

# Interactive Rendering Techniques for Highlighting in 3D Geovirtual Environments

Matthias Trapp, Christian Beesk, Sebastian Pasewaldt, Jürgen Döllner

Hasso-Plattner-Institute, University of Potsdam  
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany

**Abstract.** 3D geovirtual environments (GeoVE), such as virtual 3D city and landscape models became an important tool for the visualization of geospatial information. Highlighting is an important component within a visualization framework and is essential for the user interaction within many applications. It enables the user to easily perceive active or selected objects in the context of the current interaction task. With respect to 3D GeoVE, it has a number of applications, such as the visualization of user selections, data base queries, as well as navigation aid by highlighting way points, routes, or to guide the user attention. The geometrical complexity of 3D GeoVE often requires specialized rendering techniques for the real-time image synthesis. This paper presents a framework that unifies various highlighting techniques and is especially suitable for the interactive rendering 3D GeoVE of high geometrical complexity.

## 1 Introduction

Highlighting of objects is an important visualization principle used in human computer interaction. This form of visual feedback can be considered as an instance of higher-order visualization [1], similar to the principle of focus + context visualization [2]. Usually, a state of an object different from the original one is communicated by displaying it in a different visual style or appearance. These distinguishable styles, such as color overlay or outline, yields support of the pre-attentive cognition. Despite changes of color or texture, also geometrical properties such as scale can be used to highlight important objects [3]. A state transition can be initiated manually, e.g., by selecting one or multiple objects, or by the result of a computation, e.g., a database query. To summarize, object highlighting has a number of different applications:

**User Selection Preview:** This is the classical use case for object highlighting.

A number of selected objects are rendered using different styles. Usually the user modifies only a single object, which is in the users focus.

**Visualization of Computational Results:** In contrast to a manual selection, a number of objects can be the result of a computation, such as a database query. Instead of showing the results in a list, applied object highlight has the advantage, that the human can spatially cluster the highlighted result immediately.

**Navigation Aid:** Highlighting techniques can also be applied to landmarks, points-of-interest (POI), routes, as well as to navigation way points in order to guide the users attention.

In [4] the highlighting of scene elements, such as local and global landmarks, is considered as navigation aid that "clearly helps improve orientation in the virtual model of a real city". In [5] it is argued that, despite building aggregation and simplification, appropriate highlighting can compensate the dead value areas in virtual 3D city models. Further, Bogdahn and Coors state that "highlighting of complete buildings using false colors might be a first step. However, in a dense urban environment and for pedestrians it could be necessary to provide more sophisticated visual hints, like highlighting the correct entrance to a big building or a building complex" [6].

Robinson documented the application of highlighting techniques for information visualization in 2D GeoVE [7]. With respect to 3D GeoVE, the applications and technical implications of highlighting techniques is widely not researched. Besides the usually high geometrical complexity of 3D GeoVE, a highlighting technique has to approach the following characteristics of virtual environments:

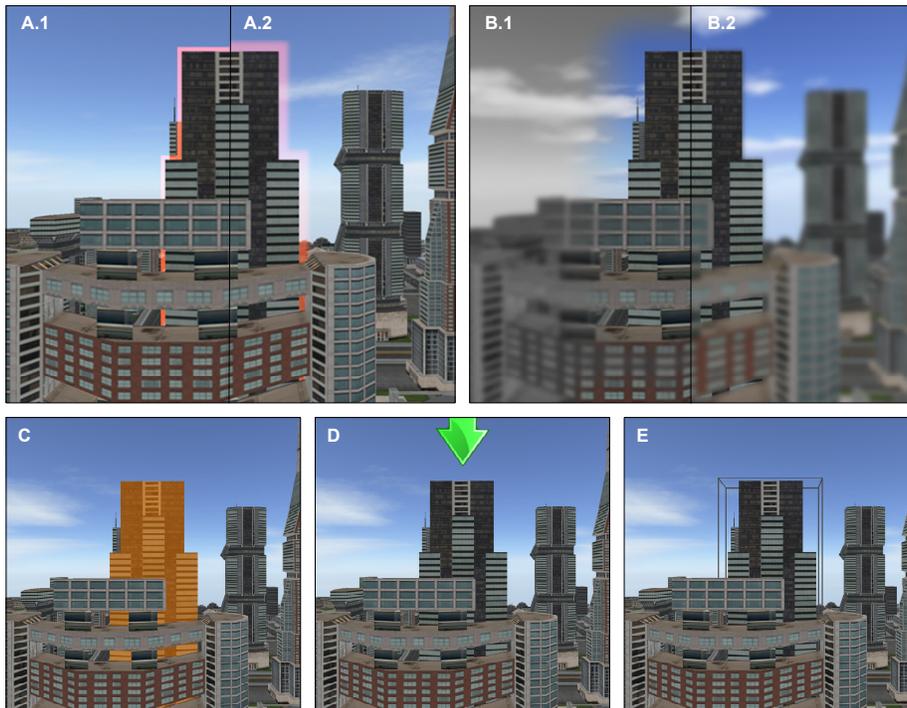
**Object Size and Shape:** In virtual city models, object size and shape can differ enormously. The size can vary from big (buildings) to very small (e.g., pedestrians, city furniture, etc.). Some shapes can have extreme spatial extent along only one major axis, thus, are not fully visible on the viewport. We can further distinguish between convex and non-convex shapes.

**Number of Objects:** Depending on the application, the visualization technique has to highlight a number of different objects simultaneously. In a simple case, only one object has to be highlighted, but in a general use case a large number of objects are encountered. For instance, this case often occurs when editing virtual city models.

**Camera Orientation and Perspective:** If unconstrained navigation and interaction metaphors are used in virtual environments, a system has to handle different perspectives and orientations of the virtual camera, which have an influence on object occlusion and the objects size on the viewport.

This paper presents applications of existing highlighting rendering techniques to 3D GeoVE. It provides a form of categorization and introduces an extensible rendering pipeline that enables an unified real-time rendering process, which can be easily integrated into existing rendering systems and applications.

This paper is structured as follows. Section 2 discusses origins of different highlighting techniques and related work concerning the topic of highlighting. Section 3 presents a conceptual overview of our framework and briefly discuss details of our prototypical implementation. Section 4 shows application examples, presents a comparison between highlighting techniques, as well as ideas for future work. Section 5 concludes this work.



**Fig. 1.** Overview of different highlighting techniques applied to a single building within a virtual 3D city model: A: outlining techniques (1: simple outline, 2: glow as generalized version of outlines) B: context-based style variances (1: vignetting, 2: semantic depth-of-field) C: focus-based style variances (color overlay) D and E: glyph-based object highlighting.

## 2 Review of Object-Highlighting Techniques

In this paper, we focus on highlighting of objects that are located on screen. In contrast to approaches for off-screen location visualization [8], on-screen highlighting techniques should enable the estimation of the 3D position, as well as the dimensions of an object.

There are a number of different approaches for 3D object highlighting and selection preview, which are mostly inferred from 2D visualization. In early stages of 2D computer graphics, different visual representations of objects are used, i.e., sprites or billboards [9] with different hues. Such an approach is not appropriate for current large-scale data set common in GeoVE. This section propose a classification of existing highlighting approaches that mainly distinguishes between three types of rendering techniques:

**Style-variance techniques** are based on modifying the appearance of an object that results in an obvious distinction from the rest of the scene. The highlighting hint depends directly on the object and is therefore referred to as *direct hint*.

**Outlining techniques** achieve this effect by enhancing the outline or silhouette of an object. We denote this setting as *attached hint*.

**Glyph-based techniques** rely on icons or glyphs that are attached to the object to be highlighted. These kind of techniques deliver an *indirect hint* to the user.

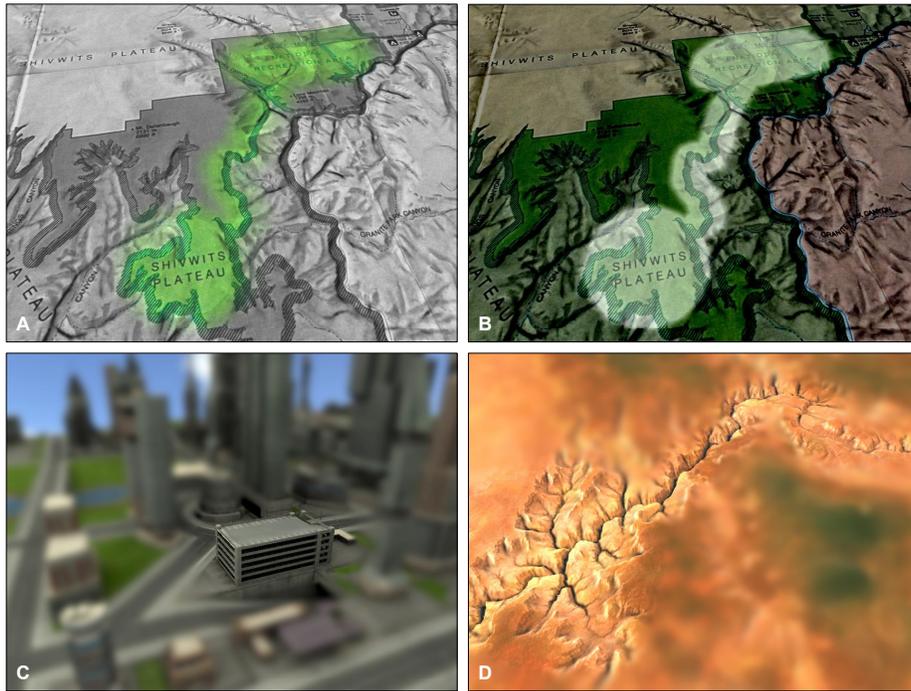
Fig. 1 provides examples of these techniques applied to a single object within a virtual 3D city model. The remainder of this section discusses these types in detail.

## 2.1 Style-Variance Techniques

Probably the most widespread highlighting techniques are instances of style variance techniques. The highlighting effect is achieved by modifying the appearance in which an object or scene is usually depicted. Such appearance modification can be arbitrarily complex, or as simple as overdrawing the same object geometry with a dominant color that can easily be distinguished from all other colors in the depicted scene. Modifications can be applied to an object or area that should be highlighted (focus-based) or to the remaining scene (context-based).

**Focus-based Style Variance Techniques:** In 3D virtual environments, focus-based style variance techniques are suitable to be applied to objects that are not occluded. Besides color overlay (Fig. 1.C), the appearance can be modified by using different rendering styles, such as wire frame rendering known from standard 3D modeling tools. Another classic rendering approach, which was often used in 2D sprite-based games, enfolds different representations of the same sprite/billboard [9]. Due to the potentially massive amount of additional data, the usage of different geometric or image-based representations of each single object is hardly manageable in large-scale 3D GeoVE. Therefore, the style invariance is created dynamically, e.g., by blending the standard appearance with a highlighting color. More advanced rendering techniques can be used, such as real-time non-photorealistic rendering (NPR) effects [10] that can be controlled manually by an artist or automatically via style transfer functions [11].

**Context-based Style Invariance:** Techniques of this category convey the appearance of the objects to highlight, while modifying the surrounding scene context in a way that the objects-of-interests are emphasized. Fig. 2 shows examples of this category of highlighting techniques: vignetting and semantic depth-of-field (SDOF) [12, 13]. In [10], an advanced technique is described that modifies the quality and intensity of edges, color saturation, and contrast.



**Fig. 2.** Vignetting using gray-scale (A) and contrast and gamma modifications (B), as well as semantic depth-of-field (C and D), as examples of context-based style variance techniques for highlighting.

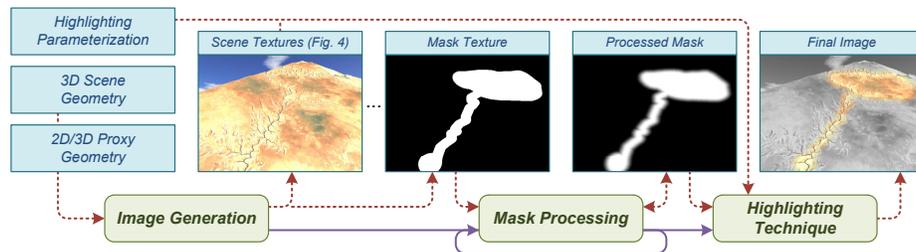
## 2.2 Outlining Techniques

Depending on the application, the change of style is not always appropriate or possible. Especially in 3D GeoVE, it can be desirable to convey the appearance of the object and the surrounding scene, such as its facade information which could be essential for orientation and navigation. In such use-cases, outlining techniques can be applied that enhance or modify the contours or silhouettes of an object only. Such enhanced contour can possess different line styles and sizes. For real-time rendering purposes, these silhouettes can be rendered using image-based [14] or the more recent geometry-based [15] approaches. Figure 1.C shows the application of image based-glow [16] as a general concept for outlining objects.

The attached hint is one of the major advantage of outline techniques: an increased visibility can be gained by increasing the thicknesses of the outline. But increased thickness of the hint also introduces occlusion of the surrounding object area. This can be compensated partially by using a drop-off function, which results in smaller occlusion than occurred for the alternative glyph-based techniques. The application of glow can be considered as a generalized variant of the outlining technique.

### 2.3 Glyph-based Techniques

Another technique performs highlighting by using additional glyphs or icons, which are placed on top or aside an object, depending on the perspective. The highlighting effect is achieved by the difference between presence or absence of a glyph. This technique is frequently used in strategic games or games employing a third person perspective. Here, usually the orientation of the virtual camera is fixed or limited. We can distinguish between *static* and *dynamic* glyphs, e.g., visugrams introduced in [17]. The latter one includes additional information about the status of an object. A simple example using the objects axis-aligned bounding box (AABB) as a glyph is depicted in Fig. 1.E.



**Fig. 3.** Conceptual overview of the rendering pipeline for applying highlighting techniques to 3D GeoVE in real-time. The data flow is depicted using stippled red and the control flow uses solid violet lines.

## 3 Interactive Rendering of Highlighting Techniques

This section introduces the basics of a highlighting framework for 3D GeoVE as well as a fully hardware accelerated, prototypical implementation for raster-based graphics. We use OpenGL and OpenGL shading language (GLSL) [18] for the presented implementation that is mainly based on image-based representations of the 3D scene [19, 20]. This has the advantage that the techniques described in Section 2 can be applied on an uniform basis. Fig. 3 shows an overview of our implementation pipeline for the real-time image synthesis. Basically, it consists of the following three components:

1. **Image Generation:** This step forms the basis for the image-based separation between objects or areas to highlight (focus) and the remaining scene (context). Fig. 4 shows the data that is required for the two subsequent steps (Section 3.1).
2. **Mask Processing:** To enable smooth transitions between focus and context regions, this step applies image-based post-processing methods, such as jump-flooding and convolution filtering to the mask texture (Section 3.2).

**3. Application of Highlighting Techniques:** In this final step, style invariance, outline, and glyph-based highlighting techniques are applied to the scene texture. The result is then written into the frame buffer (Section 3.3).

As input, the pipeline takes textured polygonal geometry. It requires a unique identifier per scene object. To increase rendering performance, we represent each numerical object ID as a per-vertex attribute of the specific object mesh. Under the assumption that static geometry is used, this procedure enables geometry batching [9] or streaming without modifying the proposed rendering pipeline. Therefore, the presented approach is suitable especially for real-time rendering of geometrical complex 3D scenes, since the geometry is rasterized only once.

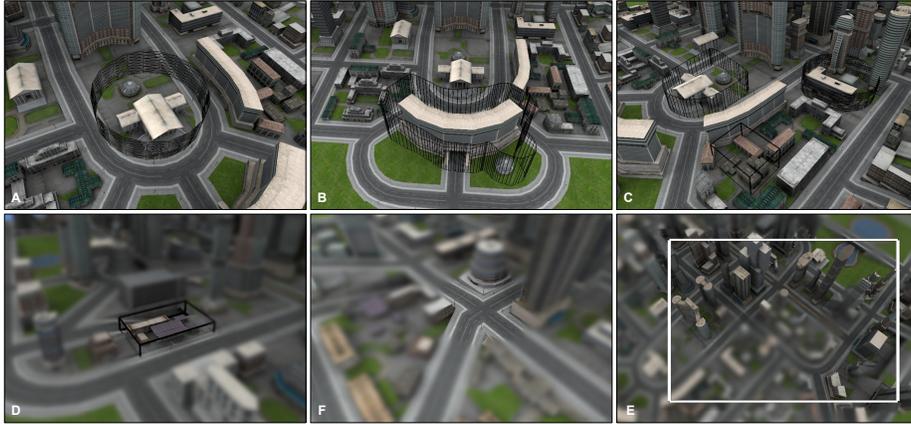


**Fig. 4.** Image-based representations (scene textures) of the 3D scene that are required by the highlighting pipeline: color (A), depth (B), color coded object IDs (C) and mask (D) values are derived within a single rendering pass.

### 3.1 Image Generation

Image generation represents the first step in the highlighting pipeline and enables the separation of focus and context regions. During a single rendering pass, image-based representations (at viewport resolution) of the 3D geometry (scene textures) and a mask texture is created (Fig. 4). Therefore, render-to-texture (RTT) in combination with fragment shaders and multiple render targets (MRT) is used, which enables to write fragments into multiple raster buffers simultaneously. The generated mask texture (Fig. 4.D) contains focus regions (white) and context regions (black). Our framework distinguishes between two methods for representing the input for the mask generation step:

**Scene Objects:** If complete scene objects (e.g., buildings) are selected for highlighting, their fragments are taken as input for generating the mask texture (Fig. 4.D). Their respective object IDs are encoded in an ID buffer (Fig. 4.C). This approach works only for highlighting complete objects (meshes). If it is required to highlight only parts of a single object or multiple objects, as well as highlighting regions of 3D scene that have no geometrical representation, proxy objects have to be used.



**Fig. 5.** Application of proxy-geometry for defining focus regions (cylindrical (A), free-form (B), multiple proxies (C)). D: Application examples of semantic depth-of-field using a 3D magic lens (D and E) and a 2D viewport lens (F) displayed with a white frame.

**Proxy Objects:** These objects are presented by additional polygonal geometry that is located in the 3D scene or on the viewport. They can operate as proxy for objects which screen size would be too small (e.g., street furniture), thus, a highlighting effect would hardly be perceivable by a user. Such a proxy object can be generated automatically, i.e., if the screen area of the projected bounding representation (e.g., AABB) an object is below a threshold, proxy geometry with a sophisticated on-screen size is created. Another use case for proxies is the partially highlighting of objects or the highlighting of routes affecting multiple objects. Here, the proxy geometry is usually created manually by the user (Fig. 5.A-C), e.g., by directly painting the proxy shapes on the viewport. The resulting line segments are converted to a polygonal representation, which is then used as input for the mask generation step. Later on, the user can modify the position, scale, rotation of a proxy interactively.

Note that the color and depth value of proxy objects are not written into the color and respective depth texture. Despite approximations of small objects, proxy shapes can be used for implementing Magic Lenses [21]. Our system mainly distinguished between 2D and 3D lenses. 2D lenses are located in screen space and move with the virtual camera. They can be used to implement auto-focus features as described in [22, 23] (Fig. 5.F). The proxy geometry of 3D lenses is placed in the 3D scene and does not align with the virtual camera (Fig. 5.D + E).

### 3.2 Mask Processing

After the mask texture is created, the mask processing step is performed. It basically enables the creation of smooth transition between the focus and context areas within the mask texture. The mask processing is implemented using RTT in combination with multi-pass rendering [24]. We basically apply two different processing algorithms: jump-flooding and convolution filtering. The jump-flooding algorithm [25] is used to perform distance transforms between focus and context areas. If only a small dilation of the input mask is required, e.g., for creating the glow outline, we apply convolution filtering instead. The final value of the mask can be controlled by a global drop-off function. It weakens or exaggerates the previous results from jump flooding or convolution filtering.

### 3.3 Application of Highlighting Techniques

After the processing of the mask texture is finished, the final pipeline phase applies the respective focus- or context-base style variance and outline highlighting techniques at a per-fragment level, using fragment shader.

Therefore, the color values in the scene texture (Fig. 4.A) are used to apply color overlay or vignetting based on the values of the mask texture. Given the object ID texture (Fig. 4.C), each object can be highlighted using a different technique. In general, the intensity of an effect, e.g., blur or opaqueness of a highlighting color, is controlled by the values of the mask texture. To implement SDOF we apply convolution filtering with different kernels in combination with multi-pass rendering. The trade-off between rendering speed and output quality can be controlled by the choice of the filter kernel. Gaussian blur requires more rendering passes than a box filter, but delivers a better blur quality.

Subsequently, the resulting texture of this previous step is applied to a screen-aligned quad that is then rendered into the frame buffer. The stored depth values (Fig. 4.B) are also written into the frame buffer. Finally, the glyph-based highlighting techniques, such as bounding boxes or arrows are applied to the frame buffer, by using standard forward rendering.

## 4 Results and Discussion

This section discusses the presented highlighting techniques and their implementation using different application examples, by providing a comparison, as well as ideas for possible future work. We tested our rendering techniques using data sets of different geometrical complexity: the generalized model of Berlin comprised 1,036,322 faces, the model of the Grand Canyon 1,048,560 faces, and the artificial 3D city model contains 34,596 faces.

The performance tests are conducted using a NVIDIA GeForce GTX 285 GPU with 2048 MB video RAM on a Intel Xeon CPU with 2.33 GHz and 3 GB of main memory. The 3D scene geometry was not batched and no view frustum and occlusion culling are applied. We are able to render all depicted scenes in

real-time, i.e., within the range of 12-32 frames-per-second. The performance of the presented image-based approach is geometry-bounds by the geometrical complexity of the 3D scene and fill-limited by the number of mask-processing passes, which have to be performed at the viewport resolution.

#### 4.1 Application Examples

Highlighting as a basic technology has various fields of application. Figure 6 shows possible application examples for the presented rendering techniques within 3D GeoVE. Besides the highlighting of single objects, we focus on route highlighting and the visualization of computational results.

**Route and Landmark Highlighting:** Fig. 6.A shows a combination of context- and focus-based style variance techniques applied to a generalized version of a virtual 3D city model of Berlin [3]. Here, a vignetting technique is used to highlight the route and color-highlighting is applied to the start and end position of the route. Additional, important landmarks that are nearby the route are highlighted in orange.

**Visualization of Computational Results:** Certain highlighting variants, such as color or glyph-based techniques, can be used for information visualization within 3D GeoVE. Fig. 6.B shows a possible visualization of a data base query result. A color-overlay technique is used to categorize a number of buildings. Here, the color blended over the facade textures can be used to encode specific data values.

**Object Highlighting:** Fig. 6.C shows the application of the focus-based style variance operator to the visualization of digital cultural heritage artifacts. The findings of a basement are highlighted in yellow to delimit it from the remaining artifacts, which base color is orange. Additional to coloring, we applied Gooch shading [26] and unsharp masking the depth buffer [27] to improve the perceptibility of the scene objects.



**Fig. 6.** Application examples of different highlighting techniques applied to 3D GeoVE. A: Highlighting as navigation aid for routing purposes within a generalized virtual 3D city model of Berlin. B: Using coloring for building categorization in an artificial virtual city model. C: Highlighting of finding groups within an interactive digital cultural-heritage application.

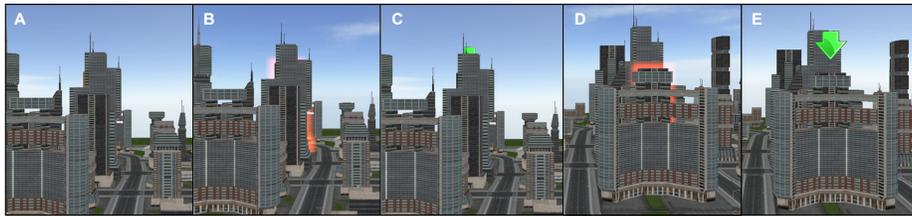
## 4.2 Comparison of Highlighting Techniques

Fig. 7 shows a comparison of style-variance, outline, and glyph-based highlighting techniques with respect to object occlusion and depth-cues. Although color- and glyph-based highlighting techniques are well established techniques to put focus on objects, they suffer from disadvantages: in the case of color highlighting, the appearance of the building is altered because the facade is dyed. As a result, important information, such as building color or texture details, become hardly recognizable. Further, in the field of city model analysis, the color attribute is often used to express a membership to semantic groups. Hence, color highlighting becomes unfeasible. Another critical problem in 3D virtual environments is occlusion. For example, if the POI is covered by another building, the viewer does not get any visual hints to guide his or her attention (Fig. 7.A).

In contrast to that, glyphs neither change texture information nor the buildings appearance. Instead, an additional geometric feature (glyph) is introduced atop of the object, e.g., in the form of a billboard. The size and the position of glyphs can be adapted dynamically, to avoid occlusion and ensure visibility (Fig. 7.C). One disadvantage of this method is missing depth cues, due to missing scale hints. If the POI is occluded by additional geometry and the scene is displayed in a pedestrian view, the user can hardly distinguish to which object the glyph belongs to (Fig. 7.E).

Using an context-based or outline highlighting technique as a method of object emphasis seems to be a promising approach. First, no relevant appearance parameters of the objects are changed. Second, even if objects are partly or completely occluded, an outline or glow can still be recognized to a certain degree (Fig. 7.B). Problems of mapping, as described above, are reduced because the outline can be seen as a propagated building silhouette and is, therefore, view invariant as well as supports an acceptable depth cue (Fig. 7.D). One drawback of huge opaque outlines is that they can reach into other objects, which may lead to an unwanted change of facade information.

The application of SDOF to 3D GeoVE exhibits a number of problems. If applied to virtual 3D city model the user is often distracted while trying to focus the blurred areas. In the case of 3D landscape models, viewed from a birds-eye



**Fig. 7.** Comparison of style-variance, outline, and glyph-based highlighting techniques with respect to occlusion (A-C) and the objects depth cue (D and E).

perspective, this effect appears less stronger. However, semantic depth-of-field fails if the screen size of a highlighted object is too small.

### 4.3 Challenges for Future Work

There are a number of possibilities for extending the presented work. We strive to extend the design space of highlighting by using NPR techniques for the stylization of 3D GeoVE [28]. Of particular interest is the question, if the visual differences between photo-realistic and NPR techniques are sufficient for the application as highlighting technique. We further like to investigate the computation of specific highlighting color sets, i.e. given the image-based representation of a virtual scene, what set of colors have the maximal visual differences, and how can they be computed automatically?

However, a user study is required to determine the effectiveness of the presented techniques with respect to certain user and visualization tasks. Based on this study, we plan the extension of the proposed framework by an automatic approach for the view-dependent selection of highlighting techniques, e.g., based on the distance between object and the virtual camera, or based on the angle with respect to the virtual terrain model.

## 5 Conclusions

This paper presents a framework for applying highlighting techniques to multiple objects in 3D GeoVE of high geometrical complexity. We further present rendering techniques to perform real-time. Further, a GPU based implementation has been presented that enables the image synthesis for various highlighting techniques at interactive frame rates. Furthermore, different highlighting-techniques have been compared with respect to 3D GeoVE. It has been shown, that the capabilities of outline- and context-based style variant techniques support the preemptive perception, while dealing with disadvantages of other highlighting techniques. Our framework can easily be integrated into existing software products, so it could be a promising addition to existing focus+context visualizations.

## Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) as part of the InnoProfile research group "3D Geoinformation" (www.3dgi.de). We like to thank Tassilo Glander for providing the generalized virtual 3D city model of Berlin.

## References

1. Björk, S., Holmquist, L.E., Redström, J.: A framework for focus+context visualization. In: Proc. of INFOVIS '99. (1999) 53
2. Cockburn, A., Karlson, A., Bederson, B.B.: A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys* **Vol. 41**(1) (2008) 1–31
3. Glander, T., Döllner, J.: Abstract representations for interactive visualization of virtual 3d city models. *Computers, Environment and Urban Systems* **33**(5) (2009) 375–387
4. Omer, I., Goldblatt, R., Talmor, K., Roz, A.: Enhancing the Legibility of Virtual Cities by Means of Residents Urban Image: a Wayfinding Support System. In: *Complex Artificial Environments - Simulation, Cognition and VR in the Study and Planning of Cities*. Springer Berlin Heidelberg (2006) 245–258
5. Jobst, M., Döllner, J.: 3d city model visualization with cartography-oriented design. In Schrenk, M., Popovich, V.V., Dirk Engelke, P.E., eds.: *REAL CORP.* (May 2008) 507–515
6. Bogdahn, J., Coors, V.: Using 3d urban models for pedestrian navigation support. In: *GeoWeb 2010.* (July 2010)
7. Robinson, A.C.: Highlighting techniques to support geovisualization. Technical report, GeoVISTA Center, Department of Geography, The Pennsylvania State University (2006)
8. Trapp, M., Schneider, L., Holz, N., Döllner, J.: Strategies for visualizing points-of-interest of 3d virtual environments on mobile devices. In: *6th International Symposium on LBS & TeleCartography*, Springer Berlin Heidelberg (2009)
9. Akenine-Möller, T., Haines, E., Hoffman, N.: *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA (2008)
10. Cole, F., DeCarlo, D., Finkelstein, A., Kin, K., Morley, K., Santella, A.: Directing gaze in 3D models with stylized focus. *Eurographics Symposium on Rendering* **Vol. 17** (2006) 377–387
11. Bruckner, S., Gröller, M.E.: Style transfer functions for illustrative volume rendering. *Computer Graphics Forum* **26**(3) (September 2007) 715–724 was awarded the 3rd Best Paper Award at Eurographics 2007.
12. Kosara, R., Miksch, S., Hauser, H.: Semantic depth of field. In: *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, Washington, DC, USA, IEEE Computer Society (2001) 97
13. Kosara, R., Miksch, S., Hauser, H., Schrammel, J., Giller, V., Tscheligi, M.: Useful properties of semantic depth of field for better f+c visualization. In: *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, Aire-la-Ville, Switzerland, Eurographics Association (2002) 205–210
14. Nienhaus, M., Döllner, J.: Edge-enhancement - an algorithm for real-time non-photorealistic rendering. *International Winter School of Computer Graphics, Journal of WSCG* **11**(2) (2003) 346–353
15. P.Hermosilla, Vzquez, P.: Single pass gpu stylized edges. In Sern, F., Rodrguez, O., Rodrguez, J., Coto, E., eds.: *IV Iberoamerican Symposium in Computer Graphics (SIACG)*. (2009)
16. O'Rorke, J., James, G.: Real-time glow. *Gamasutra* (May 2004)
17. Fuchs, G., Kreuzeler, M., Schumann, H.: Extended focus & context for visualizing abstract data on maps. In: *CODATA Prague Workshop on Information Visualization, Presentation, and Design.* (March 2004)
18. Kessenich, J.: *The OpenGL Shading Language*. The Khronos Group Inc. (2009)

19. Saito, T., Takahashi, T.A.: Comprehensible rendering of 3-d shapes. SIGGRAPH Comput. Graph. **Vol. 24**(4) (1990) 197–206
20. Eissele, M., Weiskopf, D., Ertl, T.: The  $g^2$ -buffer framework. In: Tagungsband SimVis '04, Magdeburg. (2004) 287–298
21. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., Deroose, T.D.: Toolglass and magic lenses: The see-through interface. In: SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, New York, USA, ACM Press (1993) 73–80
22. Hillaire, S., Lcuyer, A., Cozot, R., Casiez, G.: Depth-of-field blur effects for first-person navigation in virtual environments. IEEE Computer Graphics and Applications **Vol. 28**(6) (2008) 47–55
23. Hillaire, S., Lécuyer, A., Cozot, R., Casiez, G.: Using an eye-tracking system to improve camera motions and depth-of-field blur effects in virtual environments. In: VR, IEEE (2008) 47–50
24. Göddeke, D.: Playing ping pong with render-to-texture. Technical report, University of Dortmund, Germany (2005)
25. Rong, G., Tan, T.S.: Jump flooding in gpu with applications to voronoi diagram and distance transform. In: I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games, New York, USA, ACM (2006) 109–116
26. Gooch, A., Gooch, B., Shirley, P., Cohen, E.: A non-photorealistic lighting model for automatic technical illustration. In: SIGGRAPH. (1998) 447–452
27. Luft, T., Colditz, C., Deussen, O.: Image enhancement by unsharp masking the depth buffer. ACM Transactions on Graphics **25**(3) (jul 2006) 1206–1213
28. Döllner, J., Buchholz, H., Nienhaus, M., Kirsch, F.: Illustrative visualization of 3d city models. In Erbacher, R.F., Roberts, Jonathan C. and Gröhn, M.T., Börner, K., eds.: Visualization and Data Analysis. Volume 5669 of Proceedings of the SPIE., International Society for Optical Engine (SPIE) (2005) 42–51