

Service-Based, Interactive Portrayal of 3D Geovirtual Environments

Benjamin Hagedorn

benjamin.hagedorn@hpi.uni-potsdam.de

This report summarizes my work in the HPI Research School on Service Oriented Systems Engineering. It motivates and points out the underlying research questions in the field of service-based 3D portrayal of large and complex 3D geovirtual environments, presents my approach to solving these questions, and demonstrates and highlights results by example. A focus is on my work during the past months.

1 Introduction

Continuously, massive geodata is acquired by authorities, municipalities, private companies, and public communities. This includes, e.g., digital elevation and terrain models, aerial images, other photogrammetric data, land use information, infrastructure data, or building footprints. This data can be used in various application areas, e.g., urban planning, thread response, disaster management, fleet management, tourism, or location marketing. Geodata in general represents a data class that includes heterogeneous information, which can differ, e.g., in data semantics, models, formats, or complexity. However, the spatial reference of geodata forms a foundation for combining this heterogeneous data and generating new knowledge about complex spatial structures, relationships, and processes.

To tap the full potential of geodata and adding value to it, applications and systems are required that provide methods and techniques for geodata management, provisioning, access, integration, analysis, and visualization that take into account the characteristics of this data, mainly its distributed storage, and massive data size. Interoperability represents one primary issue of such systems and techniques:

“Geographic interoperability” is the ability of information systems to 1) freely exchange all kinds of spatial information about the Earth and about the objects and phenomena on, above, and below the Earths surface; and 2) cooperatively, over networks, run software capable of manipulating such information. [1]

In the geo-domain, service-based systems have evolved as *the* approach for making distributed geodata accessible, processing this data, and presenting it to end users. Various consortiums and standardization bodies are engaged in the interoperable specification of geodata formats and services. Examples include the International Organization for Standardization (ISO), the Open Geospatial Consortium (OGC), the

Open Source Geospatial Foundation (OSGeo), and others. Additionally, large companies and vendors participate in the standardization processes, e.g., by providing de facto standards.

Geovisualization, i.e., the generation of visual representations dedicated for human perception, plays a major role within the process of geodata provisioning and in today's spatial data infrastructures. For this purpose, the Open Geospatial Consortium (OGC) proposes several portrayal services for 2D and 3D geodata. So far, there is only one widely used "workhorse", the Web Map Service (WMS), for generating 2D maps. Standards for visualizing 3D geovirtual environments (3DGeoVEs) such as virtual 3D city models and landscape models have not been elaborated to a similar degree.

From this, the research question raised **Q1**) of how to provide complex 3DGeoVEs in a service-based manner. As a first finding, we identified image-based 3D portrayal services as an effective means for integrating and presenting complex, heterogeneous geodata from distributed sources. A second research question was **Q2**) how to make this image-based presentation as efficient and interactive as possible, which includes, e.g., navigation and analysis capabilities. A third question was **Q3**) how to orchestrate image-based portrayal services and how to provide their integration into more complex portrayal scenarios, aiming at higher-level geovisualization services.

My approach for solving these questions is presented in the remaining sections of the report. Section 2 introduces the approach of service-based, image-based 3D portrayal. Section 3 presents my work for introducing interaction capabilities into an image-based 3D portrayal service; Section 4 addresses the same issue by introducing a smart navigation technique. Section 5 demonstrates the potentials of orchestrating image-based 3D portrayal services, and Section 6 summarizes my research work.

2 Approach: Image-Based 3D Portrayal Services

For portraying 3D geodata, two approaches have been presented by the OGC, the Web 3D Service (W3DS) and the Web Perspective View Service (WPVS).¹ – While the W3DS provides scene graphs that have to be rendered by the service consumer, the WPVS supports the generation of images that show 2D perspective views of 3DGeoVEs encoded in standard image formats. The key advantages of this image-based approach include low hardware and software requirements on the service consumer side due to service-side model management, 3D rendering, and moderate bandwidth requirements; only images need to be transferred regardless of the model and rendering complexity. Furthermore, it can be used with simple, lightweight clients without specific 3D rendering hardware because only standard viewing functionality is required such as provided by web browsers.

As part of the OGC web services initiative, phase 4, we investigated the applicability of service-based access to distributed two-dimensional and three-dimensional geodata from the GIS, CAD, and BIM domain; terrain models, aerial-images, and building models have been accessed via WMS and Web Feature Services (WFS) and have been

¹The W3DS has discussion status. The WPVS is an OGC-internal draft specification and represents the successor of the OGC Web Terrain Server discussion paper.

integrated by an interactive 3D viewer client, which also provides functionalities for building inspection and analysis. It turned out, that interoperability can be achieved for 3DGeoVEs based on a service-based architecture and based on 3D standards such as the Industry Foundation Classes (IFC) and CityGML [3]. As key contribution, the 3D viewer client integrates 2D and 3D GIS data as well as 3D CAD and BIM data at the visualization level in a lossless and seamless way by using 3DGeoVEs as a key concept and providing information retrieval and analysis functionalities.

Based on these findings, a high-level image-based 3D portrayal web service for the visualization and analysis of 3D building information models (BIM) had been developed [4]. This specialized WPVS integrates GIS, CAD, and BIM data at the visualization level; building information are mapped to components of building virtual 3D city models. High interoperability is ensured by providing the integration result as image to a service consumer; rendering styles can be configured by service operation parameters and different views for analysis purposes can be obtained.

3 An Interactive, Image-Based 3D Portrayal Service

The weaknesses of the current WPVS approach include the limited degree of interactivity of client applications. For example, it is not possible to navigate to a position identified in the image or to retrieve information about visible geographic objects.

To overcome these limitations, we propose the *WPVS++*, an extension of the OGC WPVS that is capable of augmenting each generated color image by multiple thematic information layers encoded as images. These additional image layers provide for instance depth information and object identity information for each pixel of the color image. Additionally, we propose operations for retrieving thematic information about presented objects at a specific image position, simple measurement functionality, and enhanced navigation support. This allows *WPVS++* clients to efficiently access information about visually encoded objects in images, to use that information for advanced and assistant 3D navigation, and thereby to increase the degree of interactivity.

In the following, the main functionalities of the proposed interactive *WPVS++* are described.

3.1 Image Layers

Besides color images, the *WPVS++* generates additional image layers storing information such as depth or object id values [8] for each pixel of the image. Figure 1 shows examples of such image layers. This additional data is provided as separate images, in which each pixel value does not necessarily encode a color and is not necessarily dedicated for human cognition:

Color layer: A color layer contains a color value (e.g., RGB) for each pixel. According to the current WPVS specification, the service consumer can request a transparent background (RGBA), which requires image formats that support transparency, e.g., PNG or GIF.

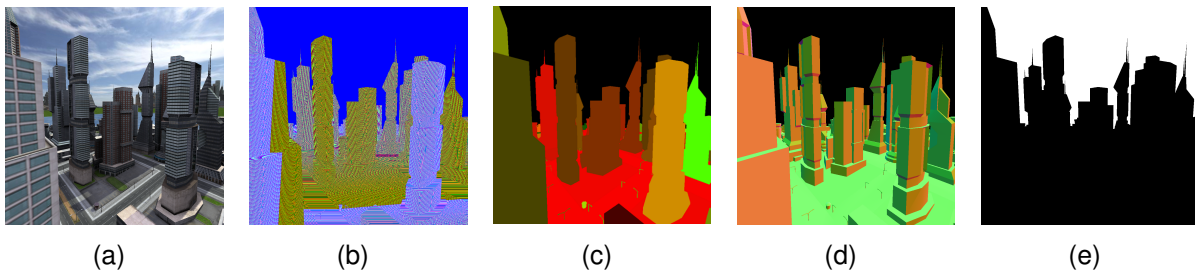


Figure 1: Examples of image layers that are provided by WPVS++: (a) color layer, (b) depth layer, (c) object id layer, (d) normal layer, and (e) mask layer.

Depth layer: A depth layer describes the distance to the camera for each pixel. Depth images can serve for multiple effects, e.g., for computing 3D points at each pixel of the image (image-based modeling) or for various rendering techniques such as depth-of-field effects. Furthermore, depth information represents a major means to compose multiple images generated from the same camera position.

Typically, graphics hardware provides logarithmic and linearized depth values $z \in [0, 1]$. Different from this, we suggest to provide depth values as distances to the virtual camera in meters. This representation abstracts from computer graphical details and the distance values can be used without additional computation in many applications.

Depth values are stored in IEEE 32 bit Float representation as images by segmenting their byte representations and assigning them to the color components of a pixel. Due to the direct storage as color components, resulting depth images possess very little pixel-to-pixel coherence, and should not be stored by lossy image formats. Thus, we suggest at least 32 bit (RGBA) images for this type of depth value storage. Consequently, a service consumer has to recompute the depth values from the color components of the requested depth layer.

Object id layer: An object id layer contains a unique id for each pixel that refers to a scene object. Using this information, we can select all pixels that show a specific feature, e.g., for highlighting, contouring or spatially analyzing the feature. For facilitating consistent interaction across multiple images, object ids should be unique over multiple requests. For that, they could be computed from an object id that is unique for the whole dataset.

Normal layer: A normal layer describes the direction of the surface normal at each pixel. This could be used for the subsequent integration of additional light sources and the adjustment of the color values around that pixel, e.g., for highlighting scene objects. Each vector component (x, y, z) of the normalized normal vector in camera space is encoded as color component (R, G, B) of a 24 bit color image.

Mask layer: A mask layer contains a value of 1 for each pixel that covers a scene object and 0 otherwise. Thus, a mask layer can be stored as 1 bit black and white image. It could support the easy altering of the scene background or provide information about

unused image space, which could be used, e.g., for blending advertisement, labels, etc.

Category layer: A category layer classifies the image contents on a per-pixel basis; the classification could be based on, e.g., feature types, object usages, or object names. The Category layer extends the so far described concept; it consists of an object id layer and a look-up table that lists for each object id the corresponding category.

Each image layer is encoded by standard image formats. This means to support the same principles for data encoding, data exchange, and client-side data loading and processing for all image data. Additionally, using images allows for applying state-of-the-art compression algorithms. For color data, which is dedicated for perception by humans, lossy compression algorithms can be used; for this data, JPEG mostly represents a suitable data encoding format. For data that requires exact values for each pixel, lossless image compression must be applied; we suggest PNG to encode this image data. PNG is also used for color images that shall contain transparency. Technically we encode JPEG images by 24 bit per pixel and PNG images by 24 or 32 bit per pixel. Higher data accuracies could be reached by using image formats with higher bit rates, e.g., 32 (48) bit images or even 48 (64) bit images. Drawbacks include increased image size and, thus, transfer load, and processing time at both the service-side and the consumer-side.

3.2 Image Sets

Various applications could require to fetch multiple, spatially coherent image layers from the WPVS++. As the WPVS++ is a stateless service (i.e., each service request is self-contained and its processing is independent from any preceding request), it potentially has to fetch, map, and render disjunctive sets of geodata for each service request. In particular, this is relevant when considering large 3DGeoVEs or remote data sources to be visualized.

From a performance point of view, it could be better to support multiple image layers within a single WPVS++ response. Provided that the requested views are spatially coherent (i.e., they are located close to each or have overlapping view frustums), this could reduce the number of switches of the rendering context. As an additional benefit, the network communication overhead caused by multiple requests is reduced.

These image sets could provide multiple image layers for the same camera specification (e.g., a set consisting of a color layer, depth layer, and object id layer) as well as images for different camera specifications (e.g., perspective views along a path through the 3D world).

3.3 Convenient Camera Specification

The current OGC WPVS provides a camera specification that is based on a point-of-interest (POI) and is not convenient for many applications. For example, specifying a setting that shows a view out of a window to a specific point in the 3D world demands for laborious computations for deriving distance, pitch, yaw, and roll angles.

As various applications could profit from a different specification of the view frustum, the WPVS++ replaces the camera specification by another one that is based on only three vectors: the 3D coordinate of the camera position (POC), the 3D coordinate of the point-of-interest (POI), and an optional camera up vector (UP) for specifying camera rolls. Furthermore, parameters for near plane and far plane are required; they describe the culling volume used during the rendering process and are necessary, e.g., for the generation and usage of depth image layers. For the specification of distorted images, we suggest replacing the angle-of-view parameter by a field-of-view angle in degrees in horizontal direction (FOVY) and to complement it by the optional vertical angle (FOVX). If FOVX is not specified, it is derived from the image dimensions.

3.4 Non-Perspective Projections

The current OGC WPVS is intended for central perspective projections only. Complementary, an image-based 3D portrayal service could offer additional projection types, such as orthographic projections, which are used, e.g., for architectural applications.

For extending the WPVS++ for orthographic projections, we suggest an alternative camera specification Orthographic that is controlled by the parameters Left, Right, Top, and Bottom, which describe the borders of the cuboid view frustum.

Beyond perspective and orthographic projections, a 3D portrayal service could support more advanced projection types such as oblique images, panoramic views, or multi-perspective views. While some of these projections could be generated from 2D images as provided by the current WPVS implementation, others require geometric information and, thus, are an integral part of the rendering process and need to be implemented by the 3D portrayal service itself.

3.5 Requesting 3D Coordinates and 2D Positions

Retrieving 3D geocoordinates for a specific 2D pixel position in a generated image is useful for many applications (e.g., specifying the position of an intended annotation). For this, the WPVS++ provides a GetPosition operation. As the WPVS++ is stateless, in addition to the 2D pixel position the image specification of the original GetView request has to be part of the GetPosition request. This operation is useful for navigation in the visualized 3DGeoVE: A user could select two image pixels that specify the desired point of camera and point of interest; the client requests corresponding 3D coordinates, and uses these for specifying the camera in a subsequent GetView request.

The other way round, the WPVS++ can compute the pixel position of a 3D coordinate if this 3D position is within the view frustum. Again, this could support the

interaction capabilities; for keeping orientation values high, a client could e.g., request the pixel position, where the virtual camera was located in the view before.

3.6 Requesting Feature Information

Similar to a WMS, a GetFeatureInfo operation provides extra information about the geobjects at a specific pixel position. According to the underlying data source and its capabilities, various response formats are supported, e.g., attribute names and values in XML format or GML-structured data sets. Additional formats can be supported and specified as mime type within the service request. In the case of structured responses (e.g., in GML format), the WPVS++ operation GetLayerInfo can be used for retrieving schema information for specific datasets.

3.7 Navigation Support

Navigation represents a fundamental interaction type for 3DGeoVEs. From virtual globes and online map systems users know to navigate by using mouse and keyboard in a real-time interactive manner. However, especially due to the image-based approach, the WPVS++ inherently provides only non-real-time step-by-step navigation. Assistant navigation techniques could compensate that drawback and allow a user to interact with the image, e.g., by sketching a desired point of interest, which serves as input of a GetCamera operation. The portrayal service automatically interprets this navigation input, taking into account scene objects, their types, and their navigation affordances, and computes and responds a camera specification, which can be used for requesting the corresponding view.

3.8 Analysis Support

Measuring within a 3D view represents a main functionality used to retrieve information about the spatial extent of features, their spatial relationships, and the overall spatial layout of a 3D scene. Thus, the WPVS++ supports the measurement of distances, paths, and areas. *Path measurement* computes the sum of the Euclidean distances between the 3D positions derived from each pair of consecutive 2D pixel positions. If only two 2D pixel positions are provided as input, the WPVS++ performs distance measurement. *Area measurement* computes the area which is outlined by the input points. In 3D, area measurement is not straight forward, as the derived 3D points are not likely to be coplanar.

3.9 Interactive, Lightweight JavaScript Client API

Navigation capabilities mainly depend on the functionality of the clients that consume the WPVS++. Simple clients can provide a step-by-step navigation based on requesting and displaying single views. For this, the additional GetPosition operation facilitates

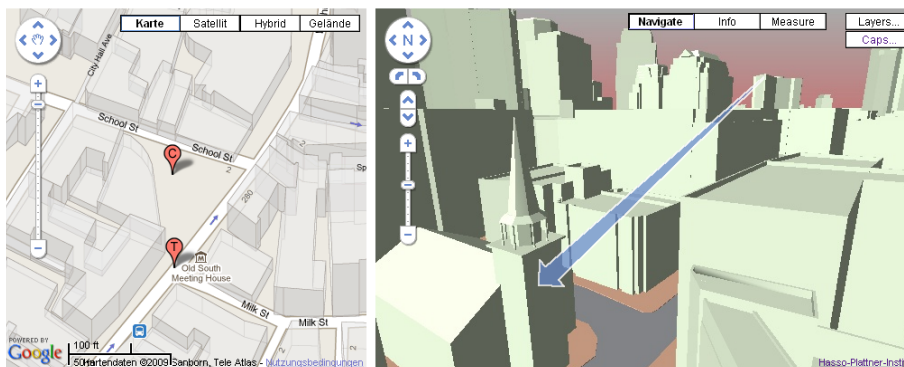


Figure 2: 3D view client (right) integrated with Google™maps (left). 2D and 3D visualization are synchronized, i.e. navigation on the 2D map immediately updates the 3D view and vice versa. The blue arrow indicates a navigation operation based on the WPVS++ GetPosition operation, i.e. setting the point of camera and point of interest. (Data: Boston Redevelopment Authority)

an easy and targeted camera manipulation. Based on depth information, more complex clients could apply, e.g., image-based rendering techniques for providing more convenient and close to real-time visualization and navigation functionality [6].

As example of lightweight clients, a JavaScript client has been implemented, which can be run in a web browser without additional plug-ins or libraries. It provides step-by-step navigation such as moving left/right, tilting up/down, or rotating around the POC or POI. Moreover, this client takes advantage of the GetPosition operation for retrieving 3D coordinates for changing the virtual camera's orientation and for implementing a "move there and orient here" navigation. The GetFeatureInfo operation is used for retrieving object information about features in the image. The GetMeasurement operation is facilitated for allowing users to measure distances in the portrayed 3DGeoVE.

The 3D view client provides an API for positioning the virtual camera and for corresponding callback functionalities. So it can be easily embedded into existing applications such as web-portals. As an example the 3D view client has been integrated with Google™maps (Figure 2), which allows for exploiting the functionalities of both 2D maps and 3D views.

4 Sketch-Based 3D Navigation Techniques

Navigation represents the fundamental interaction technique in 3DGeoVEs as it enables users to explore the 3D world and to interact with its objects. In web-based 3D portrayal services, efficient navigation techniques and strategies can help to increase the interactivity of these applications: Smart navigation techniques could hide the step-by-step image retrieval process and provide efficient navigation means, potentially compensating the absence of real-time navigation capabilities, e.g., by preventing "getting-lost" situations, confusing view configurations, or loss of visual context information.

For the application with 3D city models, we developed a sketch-based navigation technique [5]. This allows a user to sketch navigation commands by drawing gestures

and object-related sketches on a 3D view. These sketches are processed, classified, and interpreted by a navigation command system, which takes into account the sketch class, its shape, the sketching speed, potential preceding sketches, and the semantics and inherent navigation affordances of the scene objects affected by the drawings. According to the current context, the interpreted sketches are mapped to camera animations, which are automatically performed by the navigation system. This approach provides a higher-level navigation: A user is not forced to directly control the virtual camera, e.g., by keyboard, but can maneuver by task- and object-related sketches.

Within a service-based system architecture, the sketch-based 3D navigation technique could be provided by the 3D portrayal service itself, i.e., integrated within a WPVS++ and, e.g., provided by the proposed GetCamera operation. Alternatively, the sketch processing and interpretation could be performed by an additional service, e.g., an OGC Web Processing Service (WPS). In this case, the sketch interpretation service would request image layers that provide thematic information about the image such as object id images or category images.

5 Orchestration of Image-Based 3D Portrayal Services

A core concept of service-based systems represents the composition of distributed functionality in a standardized manner. This enables the construction and flexible adaption of complex and value-added systems and applications. With image-based 3D portrayal services, orchestration mainly refers to A) the combination of different 3D views, B) the embedding of additional information into a 3D view, C) the altering of a 3D view, or D) the computation of metrics (e.g., object visibility). For such functionalities, the extended WPVS++ provides various sources for computation. Considering depth-related image blending and combination, the depth layer represents a major resource for image-based processing chains [2].

We demonstrated the applicability of such a service composition for the example of 3D annotations, which represent essential elements to communicate textual and symbolic information for cartographic maps and within 3DGeoVEs [7]. A number of criteria, e.g., a clear correlation with the feature, the legibility of the annotation, and the occlusion of other annotations or important image parts have to be considered. In our approach and implementation, a composition client receives a color image and a depth image from an extended WPVS and forwards them to a new Web View Annotation Service (WVAS) along with a set of annotation descriptions and configurations. The WVAS computes and overlays embedded textual annotations. WPVS++ interaction extensions turned out to be essential for supporting such higher-level functionality. In the case of 3D annotations, picking a 3D positions from the image is fundamental for positioning the 3D annotations within the 3D view.

We showed, how two complementary 3D visualization techniques can be seamlessly combined by implementing these techniques by two independently designed, implemented, and deployed web services; chained together, they form a higher-level web service chain. The approach offers high degree of interoperability because the individual web services do not need to exchange contents of 3DGeoVEs.

6 Summary

This report provides a description of my research work during the past months and gives an overall summary of my research work in the field of service-based, image-based 3D portrayal. The contribution of my research work is in the field of interactive, image-based visualization of complex, heterogeneous, and distributed geodata by service-based 3DGeoVEs. By example I showed the feasibility of integrating such data at the visualization level, I extended the concept of image-based 3D portrayal service in a way that allows high interactivity, and I demonstrated the potentials of chaining image-based 3D portrayal services. Ongoing work includes the standardization of the interactive image-based 3D portrayal service as an OGC implementation specification; the functionalities and the prototype implementation have been presented to the OGC, a specification draft is currently prepared.

References

- [1] Geographic Information – Services. ISO 19119, International Standard, 2005.
- [2] Nadine Alameh. Chaining Geographic Information Web Services. *IEEE Internet Computing*, 7(5):22–29, 2003.
- [3] Jürgen Döllner and Benjamin Hagedorn. Integrating Urban GIS, CAD, and BIM Data By Service-Based Virtual 3D City-Models. In Massimo Rumor, Volker Coors, Elfriede M. Fendel, and Sisi Zlatanova, editors, *Urban and Regional Data Management: UDMS 2007 Annual*, pages 157–170, Stuttgart, Germany, October 2007. Taylor & Francis Ltd.
- [4] Benjamin Hagedorn and Jürgen Döllner. High-Level Web Service for 3D Building Information Visualization and Analysis. In *ACM 15th International Symposium on Advances in Geographic Information Systems (ACM GIS)*, Seattle, WA, November 2007.
- [5] Benjamin Hagedorn and Jürgen Döllner. Sketch-Based Navigation in 3D Virtual Environments. In *8th Int. Symposium on Smart Graphics 2008*, August 2008.
- [6] Benjamin Hagedorn, Dieter Hildebrandt, and Jürgen Döllner. Towards Advanced and Interactive Web Perspective View Services. In *4th International 3D GeoInfo Workshop*. Springer, 0 2009. to appear.
- [7] Benjamin Hagedorn, Stefan Maass, and Jürgen Döllner. Chaining Geoinformation Services for the Visualization and Annotation of 3D Geovirtual Environments. In *4th International Symposium on LBS and Telecartography*, November 2007.
- [8] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-D shapes. *SIGGRAPH Comput. Graph.*, 24(4):197–206, 1990.