Internal Version (Early Draft)

# Illustrative Visualization of 3D City Models

Jürgen Döllner, Henrik Buchholz, Marc Nienhaus, Florian Kirsch

University of Potsdam, Hasso-Plattner-Institute, Helmert-Straße 2-3, Potsdam, Germany

## ABSTRACT

This paper presents an illustrative visualization technique that provides expressive representations of large-scale 3D city models, inspired by the tradition of artistic and cartographic visualizations typically found in bird's-eye view and panoramic maps. We define a collection of city model components and a real-time multi-pass rendering algorithm that achieves comprehensible, abstract 3D city model depictions based on edge enhancement, color-based and shadow-based depth cues, and procedural facade texturing. Illustrative visualization provides an effective visual interface to urban spatial information and associated thematic information complementing visual interfaces based on the Virtual Reality paradigm, offering a huge potential for graphics design. Primary application areas include city and landscape planning, cartoon worlds in computer games, and tourist information systems.

**Keywords:** Geovisualization, Virtual Geo-Environments, Urban Planning, 3D City Models

## 1. INTRODUCTION

Models of real or imagined urban areas represent complex spatial data such as 3D building geometry, street networks, and vegetation as well as integrates a broad range of related thematic information. The depiction of these models involves manifold technical and artistic challenges. In this paper, we describe an illustrative real-time visualization technique for 3D city models that provides expressive representations of large-scale 3D city models.

Classical examples of 3D city model visualizations include both bird's-eye view and panoramic maps. In general they are not only useful tools for effectively encoding and communicating urban spatial information but also turn out to be works of art, serving as valuable visual guides and visual indices to spatial structures and thematic information related to a city and its surrounding area. The abstracted, stylized presentation emphasizes components of 3D city models and thereby eases recognition, facilitates navigation, exploration, and analysis of urban spatial information. Figure 1a shows an example of an illustrative 3D city model created by the presented technique.

### 1.1 Historic Remarks on City Depictions

The roots of historic *3D city depictions* date back over more than a thousand years. In pre-medieval times, depictions primarily revealed symbolic-allegoric contents, partially composed of landmark buildings but without adhering to topological and geometrical properties and relationships. Instead, topology and geometry were deduced from philosophical, mythical, or religious concepts. In medieval times, topology and geometry of urban areas were increasingly transferred to depictions, starting with orthogonal-like projections and being advanced by perspective drawings starting in the 16th century. Matthäus Merian (1593-1650) is among the most prominent "city modelers": He established the first systematic production of city depictions as commercial products – manufacturing more than 2,150 European city views in his "Topographia", a book series with more than 30 volumes (Fig. 1b).

**Figure 1.** (a) Illustrative visualization of a 3D city model. (b) Historic city depiction of a small village by Merian. (c) Example of a hand-drawn bird's-eye view map illustrating the city of Rhede, Germany.

Since that time, continuous advances in geodesy have been improving accuracy and completeness of the underlying data, while the general principles of city model visualization remained. Bird's-eye view maps are common for tourist information and description of local areas (Fig. 1c). *Physical 3D city models*, typically made of wood, represent common tools in city planning. In the 20th century, these models appeared for major cities, frequently abstracting from the photorealistic appearance of buildings while favoring representation of plan status or city development processes. As an important property, these models can be incrementally renewed.

### 1.2 Potentials of City Models

Throughout all historic variants, we can identify the ability of city models to encode and visualize complex spatial information of an urban area in a compact, abstracted way as their core potential. In particular, city models allow for representing past, current, and future states or thematic information about economy, society, transports, culture, and politics using geometry of the 3D city models as a basis of visualization. In this way, non-virtual city models represent valuable resources, sometimes using complex symbolisms for compact information display. For instance, Merian used to visualize the branches of trade of a city by exemplary objects placed in streets. He also positioned objects of known size (e.g., humans, animals) to give scale hints to the viewer.

*Virtual 3D city models* expand this potential towards interactive visual interfaces that allow for selecting, exploring, analyzing, and editing information. Therefore they represent an important *visualization paradigm* for urban spatial information and associated thematic information.

### 1.3 Nonphotorealistic 3D City Models

In our approach, we aim at interactive illustrative visualizations of 3D city models in a way that is complementary to Virtual Reality visualizations. The approach can be characterized as follows:

- Concentrate on illustrative, expressive visualizations emphasizing on high perceptual and cognitive quality to effectively communicate contents, structure, and relationships of urban objects as well as related thematic information.
- Enable meaningful visualizations even for the case of scarce urban spatial information since high-quality and complete data is rarely available for large-scale urban areas.
- Enable fully automated generation of visualizations while offering great flexibility in the amount of control of graphics design.
- Achieve real-time rendering and thereby allow for interactive manipulation, exploration, analysis, and editing of 3D city models.

Applications of illustrative visualizations of 3D city models are primarily all kinds of visual interfaces to urban spatial information required, for instance, in architectural drawings and sketches, city development planning and city information systems, radio and energy network planning, visualization of demographic development data, interactive gaming environments and comic worlds, and atmospheric and edutainment environments for narratives.

## 2. RELATED WORK

**Urban Modeling.** The underlying data of urban modeling comes from various different acquisition techniques which cluster into methods based on photogrammetry, active sensors, and hybrid sensor systems[1]. Identifying, categorizing, reconstructing building geometry by automated methods represents a major research challenge[2] and aims at acquiring 3D building data at low costs and high quality. An important additional data source represents administrative records such as the cadastre due to its legal status but mostly they do not explicitly contain 3D information. For the presented visualization technique, we assume that at least 2D ground plans and building height data are accessible. A complementary, challenging source of city data describes Ref. 3: they develop a procedural modeling technique for generating city data. For our technique, such data can be processed in the same way as typical urban data.

Building geometry as most important category of data can be generally classified into five quality levels[4]: *Level 0:* 2D ground plans with included landmark buildings; *Level 1:* Cubature objects, typically derived by extruding 2D ground plans to an average or known height; *Level 2:* 3D objects, with an approximated outer geometry including roofs, terraces, chimneys, etc.; *Level 3:* Architectural 3D models including detailed façade designs and exact geometry; *Level 4:* Architectural 3D models as in LOD-3 with additional interior design. Our technique is focusing on processing and generating buildings of level 1 and 2. In addition, it integrates buildings of level 3.

**Virtual Reality.** A large collection of algorithms and data structures provides efficiently rendering of large complex 3D terrain data (e.g. Ref. 5) and heterogeneous geo-spatial objects[6]. These methods are focusing on efficient realistic rendering (e.g., Ref. 7, 8, 9) and rely on exact texture and geometry data (e.g., Ref. 10). Numerous optimizations for virtual environments have been developed[11] such as discrete and continuous multi-resolution representations, view-frustum culling, occlusion culling, imposter techniques, and scene-graph optimizations. Looking at achieved 3D city visualizations, inherent limitations of VR techniques become apparent. 1) To achieve high quality visualizations, high quality data is required; 2) photorealistic views do not appear to be optimal for many tasks, e.g., exploring and analyzing structure and relation of urban objects or visualizing associated thematic data. Our technique aims at both limitations.

**Nonphotorealistic Rendering**. Strothotte and Schlechtweg give a broad introduction to nonphotorealistic graphics[12]. Increasingly, real-time nonphotorealistic rendering algorithms become available[13]. Ref. 14 gives an overview of algorithms for outlines and silhouettes. Edge enhancement plays a central role in our approach; we apply the algorithm described in Ref. 15. Cartoon-like appearance is achieved by specialized illumination models[16] and color schemes[17].

**Cartography.** First systematic development of panoramic maps and bird's-eye view maps had been done by Matthäus Merian. In his tradition, many cartographers further developed these techniques, for instance, cartographic relief depiction and panorama maps[18], and 3D city maps[19]. The work of these cartographers shows that design and production of those maps does not only involve technical challenges but also requires an artist's experience. We have analyzed and transferred cartographic methods such as a reduced color scheme and outlining.

## 3. COMPONENTS OF CITY MODELS

This section introduces general components we identified for constructing 3D city models. The collection distinguishes three categories of components: buildings, environmental space, and thematic information.

### 3.1 Buildings

Buildings are major visual objects of 3D city models. In general, digital data of buildings can be acquired based on administrative data (e.g., cadastre records), laser scanning, and aerial photography.

**Building Geometry.** To construct block models one can extrude 2D ground polygons to heights derived from laser scans. In practice, for large areas of 3D city models no explicitly modeled building geometry is available. For this reason, the visualization technique should allow for generating most buildings' geometry in an automated way. In addition, the visualization technique should support 3D CAD models of buildings. For instance, these models typically exist for landmark buildings. In practice, only a small percentage of buildings are specified this way but they possess a high visual importance and need to be seamlessly integrated into visualizations.

**Building Roofs.** Roof modeling represents a specialized stage of the construction process, which generates generalized roof geometry based on a coarse classification whereby the best fitting roof model is assigned to an individual building. In practice, roofs as essential visual elements provide visual structure for large numbers of aggregated buildings. Illustrative visualization precisely shows this non-exact roof information using generalized roof geometry without suggesting details that do not exist in the data.

**Building Appearance.** In practice, for large urban areas no detailed data about building appearance can be assumed. Although laser scanning and photogrammetry[1] allow for capturing facade textures, time and cost overheads are still high. Procedural approaches for grammar-based building designs[20] offer an interesting alternative. In our approach, we want to be able to cope with few and less precise appearance data. As elementary graphics attributes, color, material, and facade classification should be available, which generally can be derived from most data sources and administrative data records.

**Terrain Model.** The terrain model is used as reference surface for all graphics objects contained in a 3D city model. The building generation has to adapt 2D ground polygons to the heights defined in the terrain model because polygon data is frequently specified truly two-dimensional.

### 3.2 Environmental Space

The *environmental space* summarizes all spatial objects except buildings and terrain model:

- Transportation networks (roads, rails, …)

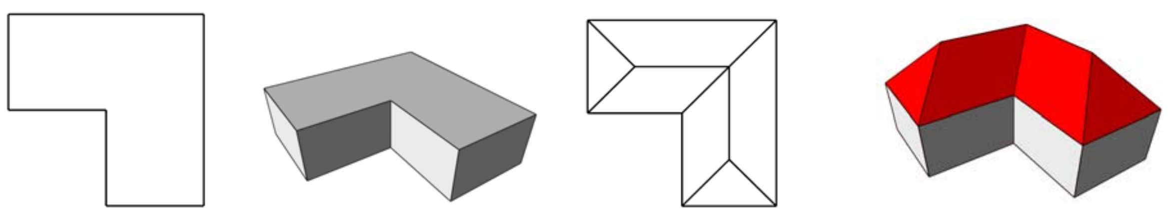

**Figure 2.** (a) Ground plan. (b) Extruded geometry. (c) Roof skeleton. (d) 3D building geometry.

- City furniture (street lights, advertising boards, …)
- Vegetation objects (trees, lawns, …)
- Population objects (people, cars, …)

Vegetation objects are not explicitly addressed in this paper. The illustrative visualization technique handles street networks as part of the ground space model, and represents city furniture as additional scene geometry (Fig. 1a).

## 3.3 Thematic Information

*Thematic information* associated with components are defined and required by applications of 3D city models. They include:

- Building information such as occupancy, industrial/residential usage, year of construction, state of restoration etc.
- Environmental information such as air pollution, noise exposure etc.
- Demographic information such as family, age, and ethnical structures

The visualization technique should be able to visualize thematic information by explicit representations such as 2D labels as well as by graphics attributes of components. Illustrative city models support mapping thematic information on appearance parameters such as edge styles, roof colors, façade structure and composition.

# 4. SPECIFICATION OF BUILDINGS

This section explains how building geometry is constructed and assembled. The process of generating 3D building geometry is illustrated in Fig. 2.

## 4.1 Building Specification

We distinguish between *regular buildings*, generated based on 2D ground polygons, and *landmark buildings*, explicitly modeled as 3D objects. To cope with large-scale 3D city models we need a compact specification of regular buildings and an efficient generation of building geometry.

**Building Blocks.** A regular building consists of one or more building blocks. A building block is specified by a 2D ground polygon, which can be concave, can have inner loops, but must not have self-intersections. It is assumed to lie on top of the terrain surface. For each polygon, a general height is specified. Complex buildings can be represented by building blocks with varying heights. We opt for building blocks as elementary unit because they can be optimally treated as rendering units.

**Thematic Information.** For each building, we specify associated thematic information by a set of key-value pairs. Examples of keys include number of floors, floor heights, roof type, building usage (industrial or residential building), year of construction, and state of restoration.

**Building Roofs.** The automated roof modeling is based on the straight-skeleton approach[21]. The straight skeleton is a set of connected edges contained inside a polygon, thereby partitioning the polygon into sub polygons. The straight skeleton provides a base structure used to construct concrete roof geometry as described in Ref. 22. For this purpose, thematic information about the roof type and roof height of a building block is required. As roof type, we support hip roofs, gable roof, mansard roof, gambrel, and Dutch hip roof, but the roof modeling can be extended if more roof types have to be distinguished. For large-scale city models exact roof geometry is rarely available due their computational costs and the lack of fully automatic reconstruction procedures but classifications can be obtained effectively, for example, taking into account the style of architecture of a district, year of construction, or using computational geometry methods to estimate roof types.

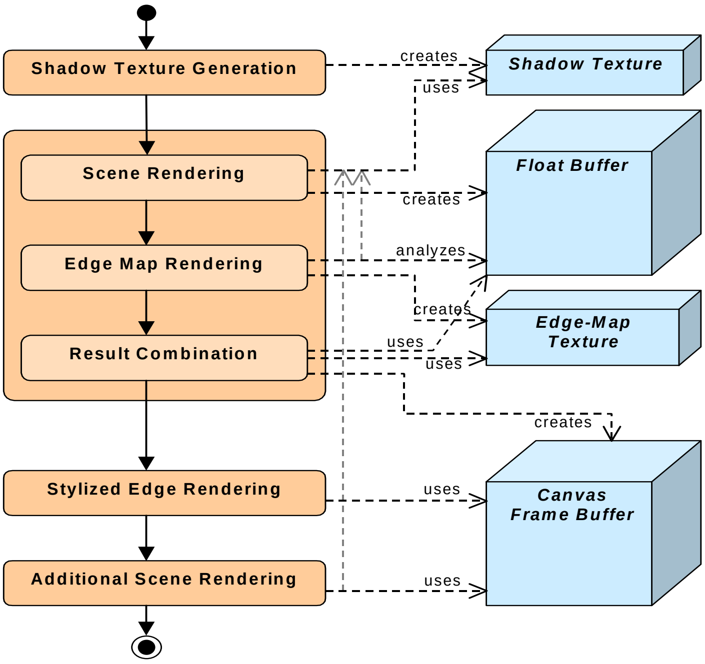

**Figure 3.** Rendering phases and graphics resources.

**Building Appearance.** In illustrative visualizations of 3D city models we cannot assume to have photorealistic facade textures, at least not for large parts of an urban area. Nevertheless we would like to visually communicate significant appearance elements of buildings and inherent properties of buildings specified by thematic information such as number of floors, planning state, architectural style etc. The resulting appearance should still suggest that the facades are generalized and do not represent the original facades in detail. The appearance of buildings can be controlled by several graphical elements, including facade textures and edge enhancement style. We use a configurable shader to specify facades; the shader defines as input parameter material textures, window textures, and door textures. It assembles the base elements to a facade texture on a per-building basis. The way materials, windows, and doors actually are composed depends on the thematic information bound to the shader. For instance, we can encode door positions defined by their distance from the start vertex along the polygon boundary; an arbitrary number of doors can be specified.

## 4.2 Specification of Ground Space

The *ground space* denotes those parts of the terrain surface not covered by ground plans, and it includes mainly street space and green space. We distinguish and have experienced with two modeling approaches, image-based and geometric ground space representations.

- For image-based modeling of ground space, we can superimpose a 2D texture onto the terrain surface that encodes ground-space objects in the digital image, for instance, taken from aerial images.
- For geometric modeling of ground space, we can use 3D geometry, which represents the objects of the ground space such as sidewalks, lawns, etc. Extruding 2D polygons representing ground-space objects can generate this kind of geometry. The objects are laid upon the terrain surface.

For illustrative visualizations, we found that most common sources of data for image-based modeling, images derived from aerial photography, can hardly be transformed to an expressive or illustrative style – it contains too much photorealistic details. We prefer geometric modeling of ground-space objects because it can be derived from common 2D vector data of street systems and green-space areas. Geometric modeling also seamlessly integrates into the nonphotorealistic rendering process, that is, all rendering effects such as edge enhancement or depth cues are straightforward to apply. Insofar, ground-space objects are technically treated in the same way as buildings. Since the ground space is typically generalized to abstract from reality, the resulting objects can clearly be delineated.

## 5. RENDERING ALGORITHM

The rendering algorithm consists of four rendering phases (Fig. 3): Phase 1 generates a texture encoding shadowed regions in image space; Phase 2 renders the scene with enhanced image-space edges, shaded and textured facades; Phase 3 renders stylized edges; and Phase 4 renders remaining components of the 3D city model.

### 5.1 Shadowing

Shadows in 3D city models are important depth cues and facilitate the perception of spatial coherence through the image. In city planning, shadows also represent a critical property of designs that needs to be visualized. To calculate shadows, we use a real-time implementation[23] of the shadow volume technique[24]. In a pre-processing step, the illustrative visualization technique computes shadow volume geometry for a given 3D building geometry. In Phase 1 of our rendering algorithm, this geometry is used together with the 3D scene geometry to generate the stencil value zero for lit areas and non-zero for shadowed areas. For later use in shaders, the shadow information in the stencil buffer is copied into an alpha texture. To apply the texture in subsequent rendering phases, we create a screen-aligned quad that fits completely into the viewport of the canvas so that the texture coordinates ($s$,$t$) of each fragment produced for the quad correspond to windows coordinates.

For calculating shadows, we consider the sun, a single light source at infinite distance that is located above the horizon. Also, only building geometry casts shadows, i.e., shadow polygons are only determined for building geometry. This allows for optimized construction of shadow polygons: They consist of 1) quads, constructed by extruding roof border edges that face away from the light and 2) triangles extruded from wall edges of which one adjacent wall faces to, and the other away from the light. The polygons are extruded away from the light source and end below the ground plane. This approach does not require shadow polygons to be closed in contrast to shadow volume algorithms that are not specifically designed for city models. Therefore, fill-rate required to render the shadow polygons is lower, i.e., performance of our approach is superior.

### 5.2 Shading of Building Geometry

Cartographic city maps and other hand drawings of cities are using a reduced color scheme for shading. In colored drawings the illustrator usually abstracts from the realistic colors of the urban objects and greatly reduces the number of colors used. In general only two or three colors and for each color only two or three tints are used. The choice of the colors is based on aesthetic and other reasons as well as the actual colors of the city.

In Phase 2, we apply $n$-tone shading for all parts of buildings. First, the angle of the polygon’s normal and the light direction is used to determine the intensity. Then, the intensity indexes a color palette of $n$ tones determining the appropriate tone for shading. For this, each building is associated with a color palette defined by thematic information. We observed that three or four tone shading supplies better results for interactive applications than two tone shading because there is much less contrast when moving through a scene and the light direction is much harder to determine. To support depth cueing, we apply a linear transformation in tri-stimulus color space as described in Ref. 17. The saturation of a color is changed according to the viewer distance: more distant objects are rendered in more de-saturated colors whereas the intensities of colors remain constant. As an additional depth cue, we look up the shadow texture to determine shadowed regions.

### 5.3 Facades of Buildings

To cope with large-scale 3D city models, we do not create individual textures for each building. Instead, we combine texture elements to facades using multi-texturing. In a preprocessing step we create multiple sets of texture coordinates for each building, which are used by a fragment shader to compose the facade texture for each building. The facade composition is performed in 7 steps:

1. The material texture is wrapped around the facade with a uniform scale factor.
2. The window texture is projected as decal onto the textured facade. The texture coordinates are determined so that along a single wall, that is, from one corner of the building to the next, we get a whole number of adjoining windows. Decoupling window texture and material texture allows us to keep the material appearance undistorted.

**Figure 4.** Example of a procedurally constructed facade.

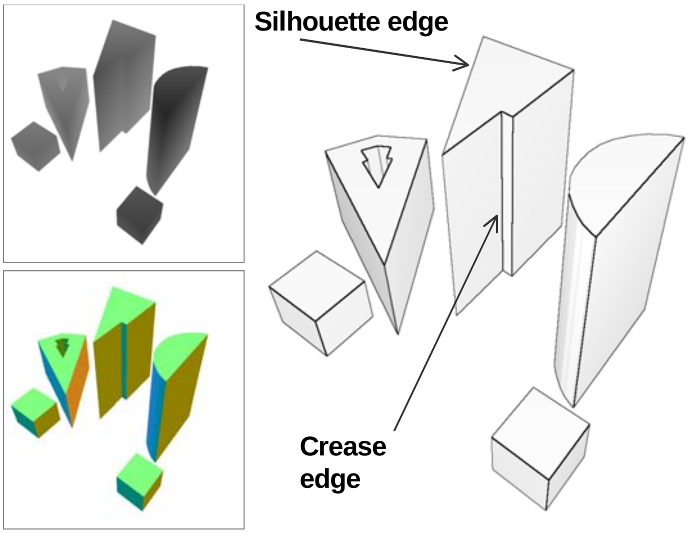

**Figure 5.** Normal buffer, depth buffer, and resulting edge map.

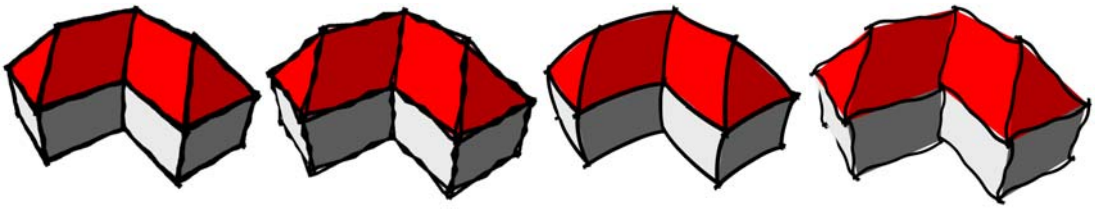

**Figure 6.** Examples of variations of stylized edges.

3. The door texture is projected as decal onto the textured facade. For this, we create texture coordinates along the facade so that the *s* coordinate is in [0,1] where a door shall appear and outside [0,1] elsewhere. Using a door texture with transparent border, we obtain doors at the specified positions.
4. To prevent overlapping of doors and windows, we hide the windows near doors. Similar to step 3), we create additional texture coordinates, for which the *s* coordinate is in [0,1] where the windows shall be hidden.
5. Texturing the whole facade with a material texture leads to an unnaturally regular appearance and adds too much visual complexity to the resulting image. Inspired by hand-made drawings we blend between the facade color and the material texture so that the material is visible only at some irregular formed areas. These areas are defined by an additional alpha texture. In Figure 4, the areas can be clearly seen.
6. Texturing the whole facade with exact replicates of a single window texture leads to unnaturally regular appearance, too. We can avoid this by choosing each singular window randomly from a small set of variations of window textures. For example, we choose from well-lit and low-lit windows. For this, we use an alpha texture filled with random values. The texture coordinates for this texture are chosen so that each pixel is mapped on a single window of a building. The fragment shader uses the value of the alpha texture to choose from different window texture variants. The area of mapped pixels of the alpha texture is chosen randomly for each wall so that the arrangement of the alterations is different for each building. Figure 4 shows an example.
7. To reduce the visual complexity of the resulting image, the shader uses different levels of detail of window texture and door texture dependent on the viewer distance. In contrast to mip-mapping, the use of explicit LOD textures allows us to consciously leave out details instead of compressing too much details into an unrecognizable size by filtering.

## 5.4 Image-Space Edge Enhancement

Image-space edge enhancement, one part of Phase 2, detects visually important edges such as silhouettes, borders, and crease edges. The enhanced edges result in a homogenous and generalized visual depiction of 3D building geometry while emphasizing their principle composition.

We obtain these edges by extracting discontinuities in the normal buffer and the *z*-buffer (Fig. 5). First, encoded normal and *z*-values of 3D scene geometry are rendered into a float buffer. Next, we extract discontinuities by rendering a screen-aligned quad, textured with the float-buffer texture and sampling neighboring texels to identify abrupt changes. The resulting image contains intensity values that represent edges of 3D scene geometry. The image is stored as texture, the *edge map*[15]. Edge maps allow for enhancing edges with varying visual weight since they can be blended into 3D scene geometry. In our examples, we reduce the "edge-ness" in the image with increasing camera distance. The edge-map technique is able to handle 3D scene geometry of any complexity; it is well suited for large-scale 3D data and virtually independent from the number of polygons. So, it is especially well suited for the ground geometry, which generally contains a high number of curved shapes leading to a large number of short edges. Furthermore, it does not require any pre-calculated information about topology and geometry of 3D objects.

## 5.5 Object-Space Edge Stylization

For object-space edge stylization we use the algorithm described in Ref. 25. It generates quads for visually important edges of building geometry, which are textured similar to artistic edge strokes. Their shape and appearance can be individually defined (Fig. 6). This way, they serve as primary tool to assign a characteristic appearance to visualizations and to depict specific thematic information such as planning state or renovation state.

**Principal Edges.** To apply edge stylization to a building, a set of visually important edges must be detected first. They correspond primarily to hard edges of the generated geometry. However, some buildings have many small cavities, which lead to many hard edges that are not characteristic for the building's shape. So, we remove such cavities from the ground polygons in a preprocessing step: For each edge $E_i$ we search in the succeeding edges for a collinear edge $E_{i+k}$, for which all vertices in between, that is the vertices of $E_{i+1} \ldots E_{k-1}$, are within an error-threshold distance from the line supported by $E_i$. If such an edge exists, we connect the first vertex of $E_i$ with the second vertex of $E_{i+k}$, removing all intermediate vertices from the polygon.

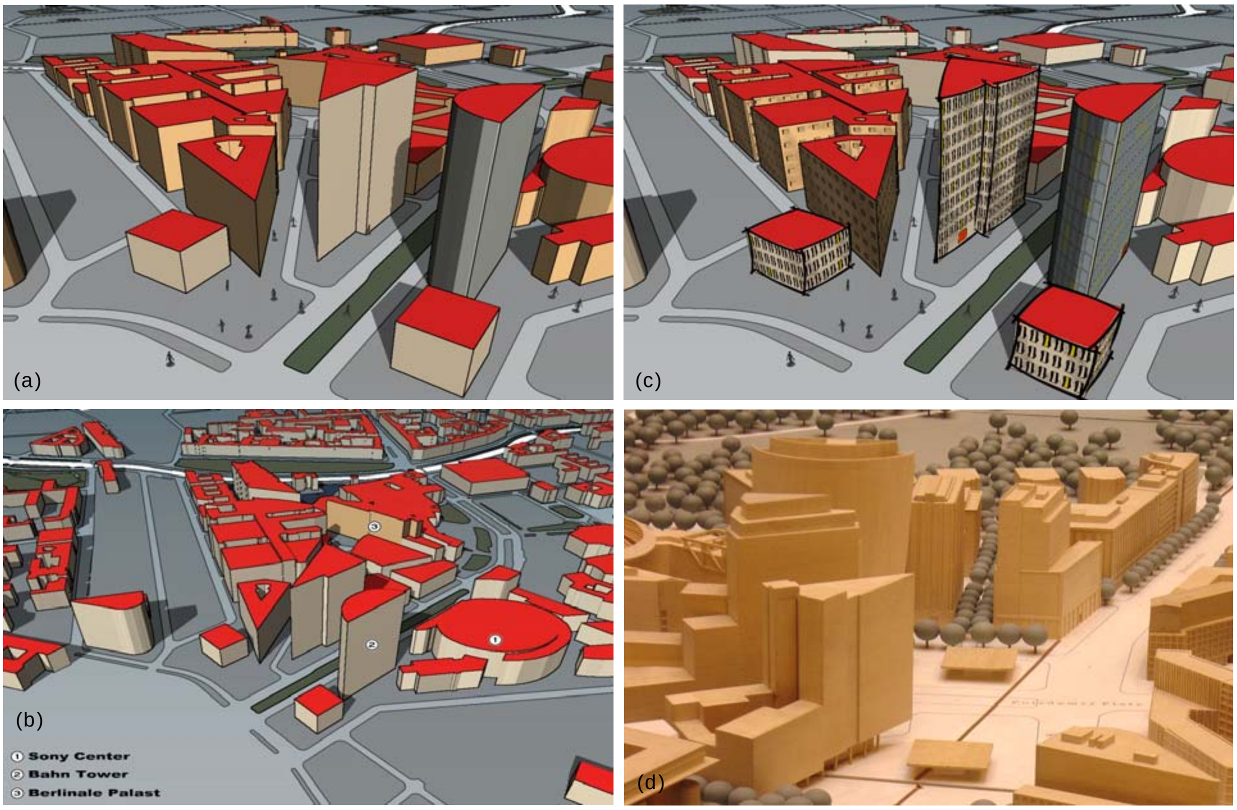

**Figure 7.** Examples of nonphotorealistic visualizations of the 3D city model of Berlin, a joint project with the Senate of City Development. (a) No stylized edges, regular shadows. (b) Bird's-eye view with reference numbers. (c) Stylized edges, facade textures, and regular shadows. (d) The physical 3D city model of Berlin (seen approximately from the opposite direction).

**Edge Rendering.** To render a principal edge, we use a quad textured with a stroke alpha-texture. The quad is rotated around the edge so that its normal points always towards the viewer. In addition, stroke color and stroke transparency can be varied. The quads can also be stretched beyond the edge endpoints and rotated around the edge midpoint. In this way, different degrees of uncertainty can be visualized, for example for planned buildings. To control stroke density in the image, we fade out principal edges with increasing distance to the viewer. Using vertex programs the edge stylization can be applied to buildings at interactive frame rates.

**Hybrid Edge Enhancement.** The hybrid algorithm allows for combining the advantages of both edge techniques. The image-space technique detects silhouettes in contrast to the object-space technique, which provides a high degree of freedom for graphics design. Together, they lead to an optimal tool for authoring 3D building geometry.

Phase 4 renders all additional scene geometry such as 3D objects indicating scale, for instance, stylized humans (Fig. 7a,c) or labels positioned in the view plane that provide texts or numbers referring to scene objects (Fig. 7b).

## 6. APPLICATIONS AND CONCLUSIONS

Cartographic applications are manifold since abstracting from a city's real appearance to communicate the principle geometry and topology as well as major landmarks and surroundings are key requirements for visual guides and maps. For example, interactive bird's-eye view maps represent a direct application area. The images shown have been taken from a city planning system currently being developed together with the Senate of the City of Berlin. Another important application domain represents 3D city information systems. The visualization technique can provide an interactive visual interface to thematic information about an urban region. Since the mapping of thematic information to 3D geometry and appearance can be configured, user-oriented and task-oriented interface can be easily constructed. For example, to explore and analyze data of radio network systems, we are developing an application for a telecommunication company. Here, the abstraction inherent to the images helps network administrators to see relevant data such as building height, building usage, shadowing, and network-cell boundaries. A further application domain represents computer games that require "cartoon-like" urban areas and take advantage of the stylistic power of NPR visualizations.

In this paper we have presented an illustrative visualization technique of 3D city models that achieves expressive, stylized depictions *based on relatively scarce input data*, mainly given by 2D ground plans, building heights, and additional associated thematic information. Since outlined and stylized buildings are central elements in 3D city models, we have focused on their presentation based on a general image-space approach combined with an object-space approach for stylized edges. The conceptual innovations include the mostly automated construction of geometry and appearance driven by thematic information as well as high design flexibility. As key insight we have demonstrated that successful visualization strategies of historical and traditional 3D city views can be transferred to the realm of interactive 3D visualization. It also has become manifest that convincing illustrative 3D views require manifold illustrative design elements – the presented technique implements just some of them. More elements need to be investigated. Our future work is addressing the automated transformation of photorealistic facade and roof textures into textures with illustrative appearance as well as the generalization of building blocks used to generate LODs for aggregated buildings. We see also potential for future enhancements in designing a set of illustrative city furniture objects and their semantic-driven use for and integration in 3D city views.

## ACKNOWLEDGEMENTS

We would like to thank Takis Sgouros, Senate Department of Urban Development, Berlin, for his collaboration and providing us the initial geodata for the Berlin city model as well as the German environmental foundation Deutsche Bundesstiftung Umwelt for supporting our research within the Lenne3D project (www.lenne3d.de). We are also grateful to Konstantin Baumann, Johannes Bohnet, and Stefan Maass for their contributions to VRS (www.vrs3d.org) and LandXplorer (www.3dgeo.de) and A.P. Mulleners from Mulleners+Mulleners Architects and Urban Planner for giving us permission for Figure 1c.

# REFERENCES

1. Hu, J., You, S., Neumann, U., “Approaches to large-scale urban modeling”, *IEEE Computer Graphics and Applications*, 23(6): 62-69, 2003.
2. Förstner, W., “3D City Models: Automatic and Semiautomatic Acquisition Methods.”, *Proceedings Photogrammetric Week*, University of Stuttgart, 291-303, 1999.
3. Parish, Y., Müller, P., “Procedural modeling of cities”, *Proceedings of ACM SIGGRAPH 2001*, 301-308, 2001.
4. Altmaier A., Kolbe, T.H., “Applications and solutions for interoperable 3D geo-visualization”, *Proceedings of the Photogrammetric Week 2003*, Wichmann Verlag, 2003.
5. Duchaineau, M., Wolinsky, M., Sigeti, D., Miller, M.C., Aldrich, C., Mineev-Weinstein, M., “ROAMing terrain: Real-time optimally adapting meshes”, *Proceedings IEEE Visualization ’97*, 81-88, 1997.
6. Davis, D., Ribarsky, W., Jiang, T.Y., Faust, N., Ho, S., “Real-time visualization of scalably large collections of heterogeneous objects”, *Proceedings IEEE Visualization 1999*, 437-440, 1999.
7. Willmott, J., Wright, L.I., Arnold, D.B., Day, A.M., “Rendering of large and complex urban environments for real time heritage reconstructions”, *Proceedings VAST 2001: The International Symposium on VR, Archaeology, and Cultural Heritage*, 2001.
8. Flack, P.A., Willmott, J., Browne, S.P., Arnold, D.B., Day, A.M., “Scene assembly for large scale urban reconstructions”, *Proceedings VAST 2001: The International Symposium on Virtual Reality, Archaeology, and Cultural Heritage*, 2001.
9. Ribarsky, W., Wasilewski, T., Faust N., “From urban terrain models to visible cities”, *IEEE Computer Graphics and Applications*, 22(4): 10-15, 2002.
10. Beck, M., “Real-time visualization of big 3D city models”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV-5/W10, 2003.
11. Schaufler, G., “Rendering complex virtual environments”, Dissertation, University of Linz, 1998.
12. Strothotte, T., Schlechtweg, S., “Non-photorealistic computer graphics: Modeling, rendering and animation”, Morgan Kaufman**,** 2002.
13. Raskar, R., “Hardware support for non-photorealistic rendering”, *SIGGRAPH/Eurographics Workshop on Graphics Hardware*, 41-46, 2001.
14. Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S., Strothotte, T., “A developer’s guide to silhouette algorithms for polygonal models”, *IEEE Computer Graphics and Applications*, 23(4):28-37, 2003.
15. Nienhaus, M., Döllner, J., “Edge-enhancement - An algorithm for Real-time non-photorealistic rendering”, *Journal of WSCG*, 11(2):346-353, 2003.
16. Gooch, A., Gooch, B., Shirley, P., Cohen, E., “A non-photorealistic lighting model for automatic technical illustration”, *Computer Graphics (Proceedings of SIGGRAPH 1998)*, 447-452, 1998.
17. Weiskopf, D., Ertl, T., “A depth-cueing scheme based on linear transformations in tristimulus space”, Technical Report No. 2002/08, University of Stuttgart, 2002.
18. Imhof, E., „Cartographic relief representation”, DeGruyter, 1982.
19. Bollmann, F., „Entstehung von Bildstadtplänen“, *Karlsruher Geowissenschaftliche Schriften*, Reihe A, Band 4, 93-98, 1986.
20. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W., “Instant architecture”, *Proceedings of ACM SIGGRAPH 2003*, 669-677, 2003.
21. Felkel, P., Obdrmalek, S., “Straight skeleton implementation”, *14th Spring Conference on Computer Graphics*, 210-218, 1998.
22. Laycock, R.G., Day, A.M., “Automatically generating roof models from building footprints”, *Proceedings of WSCG*, Poster Presentation, 2003.
23. Everitt, C., Kilgard, M.J., “Practical and robust stenciled shadow volumes for hardware-accelerated rendering”, NVidia Online Published Paper, 2002.
24. Crow, F.C., “Shadow algorithms for computer graphics”, *Computer Graphics (SIGGRAPH '77 Proceedings)*, 11(2): 242-248, 1977.
25. Döllner, J., Walther, M., “Real-time expressive rendering of city models”, Seventh International Conference on Information Visualization, *Proceedings IEEE 2003 Information Visualization*, 245-250, 2003.