EXPRESSIVE VIRTUAL 3D CITY MODELS

Jürgen Döllner and Henrik Buchholz

University of Potsdam – Hasso-Plattner-Institut
Helmert-Str. 2-3, 14482 Potsdam, Germany
doellner@hpi.uni-potsdam.de, buchholz@hpi.uni-potsdam.de

Virtual 3D city models include components such as 3D building models, transportation network models, and vegetation models. Traditional applications of virtual 3D city models focus on presenting photorealistic views. The potential of 3D city models, however, goes beyond virtual reality. In general, virtual 3D city models can serve as user interfaces to complex, spatial urban information, that is, for exploration, analysis, and communication of thematic and spatial information. For the last decades, the intrinsic concepts of real-time 3D computer graphics le~~a~~d to near-photorealistic, interactive virtual 3D city models. With the recent advent of non-photorealistic 3D rendering, a new genre in computer graphics, a repertoire of illustrative, expressive, and artistic graphics techniques becomes viable to developers and designers of virtual city models. This contribution outlines main techniques and discusses consequences for cartographic information display based on 3D city models.

Introduction

Virtual 3D city models represent a core part of urban geodata infrastructures, and an increasing number of virtual 3D models are systematically built based on a wide range of techniques for acquisition, classification, and analysis of urban data derived from, for example, laser scans, aerial photography, and cadastre information bases (Hu et al. 2003; Ribarsky 1999; Förstner 1999). Typical components of virtual 3D city models include 3D building models, transportation network models, and vegetation models. For this, these models act as integration platforms for 2D and 3D spatial data based on the underlying paradigm of the virtual city.

In computer graphics, a major research area represents rendering and modeling of virtual 3D environments. In particular, photorealistic display is used in many applications; it requires and, therefore, depends on a large amount of geometric and graphical detail to achieve impressive results, and has to cope with the resulting geometric and graphical complexity of city models. The lack of sufficient detail is often frustrating – not because highly detailed data would actually be required at early stages of construction but because without such detail the visual results are not convincing. For many applications, however, the virtual-reality paradigm is neither cognitively adequate nor adequate with respect to the task to be supported by interactive visualizations.

In cartography, the visual representation of urban spatial information has a long history and has yielded many principles for drawings of this category. The most prominent examples include panoramic maps of cities and landscapes as well as bird's-eye views of cities. They show a high degree of visual clarity and geometric abstraction – preferring abstraction to realism. The historic maps in Figures 1 illustrate a few principles of their design: choosing colors and shadings carefully; simplifying and scaling geometric structures; and sketching dimensions of and relationships between objects. Their rich contents and visually pleasing design triggers the curiosity of the observer.

Motivated by techniques of historic 3D representations, this contribution discusses limitations of photorealistic display of 3D environments and introduces a non-photorealistic rendering technique that allows for interactive, expressive virtual 3D city models.

The visual characteristics and quality of virtual 3D city models are still defined by the inherent characteristics and quality of 3D computer graphics dedicated for the last 30 years to rendering technology that aims at the production of realistic images. While photorealistic rendering has been a domain of non-interactive rendering systems (e.g., based on ray-tracing, photon-tracing, or radiosity calculations) and still involves high computational costs, photorealistic

Figure 1: City Depiction of Graz, Austria, 16th century (left). City Depiction of New York, 1870 (right).

rendering recently becomes fully real-time enabled due to progress in algorithms and computer graphics hardware (Akenine-Möller and Haines 2002).

Photorealistic Display and Its Limitations

Photorealistic display implies a number of limitations with respect to virtual 3D city models:

- To produce convincing photorealistic depictions, complete and high quality data has to be processed such as exactly matching facade textures and high-resolution aerial photography. The larger the virtual 3D city model, the higher the costs for data acquisition. In most cases, required data will not be available for a whole 3D city model. As a consequence, the images are faced by a breach of graphics style.
- To integrate thematic information (e.g., state of repair, average rental fees) in photorealistic depictions, the information needs to be visually combined with the virtual 3D city model, which turns out to be difficult because textured façades, roofs, and road systems dominate the image space.
- To visualize complex information, photorealistic details increasingly interfere with a growing number of information layers.
- To express objects of city models in different states, e.g., existing, removed, and planned buildings, photorealism does not offer a broad range of graphics styles to communicate these variations such as by sketchy and outlined drawings.
- To generate compact depictions for displays with minimal capacities, e.g., on mobile devices, photorealism frequently fails due the visual complexity inherent to photorealistic images. For example, a scaled-down version of a digital photography has a drastically lower information level compared to a scaled-down version of a hand-drawn sketch of the same scenery.

With respect to illumination and shading, physically oriented models are used in most applications. For example, the Phong illumination model, which approximates the physical interaction of light in 3D environments, is actually shaping current visualizations – it was not possible until recently to develop alternatives if real-time rendering was required because it forms part of the hardware-implemented rendering pipeline. This situation is changing fundamentally with the advent of programmable computer graphics hardware.

Another limitation represents single-pass rendering: For rendering a single image, the rendering system traverses the scene description in a single pass sending directly graphics data to the graphics hardware. However, a large number of graphics techniques become possible only by more complex, multiple rendering passes, which generally collect data, prepare intermediate images, and perform imaging operations. Edge enhancement, for example, requires one rendering pass that detect discontinuities in the z-buffer, and one rendering pass that detects discontinuities in an intermediate image encoding surface normals, and overlays both intermediate images with the final image (Nienhaus and Döllner 2003).

The geometric complexity of large 3D environments represents a limitation for real-time rendering. A number of optimization techniques have been developed, which are generally not bound to photorealistic display, that is, they can be applied to non-photorealistic display as well. Level-of-detail algorithms provide efficient rendering of large, complex 3D terrain data (Duchaineau et al. 1997) and geo-spatial objects (Davis et al. 1999; Willmott et al. 2001). Numerous optimizations for virtual environments have been developed such as discrete and continuous multi-resolution representations of geometric objects, view-frustum and occlusion culling, imposter techniques (Schaufler 1998), scene-graph optimizations, and recently out-of-core visualization techniques, which directly render large amounts of data from external memory in real-time. Akenine and Haines (2002) provide a general introduction to these subjects.

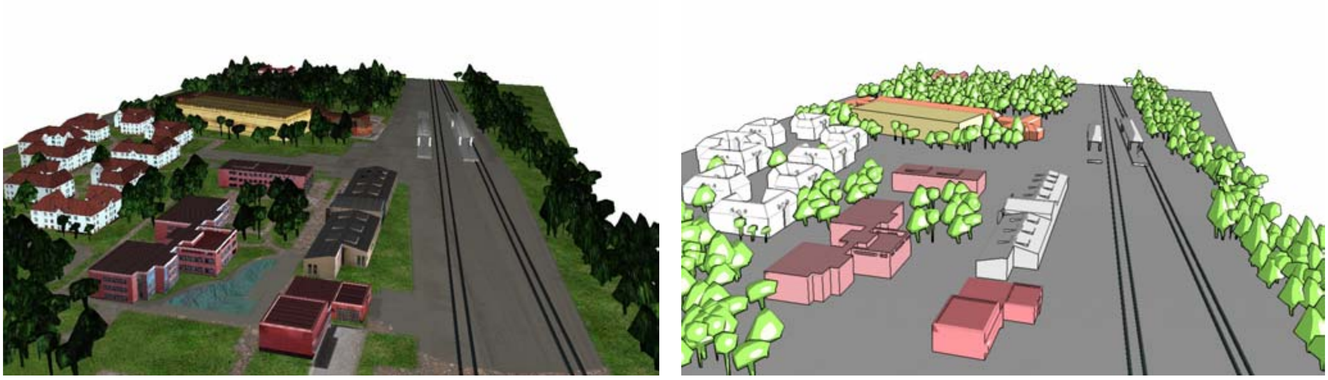

Figure 2: Real-time photorealistic and non-photorealistic display of a virtual 3D campus map.

Photorealistic rendering techniques do not provide optimal means for visual abstraction and, thus, for visualizing abstract and complex information as required by virtual 3D city models. Figure 2 illustrates our approach, comparing two snapshots of the same interactive 3D city model using photorealistic and non-photorealistic display, respectively.

Non-Photorealistic 3D Rendering

With the advent of non-photorealistic 3D rendering a repertoire of illustrative, expressive, and artistic graphics techniques becomes viable to developers and designers of 3D geovirtual environments. Most researchers agree that the term "non-photorealistic" is not satisfying because the notion of realism itself is not clearly defined nor its complement, the non-photorealism. Nevertheless, "non-photorealism" (NPR) has established itself as a key category and discipline in computer graphics starting around 1990.

Non-photorealistic computer graphics denotes the class of depictions that reflect true or imaginary scenes using stylistic elements such as shape, structure, color, light, shading, and shadowing that are different from those elements found in photographic images or those elements underlying the human perception of visual reality. As Durand (2002) points out, non-photorealistic computer graphics offers extensive control over expressivity, clarity, and aesthetic, thereby the resulting pictures "can be more effective at conveying information, more expressive or more beautiful". Strothotte and Schlechtweg (2002) as well as Gooch and Gooch (2001) give a broad introduction to concepts and algorithms of non-photorealistic computer graphics.

Generally characteristics of non-photorealistic 3D rendering include the ability to sketch geometric objects and scenes, to reduce visual complexity of images, as well as to imitate and extend classical depiction techniques known from scientific and cartographic illustrations.

Fundamental techniques of real-time non-photorealistic 3D rendering address the following characteristics:

- Lighting and Shadowing. The programmable rendering pipeline allows developers to implement new illumination models such as the model introduced by Gooch et al. (1998). In addition, real-time shadow techniques can be seamlessly integrated and, thereby, provide a valuable depth cue;
- Coloring and Shading. Non-photorealism allows for vivid and domain-specific color schemes. Cartoon-like appearance is achieved by specialized color schemes such as tone shading;
- Edges and Silhouettes. Edges as visually important elements can be treated as "first-class" objects in depictions, that is, they can be enhanced and stylized. Isenberg et al. (2003) give an overview of algorithms for outlines and silhouettes;
- Texturing. Texturing is used as a fundamental computer graphics operation (Haeberli 1990) to outline the principle curvature of surfaces, to simulate strokes using virtual brushes, to procedurally generate textures, and to implement image-based operations such as image overlays, image blending, convolution filtering, etc.

Technically, non-photorealistic 3D rendering is implemented by redefining the geometry stage and rasterization stage of the standard 3D rendering pipeline using application-defined procedures, known as shaders, which implement geometric transformations, illumination calculations, and pixel shading. For example, we can substitute the classical Phong illumination model by a cartoon-like illumination shader, which reduces the number of shades per color. Furthermore, non-photorealistic 3D rendering takes advantage of multi-pass rendering, that is, they process a 3D scene description several times, generating and combining intermediate images into a final image. This way, for example, enhanced edges and shadows can be included in the overall rendering process. Furthermore, today's programmable computer graphics hardware supports the implementation of non-photorealistic 3D rendering. Therefore, most techniques can be used in interactive or even real-time applications and systems.

Cartographic, Cognitive, and Graphics Aspects of City Models

We have analyzed the rendering of city models from the perspective of cartography, cognition, and non-photorealism. We derived several principles that are found in many non-digital drawing techniques.

City Models as Maps

City models can be considered as map-related 3D representations of spatial data. In addition to 2D information, such as topographic maps, thematic maps, and 2D ground plans, they include 3D information, for example, 3D terrain models, 3D buildings, and 3D vegetation models. Interactive representations allow for better exploration and analysis, but complicate navigation and orientation, unless special care is taken to support these user activities.

Geometric Projection

In classical bird's-eye views both orthogonal and perspective projections are used. The Bollmann's maps of many European as well as US cities are well-known examples of 'pictorial city maps' (Bollmann 1986). Traditionally they choose the east or west direction for projection because important, optically dominant landmarks such as well known public buildings and churches are oriented that way. In addition, map designers often attempt to diagonally capture important quarters and streets of houses. Whatever direction is taken, approximately half of the model is not shown in static views. Thus an interactive 3D city map can achieve a substantially better communication of spatial information.

Graphical Techniques

Maps of city models attempt to maximize visual clarity. Most importantly, edges are enhanced to stress the contours of buildings. Colors are chosen according to semantics and aesthetics – they do not necessarily correspond to the natural colors. The overall number of colors is kept small. Strothotte et al. (1994) propose graphical techniques of a sketch rendering system based on insights from perceptional psychology.

Geometrical Techniques

The terrain model represents the reference surface for city models. In traditional city maps, the terrain surface is often ignored or just indicated where its morphology is significant. For interactive visualizations, a 3D terrain model can improve the understanding of the city model and the related terrain; its impact on the visualization can be controlled by appropriate scaling of the terrain surface.

To cope with complex geometry, detail reduction is used both at the technical level by multiresolution modeling and level-of-detail techniques, and at the semantic level by generalizing buildings to quarters if their distance to the camera exceeds a given threshold. As the sides of buildings provide more characteristic information about them than the roofs, the roofs can be scaled down in height while the main bodies of the buildings are scaled up. This is done in traditional panoramic views of cities to meet the spectator's expectation, who is used to seeing the city as a pedestrian.

Orientation Cues

In a complex and dense urban area landmarks serve as means of orientation for the user. Landmarks consist mainly of those buildings that are known a priori to the user – typically monuments and buildings of public interest. Therefore, they require an exact geometric and graphical 3D model. The visualization should guarantee that landmarks, even if they are nearly out of the view volume, do not disappear. The multiresolution mechanism might want to treat them differently from the remaining buildings because the eye is sensitive to even small changes of well-known objects.

Shape Cues

The human visual system uses different strategies to derive the shape of an object from its depiction. In abstract depictions, shades are rarely used to communicate that information. Instead, contours and edges are drawn. In particular, edge enhancement is used for separating the faces of an object. To further clarify the appearance of shapes, tone shading can be applied. Imhof (1982) studied extensively methods of terrain shading for classical cartographic depictions. Buchin et al. (2004) introduce a real-time rendering technique for non-photorealistic terrain depiction applying slope lines and tonal variations and using stroke textures to shade terrain surfaces.

Depth Cues

In classical 3D city maps, using orthogonal projections the monocular depth cues linear perspective, relative size and, texture gradient are not available. However, occlusion and known sizes provide depth cues. Frequently map designers place objects of known size (e.g., people, trees, animals) in the scenery. Furthermore shadows and fog can be used to aid depth perception. Interactive visualization additionally provides motion-based depth cues. Goldstein (1999) and Ware (2000) give a comprehensive overview of depth cues.

In our approach, we aim at real-time expressive 3D city models characterized as follows:

- Concentrate on illustrative visualizations emphasizing on high perceptual and cognitive quality to effectively communicate contents, structure, and relationships of urban objects and landscape objects as well as related thematic information.
- Enable meaningful visualizations even for the case of scarce spatial information since high-quality and complete sets of geodata tend to be rarely available for large-scale urban areas or landscape.
- Enable fully automated generation of visualizations while offering great flexibility in the degree of control of graphical parameters.
- Achieve real-time rendering and, thereby, allow for interactive manipulation, exploration, analysis, and editing of all aspect of 3D city models.

Components of Virtual 3D City Models

Virtual 3D city models are based on 2D geodata such as topographic maps, thematic maps, and 2D ground plans, and 3D geodata such as 3D terrain models, 3D building geometry, and 3D vegetation objects. In this section, we discuss building and vegetation modeling of our approach.

Building Models

Buildings are major visual objects of 3D city models. In general, digital data of buildings is acquired based on administrative data (e.g., cadastre records), laser scanning, and aerial photography; for an overview of state-of-the-art techniques see Hu et al. (2003).

To construct block models one can extrude 2D ground polygons to heights derived from laser scans. In practice, high-quality, precise models of large areas are hardly available due to involved time and costs efforts. For this reason, the non-photorealistic 3D rendering technique should allow for generating compelling depictions of most buildings' geometry in an automated way. In addition, the technique should support 3D CAD models of landmark buildings. Although only a small percentage of buildings in typical 3D city model are specified this way, they possess a high visual importance and need to be seamlessly integrated into the depiction.

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. Laycock and Day (2003) introduce a roof generation technique based on the automated calculation of straight skeletons (Felkel and Obdrmalek 1998) within ground polygons.

As the sides of buildings provide more characteristic information about them than the roofs, the roofs can be scaled down in height while the main bodies of the buildings are scaled up. This is done in traditional panoramic views of cities to meet the spectator's expectation, who is used to seeing the city as a pedestrian.

For most urban areas no detailed data about building appearance can be assumed. Although laser scanning and photogrammetry allow for capturing facade textures, time and cost efforts are still high. Procedural approaches for grammar-based building designs offer an interesting alternative; Wonka et al. (2003) introduced the term "instant architecture" for a procedural approach to generating building appearance. 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.

Building Quality Levels. Building geometry as most important category of data can be generally classified into five quality levels (Altmeier and Kolbe 2003):

- 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 level 3 with additional interior design.

Our technique focuses on processing and generating buildings of level 1 and 2. In addition, it integrates buildings of level 3.

Vegetation Models

We distinguish between vegetation objects without and with individual characteristics. In the former case, objects are generally placed automatically based on a distribution function, for example, lawns and bushes. In the latter case the objects are placed and modeled as individual 3D objects, for example, large trees.

To model a specific kind of vegetation object, a 3D geometry prototype is designed using the xfrog modeling software for organic objects. The prototype can be instantiated multiple times, and it can be varied with respect to parameters such as height, density of foliage, and geometric distortion to derive copies with sufficient individual visual characteristics. Deussen et al. (2005) provide a survey of modeling and rendering virtual 3D landscapes with vegetation objects as major building blocks. Döllner et al. (2005) give an overview of the interactive authoring and presentation system Lenné3D for virtual 3D landscape models.

3D vegetation rendering is confronted with the high geometric complexity of vegetation objects. For example, a regular tree typically consists of 50.000 to 100.000 triangles. Coconu and Hege (2002) describe a first hardware-accelerated rendering technique that copes with massive numbers of vegetation objects such as trees in virtual 3D landscapes. The geometry models allows for both photorealistic depiction as well as non-photorealistic depiction. Terrain rendering and vegetation rendering are not explicitly addressed in this paper.

Non-Photorealistic Rendering Technique

The rendering technique we propose provides stylized edges, enhanced edges, n-tone shading, shadow casting, and procedural facade textures.

Stylized Edges

Stylized edges denote the explicit representation and rendering of visually important edges of 3D objects. These edges are additional elements of scene objects. A detailed overview of stylized edges and its application for city models can be found in Döllner and Walther (2003).

Given the start and end points of an edge, we want to draw a sketchy line along the edge. In hand drawings, lines along

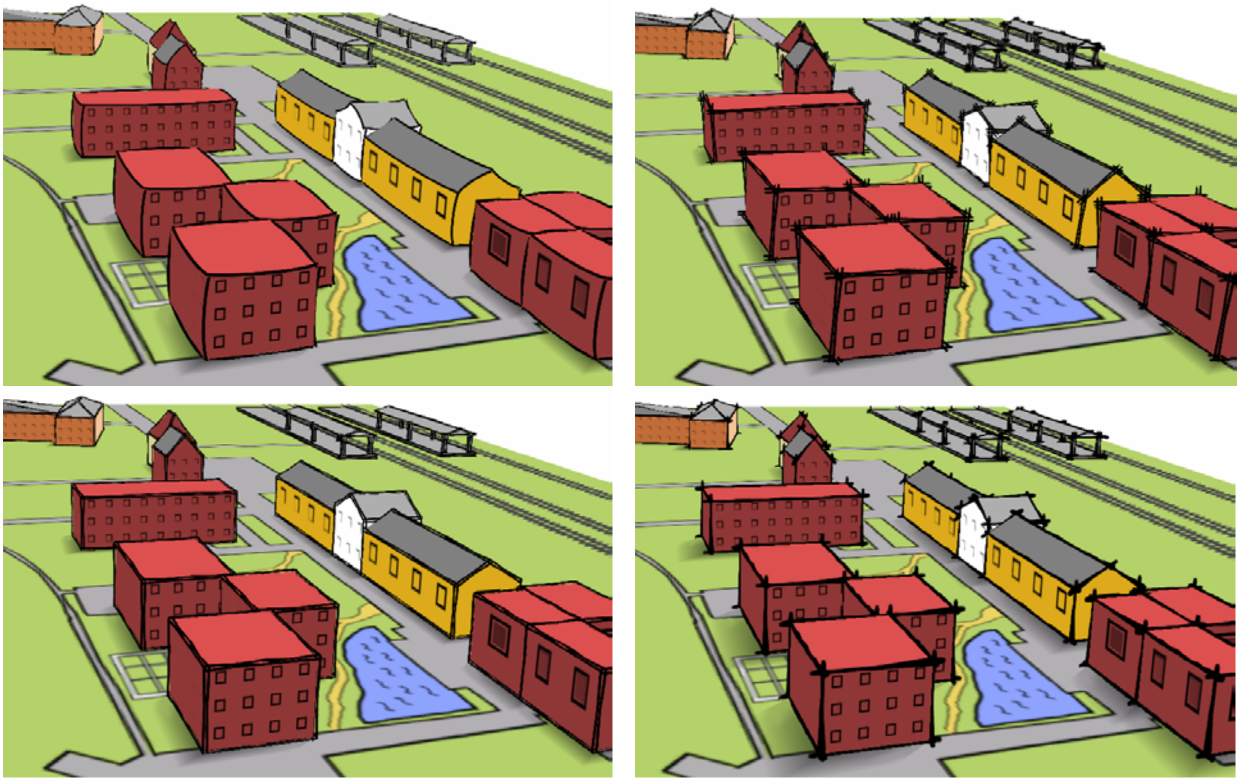

Figure 3: Stylized edges to enhance building geometry. Different stroke textures, stroke lengths, and stroke tilts are used to achieve different graphics styles.

edges usually do not start and end exactly at the ends, but near the ends. This results in a higher visual quality and expressiveness. Inspired by this, we introduced two rendering parameters, one determining the tilting angle and one the length of the line. To express uncertainty, tilting and length can be varied as illustrated in Figure 3. After moving the starting and end points according to the tilt and length parameter, we apply cylindrical billboarding, using the vector from the starting point to the end point as an axis. For implementation, we can employ per-vertex programming on the graphics hardware for calculating the billboards.

Stylized edges allow us to control appearance attributes such as stroke form, stroke width, brightness, color, length, and exactness. For example, the actual drawn edge can be stretched, shrunk, or disturbed to express fuzzy, sketchy, or slightly overlapping outlines. The object-space approach offers a high degree of expressiveness, in particular, if the underlying model explicitly defines edges as in city models. When rendering a complex scene we do not always want to draw all edges and not necessarily all edges with the same level of detail. For instance in our implementation we draw only the edges that are best visible to the viewer using three textures as explained above. Edges with low visual importance are either drawn using only the texture for the middle or not at all. As criteria the actual length of an edge in image space and the distance to the camera can be used.

Edge Enhancement

We apply edge enhancement to render building geometry and ground geometry such as streets, green-space areas, or rivers. Image-space edge enhancement detects visually important edges such as silhouettes, borders, and crease edges. The enhanced edges result in a homogenous and generalized visual depiction of 3D geometry while emphasizing their principle composition.

We obtain these edges in screen space (i.e., not on a per-object basis) by determining discontinuities in the depth buffer and in the surface normal space of the rendered scene. The results are combined into an intensity image that encodes screen-space edges of 3D scene geometry. The image is stored as texture, the edge map, which allows for enhancing edges with varying visual weight since they can be blended into 3D scene geometry. Nienhaus and Döllner (2003) describe this rendering technique in detail. In our examples, we reduce the “edge-ness” in the image with increasing camera distance. In Figure 4, an example of intermediate results and the final edge-enhanced image is given.

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.

Tone Shading

Maps of city models attempt to maximize visual clarity. Most importantly, edges are enhanced to stress the contours of buildings. Colors are chosen according to semantics and aesthetics – they do not necessarily correspond to the natural colors. The overall number of colors is kept small. Strothotte et al. (1994) propose graphical techniques of a sketch rendering system based on insights from perceptional psychology.

Major inspirations for our work were cartographic picture maps and other hand drawings of cities. In colored drawings the illustrator usually abstracts from the realistic colors of the urban objects and greatly reduces the number of colors used in the representation. In general only two or three colors and for each color only two or three tints are used.

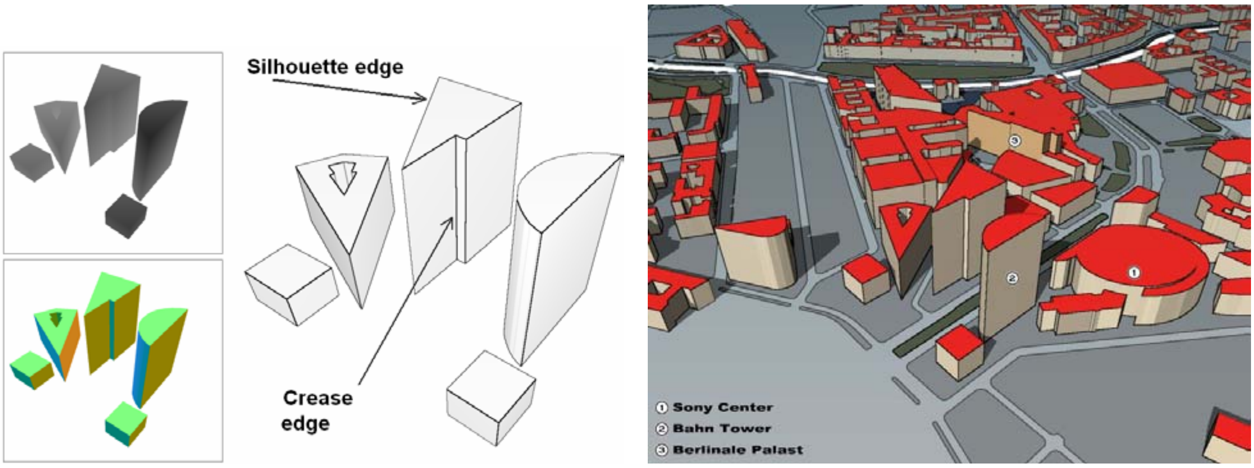

Figure 4: Edge enhancement. Based on the depth image and the normal image (left), an edge image is calculated (middle). The edge map is used to enhance geometry in image space (right).

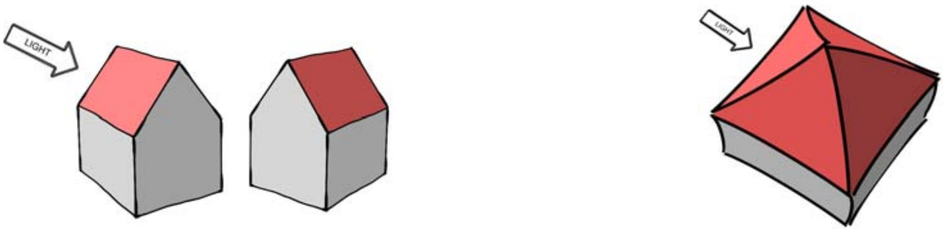

Figure 5: Two-tone shading (left). Three-tone shading (right).

The choice of the colors is based on aesthetic and other reasons as well as the actual colors of the city. Inspired by this we decided to use a two-tone or three-tone shading for the faces, which assigns the color according to the angle of the face to the light as illustrated in Figure 5.

Technically this can be done in the same way as cartoon shading but instead of the intensity we use the dot product of the normal of the face and the light direction to determine the color. We observed that three tone shading supplies better results for interactive applications than two tone shading because neighboring faces of one building will have different colors assuming the building does not have more than four sides. Using two-tone shading this is not the case. Thus there is much less contrast when moving through a scene and the light direction is much harder to determine. This, of course, is not a problem in traditional cartography where the scenes are static and the light direction can be chosen suitably. Often the light shines from the upper left. This can also be done with our shading when producing static scenes.

Shadows

In architectural drawings shadows are used for conveying three-dimensionality and spatial relationships, which aids the comprehensibility and naturalism, and, for adding vividness. 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. For example, the length of shadows represents a measure for the heights of the buildings. Shadows should not obscure any parts of the geometry. Also they should harmonize with the rest of the drawing and be clearly recognizable.

There are different approaches to compute shadows in real-time. Most important techniques include shadow volumes and shadow maps. We apply a real-time implementation (Everitt and Kilgard 2003) of the shadow volume technique (Crow 1977). In a pre-processing step, the rendering technique computes shadow volume geometry for a given 3D 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 of each fragment produced for the quad correspond to windows coordinates.

Procedural Façade Textures

To automate the depiction process for large-scale 3D city models, we combine texture elements to facades using multi-texturing for facades and walls for which we do not have an individual texture. In a preprocessing step we create multiple sets of texture coordinates for each façade and wall, which are used by a fragment shader to compose the facade texture for each building. To compose a façade procedurally, the algorithm has the following principle steps:

1. The material texture is wrapped around the facade;
2. The window texture is copied to all estimated windows of the facade;
3. The door texture is placed at explicitly given door position of the facade;
4. Inspired by hand-made drawings we blend between the facade color and the material texture so that the material is visible at some irregular formed areas.

A general overview of procedural texture generation for photorealistic city depictions is given by Parish and Müller (2001). In Figure 6, procedural facade textures are demonstrated.

Applications of Expressive Virtual 3D City Models

Expressive city models allow for manifold new applications in cartography, such as illustrative maps, artistic maps, and informal maps. More general, their applications encompass visual interfaces to spatial information required by decision-support and information systems.

Figure 6: Real-time expressive virtual 3D city model based on stylized edges, enhanced edges, shadows, n-tone-shading, and procedural facade textures.

A major application area represents urban planning and development. The achieved abstract, expressive display comes close to the graphics quality of architectural plans, while being fully interactive. Furthermore, they serve to illustrate concepts and principles underlying suggested plans. In addition, expressive 3D city models facilitate people's participation in decision-making through comprehensible representations. The presented approach is being applied within Lenné3D, a system for interactive, virtual 3D landscape planning and participation (Paar and Rekittke 2005).

Another application area includes geo-information systems, transport information systems, and radio-network planning systems. Here, expressive virtual 3D city models play a fundamental role in the user interface, providing visualization of data and monitoring processes related to urban areas.

Further applications can be found in education and entertainment, for example interactive gaming environments and comic worlds, and atmospheric and edutainment environments for narratives–imagine a cartoon-like story, taking place in a 3D non-realistically drawn "real city" or an Internet platform for neighborhood communication.

Conclusions

Expressive virtual 3D city models provide excellent means for displaying geovirtual 3D environments. In particular, they facilitate visual abstraction as a primary technique to effectively communicate complex spatial information and associated thematic information. They circumvent classical weaknesses of photorealistic 3D city models such as low-quality facade textures, inconsistent illumination, or high visual complexity induced by aerial photography.

Non-photorealistic 3D rendering still raises several questions with respect to its impact to map-like and cartographic presentations. What are specific non-photorealistic 3D rendering techniques for interactive map-like depictions? How do non-photorealistic geovisualizations cooperate with photorealistic ones? How to generalize 3D geo-objects for non-photorealistic depiction?

Acknowledgements

We would like to thank our colleagues of the LandXplorer system (www.3dgeo.de), which has been used as implementation platform. We would like to thank the German environmental foundation Deutsche Bundesstiftung Umwelt for supporting our work within the Lenné-3D research project (www.lenne3d.de). We also would like to thank Prof. Christian Herrmann, Karlsruhe, for encouraging and inspiring our work in non-photorealism and cartography.

References

Akenine-Möller, T., Haines, E. 2002. Real-Time Rendering. A K Peters Ltd, 2nd Ed.

Altmaier A., Kolbe, T.H. 2003. Applications and Solutions for Interoperable 3D Geo-Visualization. Proceedings of the Photogrammetric Week 2003, Wichmann Verlag.

Bollmann, F. 1986. Entstehung von Bildstadtplänen. Karlsruher Geowissenschaftliche Schriften, Reihe A, Band 4, 93-98.

Buchin, K., Sousa, M.C., Döllner, J., Samavati, F., Walther, M. 2004. Illustrating Terrains using Direction of Slope and Lighting. 4th ICA Mountain Cartography Workshop, Vall de Nuria, Spain.

Coconu, L., Hege, H.-Chr. 2002. Hardware-Accelerated Point-Based Rendering of Complex Scenes. Proceedings of the 13th Eurographics Workshop on Rendering Techniques. Pisa, Italy, 43-52.

Crow, F.C. 1977. Shadow Algorithms for Computer Graphics. Computer Graphics (Proceedings of SIGGRAPH '77), 11(2):242-248.

Davis, D., Ribarsky, W., Jiang, T.Y., Faust, N., Ho, S. 1999. Real-Time Visualization of Scalably Large Collections of Heterogeneous Objects. IEEE Visualization 1999, 437-440.

Deussen, O., Colditz, C., Coconu, L., Hege, H.-Chr. 2005. Efficient Modeling and Rendering of Landscapes. In: Bishop, I. and Lange, E. Visualization in Landscape and Environmental Planning. Spon Press, London, 56-61.

Döllner, J., Walther, M. 2003. Real-Time Expressive Rendering of City Models. Seventh International Conference on Information Visualization, Proceedings IEEE 2003 Information Visualization. 245-250, London.

Dölllner, J., Buchholz, H., Baumann, K., Paar, P. 2005. Real-Time Virtual Landscapes in Landscape and Urban Planning. Proceedings of GIS PLANET 2005, Estoril, Portugal, in print.

Duchaineau, M., Wolinsky, M., Sigeti, D., Miller, M.C., Aldrich, C., Mineev-Weinstein, M. 1997. ROAMing terrain: Real-time optimally adapting meshes. Proceedings IEEE Visualization '97, 81-88.

Durand, F. 2002. An Invitation to Discuss Computer Depiction. Symposium on Non-Photorealistic Animation and Rendering (NPAR).

Everitt, C., Kilgard, M.J. 2002. Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering. NVidia Online Published Paper (http://www.nvidia.com).

Felkel, P., Obdrmalek, S. 1998. Straight Skeleton Implementation. 14th Spring Conference on Computer Graphics, 210-218.

Förstner, W. 1999. 3D City Models: Automatic and Semiautomatic Acquisition Methods. Proceedings Photogrammetric Week, University of Stuttgart, 291-303.

Goldstein, E. B. 1999. Sensation and Perception. Brooks/Cole, US.

Gooch, A., Gooch, B., Shirley, P., Cohen, E. 1998. A Non-Photorealistic Lighting Model for Automatic Technical Illustration. Computer Graphics (Proceedings of SIGGRAPH '98), 447-452.

Gooch, A., Gooch, B. 2001. Non-Photorealistic Rendering. A K Peters Ltd.

Haeberli, P. 1990. Paint by Numbers: Abstract Image Representations. Computer Graphics (Proceedings of SIGGRAPH '90), 207-214.

Hu, J., You, S., Neumann, U. 2003. Approaches to Large-Scale Urban Modeling. IEEE Computer Graphics and Applications, 23(6):62-69.

Imhof, E. 1982. Cartographic Relief Representation. DeGruyter.

Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S., Strothotte, T. 2003. A Developer's Guide to Silhouette Algorithms for Polygonal Models. IEEE Computer Graphics and Applications, 23(4):28-37.

Laycock, R.G., Day, A.M. Automatically Generating Roof Models from Building Footprints. Proceedings of WSCG, Poster Presentation, 2003.

Nienhaus, M., Döllner, J. 2003. Edge-Enhancement - An Algorithm for Real-time Non-Photorealistic Rendering. Journal of WSCG, 11(2):346-353.

Paar, P., Rekittke, J. 2005. Lenné3D – Walk-through Visualization of Planned Landscapes. Bishop, I. and Lange, E. (Eds.) Visualization in Landscape and Environmental Planning. Spon Press, London, 152-162.

Parish, Y., Müller, P. 2001. Procedural Modeling of Cities. Computer Graphics (Proceedings of SIGGRAPH 2001), 301-308.

Ribarsky, W., Wasilewski, T., Faust N. 2002. From Urban Terrain Models to Visible Cities. IEEE Computer Graphics and Applications, 22(4):10-15.

Schaufler, G. 1998. Rendering Complex Virtual Environments. Dissertation, University of Linz.

Strothotte, T., Schlechtweg, S. 2002. Non-Photorealistic Computer Graphics: Modeling, Rendering and Animation. Morgan Kaufman.

Strothotte, T., Preim, S., Raab, Schumann, Forsey. 1994. How to Render Frames and Influence People. Computer Graphics Forum (Proceedings of EuroGraphics 1994), 13(3):455-466.

Ware, C. 2000. Information Visualization. Morgan Kaufman.

Willmott, J., Wright, L.I., Arnold, D.B., Day, A.M. 2001. Rendering of Large and Complex Urban Environments for Real-Time Heritage Reconstructions. Proceedings VAST 2001: The International Symposium on VR, Archaeology, and Cultural Heritage, 111-120.

Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W. 2003. Instant Architecture. Computer Graphics (Proceedings of SIGGRAPH), 669-677.