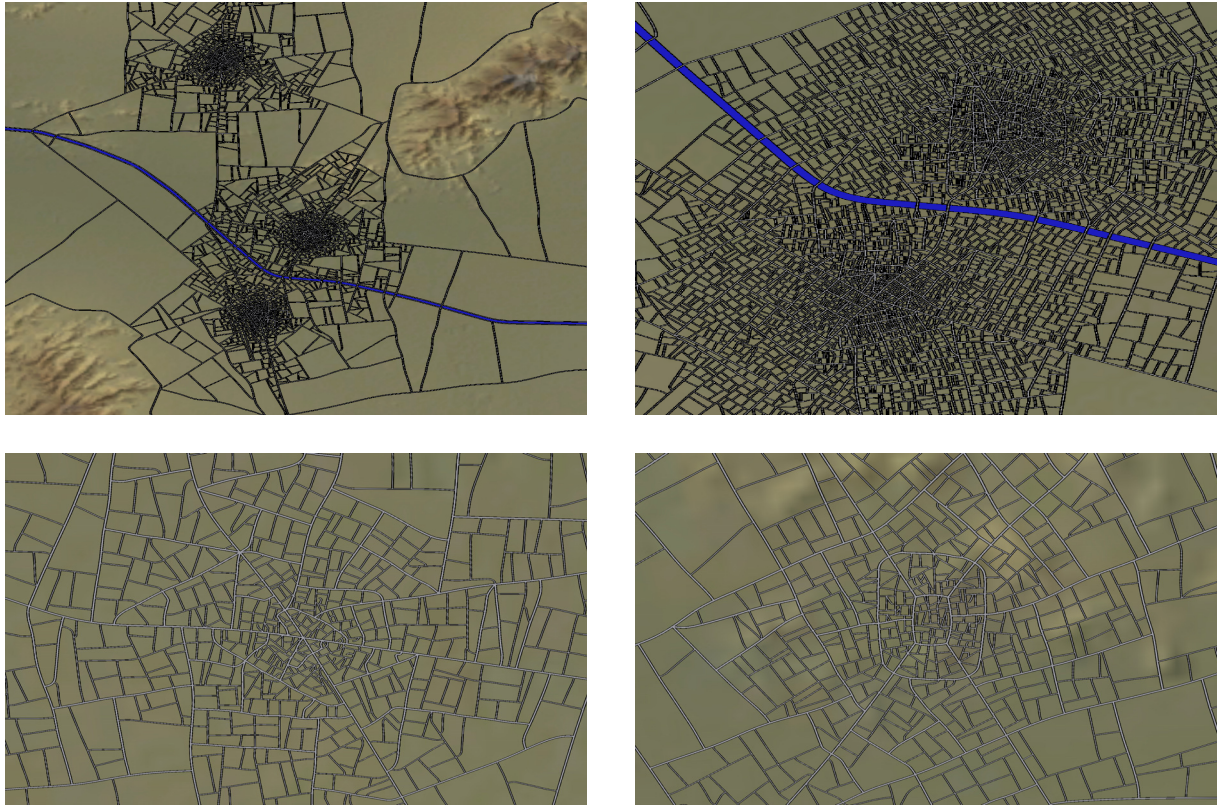


Figure 3: Top Left: Street networks with three city centers and medium street density. Mountain areas and river borders are integrated into the road networks of the metropolitan areas. Top Right: One city grown from two suburbs, which have merged into one single metropolitan area. The two city cores are separated by a river and interconnected by several bridges. Bottom Left: Highways end in one city center point. Ring roads with a higher noise parameter value encircle the center. Bottom Right: The highways merge into a ring road, which encircles the city.

the street curve. The generated results are suitable for several applications like simulation games, movies or commercials.

References

- [AVB08] ALIAGA D. G., VANEGAS C. A., BENEŠ B.: Interactive example-based urban layout synthesis. In *ACM SIGGRAPH Asia 2008 papers* (New York, NY, USA, 2008), SIGGRAPH Asia '08, ACM, pp. 160:1–160:10. [2](#)
- [Bot] BOTSCH M.: <http://www.openmesh.org>. OpenMesh Documentation. [2](#)
- [BSBK02] BOTSCH M., STEINBERG S., BISCHOFF S., KOBELT L.: OpenMesh - a Generic and Efficient Polygon Mesh Data Structure. In *Proceedings of OpenSG Symposium 2002* (2002). [2](#)
- [CEW*08] CHEN G., ESCH G., WONKA P., MÜLLER P., ZHANG E.: Interactive Procedural Street Modeling. *ACM Transaction on Graphics* 27, 3 (2008), 103:1–103:10. [1](#)
- [dSM06] DA SILVEIRA L. G., MUSSE S. R.: Real-time generation of populated virtual cities. In *Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, 2006), VRST '06, ACM, pp. 155–164. [1](#)
- [FWB*01] FLACK P. A., WILLMOTT J., BROWNE S. P., ARNOLD D. B., DAY A. M.: Scene Assembly for large Scale Urban Reconstructions. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage* (New York, NY, USA, 2001), ACM, pp. 227–234. [2](#)
- [KM07] KELLY G., MCCABE H.: Citygen: An Interactive System for Procedural City Generation. In *Fifth International Conference on Game Design and Technology* (2007), pp. 8–16. [1](#)
- [Mar05] MARSHALL S.: *Streets & Patterns*. Spon, 2005. [1](#), [2](#)
- [PM01] PARISH Y. I. H., MÜLLER P.: Procedural Modeling of Cities. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 301–308. [1](#)
- [SYBG02] SUN J., YU X., BACIU G., GREEN M.: Template-Based Generation of Road Networks for Virtual City Modeling. In *VRST '02: Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, 2002), ACM, pp. 33–40. [1](#)
- [VAW*09] VANEGAS C. A., ALIAGA D. G., WONKA P., MÜLLER P., WADDELL P., WATSON B.: Modeling the Appearance and Behavior of Urban Spaces. Pauly M., Greiner G., (Eds.), Eurographics Association, pp. 1–16. [1](#)