Towards Service-Oriented, Standards- and Image-Based Styling of 3D Geovirtual Environments

Dieter Hildebrandt
dieter.hildebrandt@hpi.uni-potsdam.de

The server-side, service-oriented rendering of massive *3D geovirtual environments* (3DGeoVEs) has the potential to enable users with lightweight clients to access high quality, image-based visual representations of the environments without having to download massive amounts of geodata. *Image post-processing* (IPP) is a well-known concept that allows for decoupling the process of generating an image from applying enhancement processes to already generated images. Moreover, several visual effects are implemented most efficiently and effectively as image post-processes. However, so far no proposals exist for providing image post-processing of 2D images of projective views of visual representations of 3DGeoVE in a SOA and based on standards.

In this paper, we investigate how IPP functionality and the functionality of styling of visual representations of 3DGeoVE based on IPP can be provided in a SOA based on standards. First, we introduce the concepts of IPP and styling. Then, we present an analysis of different characteristics of styling and IPP and design dimensions relevant when building a system for styling and IPP. From the analysis, we derive a set of requirements for a concept and system providing IPP and styling functionality based on IPP. We present a preliminary concept for a system meeting the identified requirements and report on initial implementation and evaluation results.

1 Introduction

The server-side, service-oriented rendering of massive *3D geovirtual environments* (3DGeoVEs) has the potential to enable users with lightweight clients to access high quality, image-based visual representations of the environments without having to download massive amounts of geodata. *Image post-processing* (IPP) [2] is a well-known concept that allows for decoupling the process of generating an image from applying enhancement processes to already generated images. Moreover, several visual effects are implemented most efficiently and effectively as image post-processes. In a *service-oriented architecture* (SOA), providing the functionality of image post-processors as dedicated, loosely coupled services as an application of the principle of separation of concerns offers several advantages. These services easily can be reused and recomposed with further services to form different distributed applications, the increased modularity has the potential to improve the maintainability and flexibility of the resulting systems, and existing applications and services can be extended to take advantage of image post-processing functionality without having to implement the functionality themselves. Providing this functionality based on open standards such as approved by the W3C and the *Open Geospatial Consortium* (OGC) [7]) facilitates building effective, open and interoperable systems. However, so far no proposals exist for providing image post-processing of 2D images of projective views of visual representations of 3DGeoVE in a SOA and based on standards.

In this paper, we investigate how IPP functionality and the functionality of styling of visual representations of 3DGeoVE based on IPP can be provided in a SOA based on standards. First, we introduce the concepts of IPP and styling. Then, we present an analysis of different characteristics of styling and IPP and design dimensions relevant when building a system for styling and IPP. From the analysis, we derive a set of requirements for a concept and system providing IPP and styling functionality based on IPP. We present a preliminary concept for a system meeting the identified requirements and report on initial implementation and evaluation results.

This paper is structured as follows. In Section 2, we introduce the fundamentals of IPP and styling. We present an analysis of characteristics and design dimensions of IPP and styling and a set of derived requirements in Section 3. The preliminary design of the concept is presented in Section 4, initial implementation and evaluation results in Section 5. Section 6 concludes this paper with a summary, conclusions and next steps.

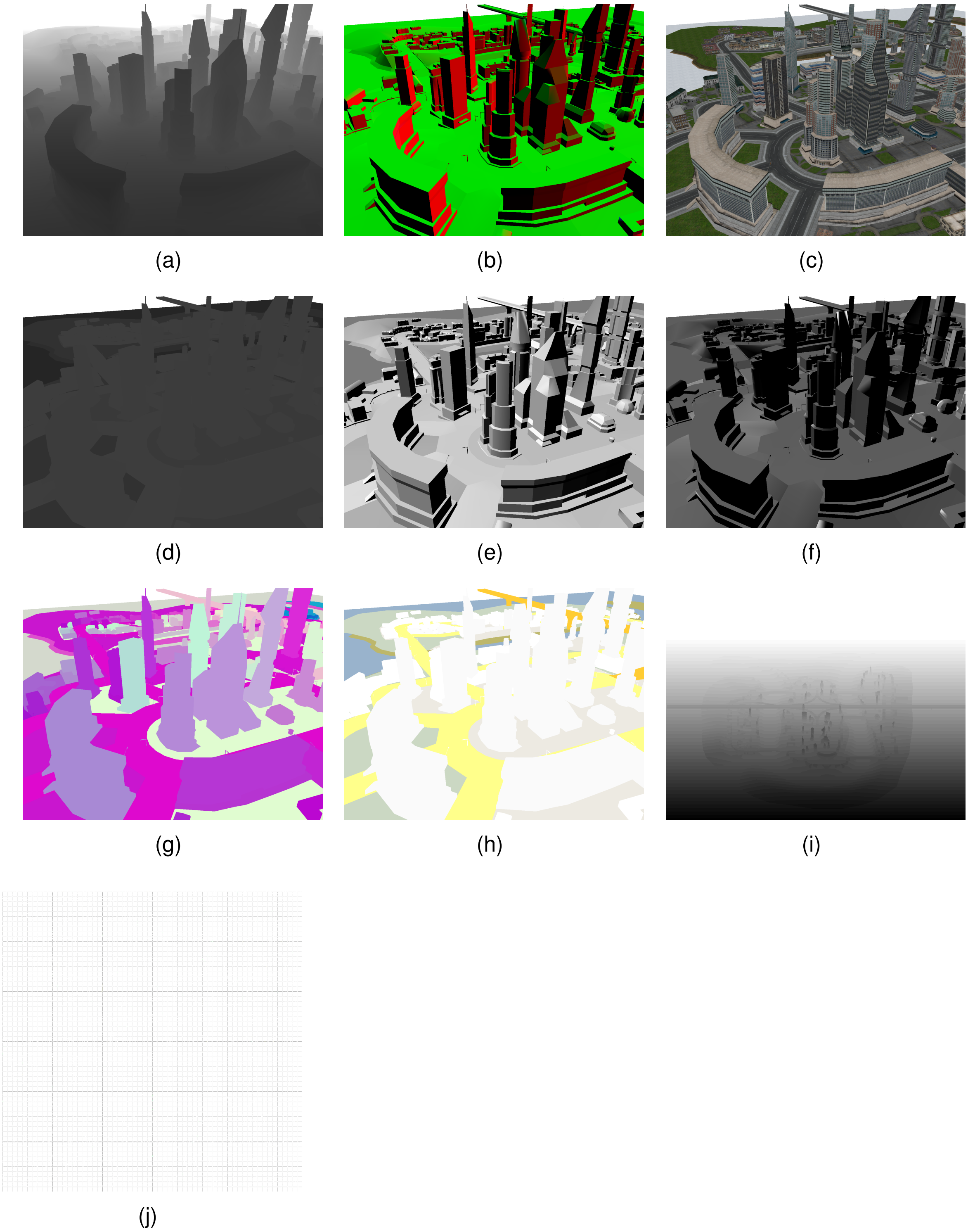
2 Fundamentals

2.1 Image Post-Processing

Digital image processing encompasses processes whose inputs and outputs are images and, in addition, encompasses processes that extract attributes from images, up to and including the recognition of individual objects [9]. In the context of computer graphics, performing image processing after rendering is called *image post-processing* (*IPP*) [2]. We define *IPP effect* as a unit of image post-processing with a specific functionality and purpose. IPP effects can be efficiently implemented on current graphics processing units (GPUs) [2]. The concept of IPP can be applied for different applications. In this paper, we focus on the use of IPP for styling 2D images of projective views of 3DGeoVE. On the contrary, in general, styling can be implemented without the use of IPP.

As input and output for IPP, 2D images representing G-buffer [23] can be used. For a projective view of a visual representation of a 3DGeoVE, a G-buffer encodes per pixel a specific type of information such as color, depth, normal, and material properties. Figure 1 depicts example G-buffer containing different, exemplary types of information.

In this context, several characteristics and benefits of IPP are important. IPP allows for decoupling the process of generating an image from applying enhancement processes to already generated images. Moreover, several visual effects are implemented most efficiently and effectively as image post-processes. The complexity of representing a perspective view in an image and applying IPP to the image depends



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 1: Examples of different types of information encoded per pixel in a G-buffer: (a) depth, (b) normal, (c) albedo (color texture), (d) ambient lighting, (e) diffuse lighting, (f) specular lighting, (g) object ID, (h) colorCode (encoding semantics of objects with a color), (k) shadowMap (depth of scene as seen from a light source), and (i) projective texture (a texture to be applied to the scene via projective texturing).

only on the image dimensions and not the complexity of the scene [16]. Furthermore, images as inputs and outputs for IPP are conceptually simple, robust, commonly used and supported. Additionally, image formats exist (e.g., JPEG, PNG) that are standardized, commonly used and supported, and storage and processing efficient. Using the G-buffer concept [23], multiple information layers of a 3D model (e.g., 3D position, normal, color, and object ID of surface elements) can be encoded into 2D images. Thus, images can be used as an alternative representation for 3D models that sample 3D models in a discreet and multidimensional way and reduce the diversity and heterogeneity of their original representations (e.g., points, triangles, NURBS, voxel) to a simpler, unified representation.

2.2 Styling

According to the OGC, the importance of the visual portrayal of geographic data cannot be overemphasized. The skill that goes into portraying data (whether it be geographic or tabular) is what transforms raw information into an explanatory or decision-support tool. Fine-grained control of the graphical representation of data is a fundamental requirement for any professional mapping community [19, 21]. In this context, styling can be defined as the mapping of data to geometry and/or appearance attributes.

The are efforts from the OGC to standardize the styling of 2D and 3D portrayal of geospatial data in a SOA. *Symbology Encoding (SE)* [21] represents a language for defining rendering parameters for specific features and coverages. SE describes the symbolizer (line, polygon, point, text, raster) to use for rendering the feature geometry, which appearance parameters to consider, and for which scale this styling is applicable. The *Styled Layer Descriptor (SLD) Profile for WMS* [19] allows for user-defined styling. Together with the *GetMap* request an SE-encoded SLD description is transmitted inline or as URL reference. Therefore, the *Web Map Service* (WMS) [8] interface is extended for retrieving the feature types of a layer. In contrast to styling the 2D portrayal from a WMS, until now no OGC standard exists that defines the styling of 3D portrayal from a *Web 3D Service* (W3DS) [24] and the *Web View Service* (WVS) [12,13]. Nevertheless, Haist et al. [14] and Neubauer et al. [22] propose separate extensions for the SLD and SE for 3D portrayal.

Within the visualization pipeline [11], functionality for styling can be conceptually located in the filtering, mapping, rendering or even after the rendering stage. Widely-used is styling in the mapping stage (*M-styling*). The already introduced concepts of SLD and SE for 2D and 3D portrayal in the context of standardization apply styling in the mapping stage. However, styling can also applied in or even after the rendering stage (*R-styling*). For styling 2D images of projective views, IPP can be applied in or after the rendering stage.

3 Analysis

In this Section, we analyze different characteristics of styling and IPP and design dimensions relevant when building a system for styling and IPP. From the analysis, we derive a set of requirements for a concept and system providing IPP and styling functionality based on IPP.

Granularity of IPP Effects On the lowest level, images can be processed by iterating in one pass over all pixel of the target image and applying an algorithm for calculating for each pixel the resulting value from the input images (e.g., alpha blending of two input images). On a higher level, specific effects can only be calculated by combining several basic passes and using outputs of passes as inputs for other passes (e.g., image abstraction by structure adaptive filtering [18]).

Programmable vs. Configurable When the processing of IPP is offered as a service, it can be programmable and/or configurable. The processing is *programmable* if a service consumer can provide the service with an algorithmic description of the effect that service provider is supposed to execute. The focus is on offering processing as a service. The *Web Coverage Processing Service* (WCPS) [4] is an example of such a service that offers processing of coverages. The processing is *configurable* if a service provider already implements and offers one more IPP effects. The service consumer can choose from the offered effects and can configure their behavior. In this case, the service provider offers IPP effect implementations as well as processing as a service.

IPP Effect Composition (Service Interface) Services that provide the functionality of IPP and IPP-based styling must provide an interface to service consumers that allows accessing this functionality. The interface must allow specifying which IPP effects in what configuration are to be applied to what geospatial data and what output images are required.

For specifying how input images are transformed by a set of IPP effects into output images, the concept of *data flow graphs* (DFG) [27] is commonly used. Nodes represent IPP effects and directed edges the flow of data between the nodes. The resulting graphs are *directed acyclic graphs* (DAG) with the tendency to form a tree with the inputs as leafs and the result as root node. This concept is used in commercial software products such as *Adobe Pixel Bender* [1] and *Apple Quartz Composer* [3].

For styling 2D portrayal in the mapping stage (M-styling), the OGC proposes the concepts of SLD and SE. In essence, these concepts allow a service consumer to specify which symbolizer from a predefined list in what configuration is to be applied to which features (selected as layers of features with a set of predefined selection operators). As a simpler alternative, image generating services (such as the WMS) allow choosing a style from a predefined list of styles for each layer that is selected for portrayal.

Semantic-based and Selective Styling and Application of IPP Efficient and effective communication of geospatial information in 3DGeoVEs typically requires that for subsets of the geospatial data specific, adequate representations are choosen. The mapping of geospatial data to visual representations is accomplished by styling. It follows that, generally, effective styling requires that different styling can be applied to different subsets of the geospatial data. As an example, to guide the viewers gaze to a focus area in a visual representation, the focus area could be visually highlighted and represented in detail (e.g., by applying photorealistic rendering techniques) whereas the surrounding context area would be represented abstracted and with less detail (e.g., by selectively applying non-photorealistic rendering techniques).

Geospatial data is commonly organized in *features* and collections of features called *layers*. When using a semantic data model such as *CityGML* [10], features are enriched with semantic information. Layers of features or individual features can be selected explicitly or rule-based (e.g., all features within a buffer region from a road or with a specific semantics or thematic attribute) and styled individually. For selecting features from collections of features, the OGC proposes a dedicated filter language [28]. When applying IPP for styling, additional methods for selection become viable. Applying IPP to the whole scene or on spatial regions across feature and layer boundaries can be accomplished efficiently and effectively.

Architecture From an architectural perspective, in a SOA, the functionality of IPP and IPP-based styling can be located in three places:

1. Integrated with the service that provides images as input to the IPP (e.g., WVS),
2. Provided as a dedicated service that receives input images from other services or the calling service consumer and provides the service consumer with output images, or
3. Integrated with the interaction service [15] that a human user directly interacts with.

From the analysis we derive the following set of requirements for a concept and system that provides IPP and styling based on IPP in a SOA based on standards:

- Standardization: The service interface and employed data models and encodings in the interface should adhere to or build on existing open standards.
- Granularity of IPP Effects: Service consumers can specify IPP on low-level (e.g., single-pass IPP effect) and high-level granularities (e.g., multi-pass IPP effects).
- Programmable vs. Configurable: Service providers can provide IPP effect implementations and their processing to service consumers. Service consumers can algorithmically specify IPP effects, and transmit the specification to the service provider for execution.
- Architecture: The IPP functionality must be accessible as a dedicated service and integrated in a image rendering service (i.e., the WVS). Integrating IPP in the interaction service is not recommended unless unavoidable to keep the interaction service as lightweight as possible and the IPP functionality reusable as a service.

- Selective Application of IPP: Subsets of geodata for styling must be selectable on the level of parts of space and features, individual features, layer, whole scene.
- IPP Effect Composition (Service Interface): Services providing IPP functionality must provide an interface based on data flow graphs. Services providing functionality for styling based on IPP must provide an interface that is based on SLD and SE.

4 Design

In this Section, we briefly sketch the preliminary concept for services providing the functionality of IPP and IPP-based styling.

A service consumer can specify IPP with a data flow graph (DFG) expressed in XML. A DFG is directly suitable for being processed by the service provider. For specifying IPP-based styling, the current proposals for SLD and SE are extended to support the specific characteristics of IPP-based styling. The SLD/SE-based styling description is more abstract than the DFG-based description, while the latter is more expressive. We assume that the SLD/SE-based styling description can be mapped to a DFG-based description that then can be directly processed.

Service providers can offer a predefined set of IPP effect implementations that can be referenced in DFG-based and SLD/SE-based IPP descriptions. Additionally, service providers allow users to provide their own executable code for IPP effects. We plan to evaluate both the open standards *OpenGL Shading Language* (GLSL) and *OpenCL* for this purpose.

The IPP and IPP-based styling is implemented as a library that is used for implementing a dedicated service and for integration in the WVS. The dedicated service is based on the OGC's generic *Web Processing Service* (WPS) [25] proposal.

We have identified a first collection of general-purpose, reusable IPP effects that enable the composition of web view services for 3DGeoVEs operating on G-buffer images. The collection includes:

- 3D image synthesis effects generate the base G-buffer.
- Shadow mapping synthesize effects generate and apply shadow maps [29] in screen-space.
- Ambient occlusion synthesis effects generate approximated ambient light intensities in screen-space (SSAO, [26]).
- Non-photorealistic image processing effects emphasizes edges and remove detail from surfaces [2, 18].
- Depth-of-field effects infiltrate an artificial object-based focus area in the view [2].
- Projective texture effects superimpose projective textures on the given view [2].
- Highlighting effects emphasize the silhouette of selected objects.

- Generic G-buffer and image converter, blending, and convolution operators.

5 Implementation and Evaluation

In this Section, we briefly report on the initial implementation and evaluation of the concept for services providing the functionality of IPP and IPP-based styling.

As a basis, we implemented a library for specifying and executing IPP based on data flow graphs in C++. Second, we implemented a set of exemplary low-level and high-level IPP effects in C++, OpenGL and GLSL. A list of the implemented IPP effects is presented in Figure 2.

We implemented ambient occlusion synthesis effects that generate approximated ambient light intensities in screen-space (SSAO, [26]). For evaluation, we implemented two different techniques: [17, 20] and [5, 6].

We implemented non-photorealistic image processing effects that emphasize edges and remove detail from surfaces [18]. For evaluation, we implemented two different techqniues. The first NPR technique "NPR1" is based on image abstraction by structure adaptive filtering [18], and requires only one color image as input.

The second NPR technique, "NPR2 ColorBlend and EdgeEnhance", requires as input two color images, depth, and diffuse lighting. The two color images are intended to be both appearance representations of the 3D scene. However, the first one is expected to be a more abstract representation than the second one. In our implementation and evaluation, the first color image represents a color coding of semantics of the scene objects. Each pixel in the image is assigned a color that is based on the semantics of the object that the pixel belongs to. We used a color scheme that was inspired by common 2D web mapping applications. For the second color image, in our evaluation, we used the output of the first technique NPR1 based on image abstraction. The two color code images are then blended (via mix) based on the distance of each fragment from the virtual camera. Optionally in this blend, the second image is modulated by the first image to preserve its effect. The effect of the blending is that the more detailed representation dominates fragments near the virtual camera while, optionally, the color coding is still visible. This representation is gradually blended with the more abstract representation that dominates fragments farther from the camera. Subsequently, the diffuse lighting is applied to the new appearance image that results from the blend. Finally, edges in the image are detected using discontinuities in the depth and color code images. Found edges are enhanced by darkening respective fragments.

For evaluating the implementation, we performed experiments. We build a graph specifying a complex IPP (Figure 3) using the implemented IPP effects. As input to the graph, the ten G-buffers depicted in Figure 1 are used. Intermediate G-buffer processing results of the data flow graph are presented in Figure 4. The final result (color buffer) is displayed in Figure 5. Figure 2 reports on the processing time for the individual effects. The experiment is performed on a Core2Duo E6600 with 2.4Ghz, nVidia GTX 260. Measurements were performed on G-buffer resolutions of 800x600 and averaging the timings of 5.000 processing calls. The timings indicate that even complex

Category	Effect	Effect Mode/Subtype/Config	Avg Processing Time (ms)			View Independ.	Number of Passes
			Single	Accum	Accum.		
Integration	Composite	Depth and ObjectId Selector	0,26		0,26	x	1
	Blend	Add	0,22		0,22	x	1
		Hardlight	0,22			x	1
Abstraction	NPR1	Image Abstraction	16,80		16,80	o	6
	NPR2	ColorBlend and EdgeEnhance	2,77		2,77	o	1-2
Photorealism	SSAO	Mittring (half res)	3,88	4,74	10,05	o	3
		Mittring (full res)	5,60			o	3
		Bavoil (s=8, d=16)	9,72	12,71		o	3
		Bavoil (s=8, d=16, halfres)	5,30			o	3
		Bavoil (s=24, d=32)	32,63			o	3
		Bavoil (s=3, d=4)	3,20			o	3
	ShadowMapping	Hard Shadows	0,35		1,59	o	2
		PCSS	2,84			o	
Focus and Context	Highlight	Halo (r=2)	1,07	1,57	0,92	-	3
		Halo (r=7)	1,40			-	3
		Halo (r=20)	2,26			-	3
		Background Grey	0,27			x	1
		Brightness	0,30			x	1
		ColorOverlay	0,26			x	1
	Depth of Field		1,01		1,01	-	5
Augmentation	ProjectiveTexturing		0,40		0,40	x	1
Low Level Processing	ColorAdjust		0,22		0,22	-	1
	Convolution	Gauss (r=3)	0,64	1,06	0,85	x	2
		Gauss (r=10)	1,48			x	2
		Box (r=3)	0,75	1,19		x	2
		Box (r=10)	1,64			x	2
		Unsharp Masking (Laplace)	0,27	0,31		x	1
		Unsharp Masking (Sobel)	0,34			x	1
Environmental	Fog	*not implemented yet*					
	Rain	*not implemented yet*					
	Water	*not implemented yet*					

Figure 2: Overview of the implemented IPP effects, their average processing time (measured on Core2Duo E6600 with 2.4Ghz, nVidia GTX 260, resolution 800x600), view independence characteristic, and number of passes.

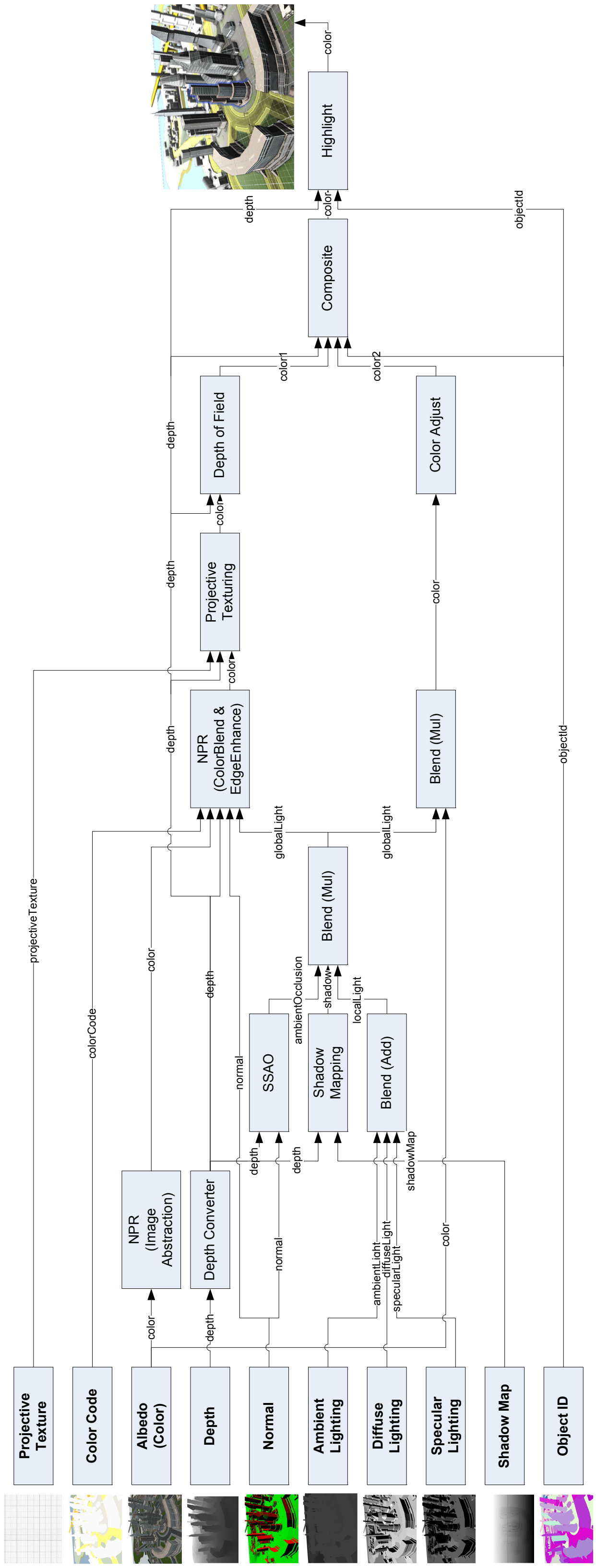
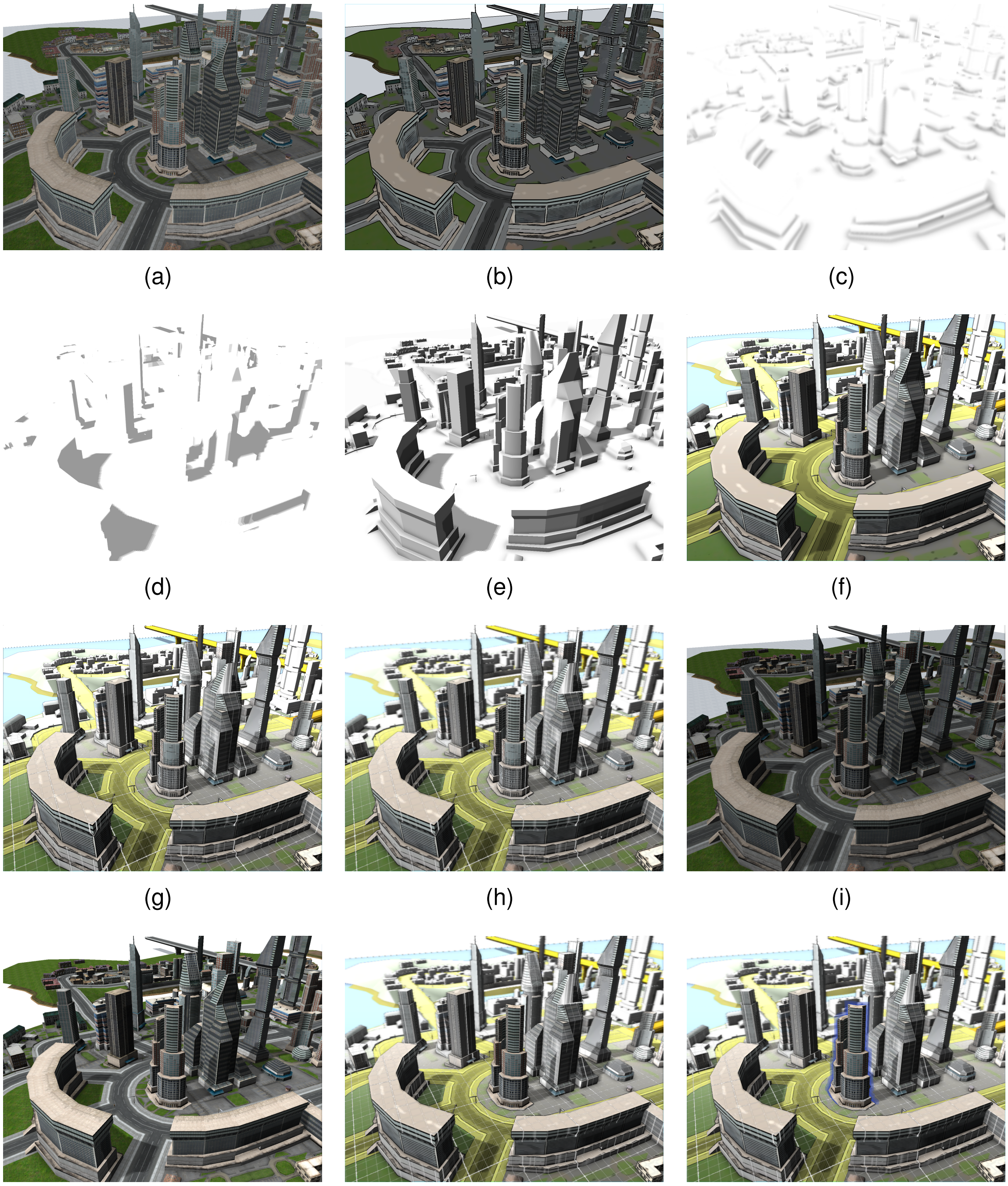


Figure 3: Data flow graph of a complex IPP effect composition using the implemented IPP effects build for the evaluation experiment.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 4: Intermediate G-buffer processing results of the data flow graph. (a) color as the most important input appearance data, (b) output of the NPR (Image Abstraction) node, (c) output of the SSAO node, (d) output of the Shadow Mapping node, (e) output of the first Blend (Mul) node, (f) output of the NPR (ColorBlend and EdgeEnhance) node, (g) output of the Projective Texturing node, (h) output of the Depth of Field node, (i) output of the second Blend (Mul) node, (j) output of the Color Adjust node, (k) output of the Composite node, (l) output of the Highlight node.

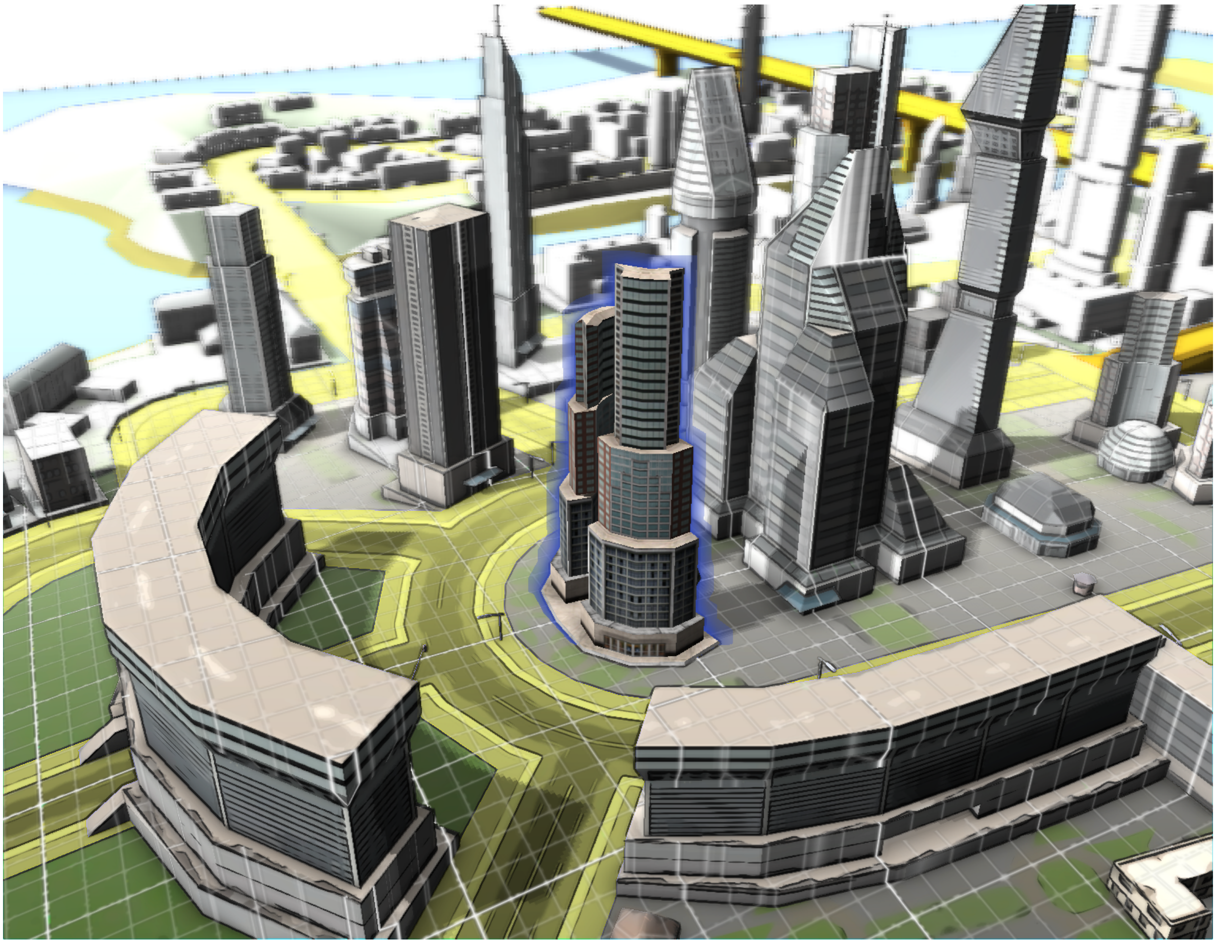


Figure 5: Final result (color buffer) of processing the data flow graph build for the evaluation experiment.

IPP-based styling can be performed at interactive frame rates on current consumer hardware. As expected, IPP effects have the tendency to take more time the more complex they are (in terms of algorithmic and memory access complexity per pixel) and the more passes they require. Surprisingly, the SSAO (screen-space ambient occlusion) effect turned out to be comparatively costly with respect to its contribution to the final result.

6 Summary, Conclusions, and Next Steps

In this paper, we investigated how IPP functionality and the functionality of styling of visual representations of 3DGeoVE based on IPP can be provided in a SOA based on standards. We introduced the concepts of IPP and styling and presented an analysis of different characteristics of styling and IPP and design dimensions relevant when building a system for styling and IPP. From the analysis, we derived a set of requirements for a concept and system providing IPP and styling functionality based on IPP. We presented a preliminary concept for a system meeting the identified requirements and reported on an initial implementation and evaluation results.

The first results indicate that IPP functionality and the functionality of styling of visual representations of 3DGeoVE based on IPP can be provided in a SOA based on standards. Furthermore, IPP-based styling promises to offer powerful ways of styling visual representations with interactive frame rates decoupled from the process of image generation.

My next steps include extending the SLD/SE for IPP, integrating IPP and IPP-based styling in the WVS and offering it as a dedicated service, and allowing service consumers to specify IPP effects by providing executable code.

References

[1] Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA. *Adobe Pixel Bender Developer's Guide*, 2010.

[2] Tomas Akenine-Möller, Eric Haines, and Natty Hoffman. *Real-Time Rendering*. A. K. Peters, Ltd., Natick, MA, USA, third edition, 2008.

[3] Apple Inc., 1 Infinite Loop, Cupertino, CA 95014. *Apple Quartz Composer User Guide*, July 2007.

[4] Peter Baumann, editor. *OGC Web Coverage Processing Service (WCPS), Version 0.0.3*. Open Geospatial Consortium Inc., July 2006.

[5] Louis Bavoil and Miguel Sainz. Image-Space Horizon-Based Ambient Occlusion. In Wolfgang Engel, editor, *ShaderX7 - Advanced Rendering Techniques*, pages 425–444. Charles River Media, 2009.

[6] Louis Bavoil, Miguel Sainz, and Rouslan Dimitrov. Image-Space Horizon-Based Ambient Occlusion. In *SIGGRAPH 2008 Talks*. ACM, New York, NY, USA, 2008.

[7] Open Geospatial Consortium. Open Geospatial Consortium (OGC) Website. URL, `http://www.opengeospatial.org/`, 2009. Accessed 16.10.2010.

[8] Jeff de la Beaujardiere, editor. *OpenGIS Web Map Server Implementation Specification, Version 1.3.0*. Open Geospatial Consortium Inc., March 2006.

[9] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall International, Upper Saddle River, N.J., third edition, 2008.

[10] Gerhard Gröger, Thomas H. Kolbe, Angela Czerwinski, and Claus Nagel, editors. *OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 1.0.0*. Open Geospatial Consortium Inc., August 2008.

[11] R.B. Haber and D. A. McNabb. Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In B. Shriver, G. M. Nielson, and L. Rosenblum, editors, *Visualization in Scientific Computing*, pages 74–93, Los Alamitos, 1990. IEEE Computer Society Press.

[12] Benjamin Hagedorn, Dieter Hildebrandt, and Jürgen Döllner. Towards Advanced and Interactive Web Perspective View Services. In *Developments in 3D Geo-Information Sciences*, Lecture Notes in Geoinformation and Cartography, Berlin, Heidelberg, November 2009. Springer.

[13] Benjamin Hagedorn, Dieter Hildebrandt, and Jürgen Döllner. *Web View Service Discussion Paper, Version 0.6.0*. Open Geospatial Consortium Inc., February 2010.

[14] Jörg Haist, Hugo Miguel Figueiredo Ramos, and Thorsten Reitz. Symbology Encoding for 3D GIS - An Approach to Extending 3D City Model Visualization to GIS Visualization. In *Urban Data Management Symposium*, pages 121–131, October 2007.

[15] Dieter Hildebrandt and Jürgen Döllner. Service-oriented, standards-based 3D geovisualization: Potential and challenges. *Journal on Computers, Environment and Urban Systems*, 34(6):484–495, November 2010. GeoVisualization and the Digital City - Special issue of the International Cartographic Association Commission on GeoVisualization.

[16] Dieter Hildebrandt, Benjamin Hagedorn, and Jürgen Döllner. Image-Based, Interactive Visualization of Complex 3D Geovirtual Environments on Lightweight Devices. In *7th International Symposium on LBS and Telecartography*, 2010.

[17] Vladimir Kajalin. Screen-Space Ambient Occlusion. In Wolfgang Engel, editor, *ShaderX7 - Advanced Rendering Techniques*, pages 413–424. Charles River Media, 2009.

[18] Jan Eric Kyprianidis and Jürgen Döllner. Image Abstraction by Structure Adaptive Filtering. In Ik Soo Lim and Wen Tang, editors, *EG UK Theory and Practice of Computer Graphics*, pages 51–58. Eurographics Association, 2008.

[19] Markus Lupp, editor. *Styled Layer Descriptor Profile of the Web Map Service Implementation Specification, Version 1.1.0*. Open Geospatial Consortium Inc., June 2007.

[20] Martin Mittring. Finding next gen: CryEngine 2. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, pages 97–121, New York, NY, USA, 2007. ACM.

[21] Markus Müller, editor. *Symbology Encoding Implementation Specification, Version 1.1.0*. Open Geospatial Consortium Inc., July 2006.

[22] Steffen Neubauer and Alexander Zipf. Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into the third Dimension - An Analysis of possible Visualization Rules for 3D City Models. In *Urban Data Management Symposium*, Stuttgart, Germany, October 2007.

[23] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-D shapes. *SIGGRAPH Computer Graphics*, 24(4):197–206, 1990.

[24] Arne Schilling and Thomas H. Kolbe, editors. *Draft for Candidate OpenGIS Web 3D Service Interface Standard, Version 0.4.0*. Open Geospatial Consortium Inc., 2010.

[25] Peter Schut, editor. *OpenGIS Web Processing Service, Version 1.0.0*. Open Geospatial Consortium Inc., June 2007.

[26] Perumaal Shanmugam and Okan Arikan. Hardware accelerated ambient occlusion techniques on GPUs. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, I3D '07, pages 73–80, New York, NY, USA, 2007. ACM.

[27] Michael A Shantzis. A Model for Efficient and Flexible Image Computing. In *SIGGRAPH 94 Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, volume 28, pages 147–154, New York, NY, USA, 1994. ACM Press.

[28] Peter Vretanos, editor. *OpenGIS Filter Encoding 2.0 Encoding Standard, Version 2.0.0*. Open Geospatial Consortium Inc., March 2010.

[29] Lance Williams. Casting curved shadows on curved surfaces. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, volume 12 of *SIGGRAPH '78*, pages 270–274, New York, NY, USA, 1978. ACM.