3D Gevisualization Services for Efficient Distribution of 3D Geodata

Jan Klimke

Computer Graphics Systems
Hasso-Plattner-Institut
jan.klimke@hpi.uni-potsdam.de

Design, implementation, and operation of services for interactive 3D geovisualization are faced with a large number of challenges including (a) processing and integration of massive amounts of heterogeneous and distributed 2D and 3D geodata such as terrain models, buildings models, and thematic georeferenced data, (b) assembling, styling, and rendering 3D map contents according to application requirements and design principles, and (c) interactive provisioning of created 3D maps on mobile devices and thin clients as well as their integration as third-party components into domain-specific web and information systems. This report presents a concept and implementation of a service-oriented platform that addresses these major requirements of 3D web mapping systems. It is based on a separation of concerns for data management, 3D rendering, application logic, and user interaction. The main idea is to divide 3D rendering process into two stages. In the first stage, at the server side, an image-based representation of the 3D scene is created by means of multi-layered virtual 3D panoramas; in the second stage, at the client side, the 3D scene is interactively reconstructed based on these panoramas. Beside the interactive client introduced earlier, scalable variant for provisioning of virtual 3D city models is described, that utilizes the capabilities of our rendering service for data preparation.

1 Introduction

The availability of 3D geodata, its volume and its quality are constantly growing. Therefore, a high quality 3D visualization of massive and detailed data sets represents a computationally expensive task: It consumes large amounts of main memory, disk, network, CPU, and GPU resources in order to deliver an interactive user experience and a constantly good visual quality. For distribution of such 3D contents in a web environment, two principal concepts can be found in existing 3D web mapping approaches, client-side 3D rendering, i.e., assembly and streaming of 3D display elements such as 3D geometry and textures by a 3D server, whereby the corresponding 3D clients manage scene graphs and perform real-time 3D scene rendering (e.g., Google Earth, OGC W3DS); and server-side 3D rendering, i.e., assembly and rendering of 3D display elements by the 3D server, which delivers views of 3D scenes to lightweight clients (e.g., OGC WVS).

Taking into account the increasing complexity and size of geodata for 3D maps (e.g., complex 3D city models), the need for high-quality visualization (e.g., illustrative or photorealistic rendering), and rapidly growing use of mobile applications (e.g., smart phones and tablets), these principal concepts are faced by fundamental limitations: client-side 3D rendering fails to provide fast access to massive models due to bandwidth limitations and cannot guarantee high-quality rendering results as the graphics capabilities of nearly all mobile devices differ significantly, while a pure server-side 3D rendering is inherently limited with respect to interactivity and does not take advantage of todayâĂŹs mobile device graphics capabilities.

In this report i will present a framework based on a Web View Service as core component encapsulating the complexity of 3D model data and computer graphic techniques. Conventional techniques for visualization of 3D geodata, in particular virtual 3D city models, are built upon client-side rendering of 3D geometry and texture data transmitted from remote servers to client applications. These approaches are usually limited in terms of complexity of models and the visual quality they can provide, since high-end computer graphic technique demand for state of the art 3D hardware.

Utilizing this type of service as a basis for client applications allows for provisioning high quality visualization of 3D model data of nearly arbitrary sizes on devices and platforms, that would otherwise not be able to provide usable, high-quality 3D visualization. Through externalizing image-generation from client side on server side, a consistent presentation of 3D geodata is possible, regardless of client hardware or software.

Approaches for image-based client applications presented earlier, are allays bound by the performance of the underlying rendering server system in order to provide a fully interactive user experience. In applications, this full 3D interactivity is often not necessary and leads to a larger complexity for users regarding camera interaction. Therefore, in Section 5, we introduce a scalable application that was implemented for mobile devices as well as for web browsers. These applications addresses these two challenges, be limiting the possible camera interaction. This allows to pregenerate the necessary image data that has to be transmitted to clients.

The remainder of this report is organized as follows. Section 2 introduces related work in the are of image based remote visualization. Section 3 introduces the specific challenges that arise when building service-based visualization system in more detail. Section 4 describes briefly the overall architecture of the visualization system which forms the basis of my work. Finally, an outlook of planned future work concludes this report in Section 6.

2 Related Work

The interoperability of systems and applications dealing with geodata is an central issue in order to build systems out of interoperable software components for geodata access, processing, and visualization. Beside a common understanding on information models [3], definitions of service interfaces are necessary. The Open Geospatial Consortium (OGC) defines a set of standardized services, models, and formats for geodata encoding and processing. For example, a Web Feature Service (WFS) [16] can provide geodata, encoded in the Geography Markup Language (GML) [13] or City Geography Markup Language (CityGML) [6], and processed by a Web Processing Service (WPS) [15]. For geovisualization processes a general portrayal model is provided by the OGC that describes three principle approaches for distributing the tasks of the general visualization pipeline between portrayal services and consuming applications [1, 7]. While the OGC Web Map Service (WMS), providing map-like representations of 2D geodata, is widely adapted and used, 3D geovisualization services have not been elaborated to a similar degree. Several approaches for 3D portrayal have been presented [2]. Two types of 3D portrayal services, currently discussed in the OGC context, can be distinguished: Web 3D Service (W3DS) [14]: It handles geodata access and mapping to renderable computer graphics primitives (e.g., textured 3D geometry represented by scene graphs) and their delivery to client applications. Web View Service (WVS) [8]: It encapsulates the image generation process (i.e., it implements the geovisualization pipeline) of 3D models, delivering rendered image representations ("portrayals") to client applications.

By focusing on developing international standards for 3D Portrayal, an interoperable, service-based system can be built, that allows replacing component implementations selectively with other implementations. Further, system components, especially a 3D rendering service or a W3DS instance can be reused in other systems. Several approaches exist for remote rendering of 3D virtual environments [4, 12].

Mostly, they rely on constant transmission of single images [17] or video streams [10, 11] to client applications. In contrast to those applications that are completely dependent on the current data throughput of the network connection, our approach uses a latency hiding technique. This technique allows for a continuous user interaction operating on the locally available data also in situations with very low data transmission rates between 3D rendering server and clients. Another approach is to manage and render a low resolution 3D model on client side, to allow users to explore the model interactively; when interaction stops, remote rendering is used to create and display an image of the high-resolution 3D model [9]. For delivery of, e.g., large-scale textured 3D city models, this approach is not suitable, since a transmission and rendering of low resolution model representations on client-side would still exceed network and client-side rendering capabilities. In approach an image-based 3D approximation of a model is created by the client, using image data transmitted from a 3D rendering service.

3 Challenges

Design, implementation, and operation of interactive 3D geovisualization services are faced with a number of challenges including:

Data processing and integration Massive amounts of heterogeneous and distributed 2D and 3D geodata such as terrain models, buildings models, and thematic georeferenced data form the basis for 3D maps. This data has to be processed for visualization and integrated into the overall visualization system in order to be combined in a 3D map.

3D map content assembly, rendering and styling To communicate spatial and georeferenced information, 3D map content must be selected, styled and rendered according to application requirements and design principles.

Interactive provisioning Created 3D maps should be available on mobile devices and thin clients in an interactive manner. Further, third-party vendors should be able to integrate 3D maps as components into their domain-specific web and information systems.

Interoperability A system for 3D web mapping should rely on established standards in terms of data formats and service interfaces.

Upscaling When it comes to publishing of high-quality 3D visualization of 3D geodata, there should be a way to provide at least parts of the functionality to large numbers of concurrent users.

In particular, challenge (c) influences the way a service-oriented system for 3D mapping can be built. Large amounts of data have to be transmitted, processed, and stored for generating a useful 3D map, common client applications have to scale with the size of the underlying 3D model, because they deal with the massive amounts and the complexity of model data on client side. In order to provision 3D maps on a large variety of devices with heterogeneous hardware and software capabilities and undefined as network connection speed, the effort of processing, transfer, and rendering of 3D map content should be decoupled from a client application, while still providing a user with an interactive user experience. This diversity and performance considerations is especially an issue, when designing applications for mobile devices or browser-based 3D mapping solutions. Clients for these platforms should provide a equally high visual quality, regardless of the capabilities of the individual device or platform.

4 A Service-based 3D Visualization System

An architectural overview over the system is provided in Figure 1. There are many heterogeneous sources and formats for 3D geodata, many of them are not optimized for visualization purposes, although containing information about the detailed visual appearance of features. In order to be rendered efficiently by the 3D rendering service, the different formats are translated into an intermediate format that fulfills two requirements: (a) allow for highly efficient rendering of massive data amounts on the one hand side and (b) maintains the semantical structure of the original geodata in order to allow for professional applications requiring access to the underlying data sources and their structure. The Preprocessing service provides this type of functionality. In course of the preprocessing, a database is built up, containing meta informations, such as the coordinate bounding box, parent-child relation of objects, their type (e.g., building, road, city furniture, etc.), and a mapping from the original id of an imported feature and the assigned object id that is later used for rendering.

The resulting data format allows for a consistent implementation of 3D rendering techniques, since these can be built against a widely consistent kind of data (e.g., numerical scale of coordinates, geometry structure, and texture format). This eases the development of graphical stylizations and real-time rendering techniques for the 3D rendering service implementation significantly. The 3D rendering service, as core component of the service-based visualization system, creates rendered images of 3D geodata. It integrates data locally available preprocessed data as well as data available from external services, such as map data from standardized Web Map Services or 3D model data from Web 3D Services. The integration of external data services for provides more flexibility for visualization applications using the 3D rendering service. Especially, constantly changing, dynamic data, such as vehicle positions or water levels, can be integrated without a prior translation into an intermediate data format.

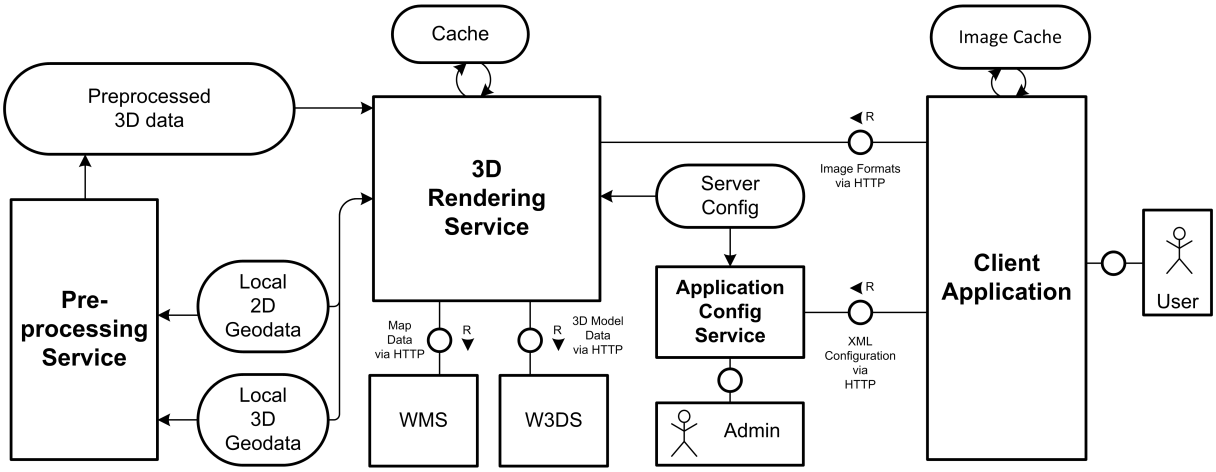


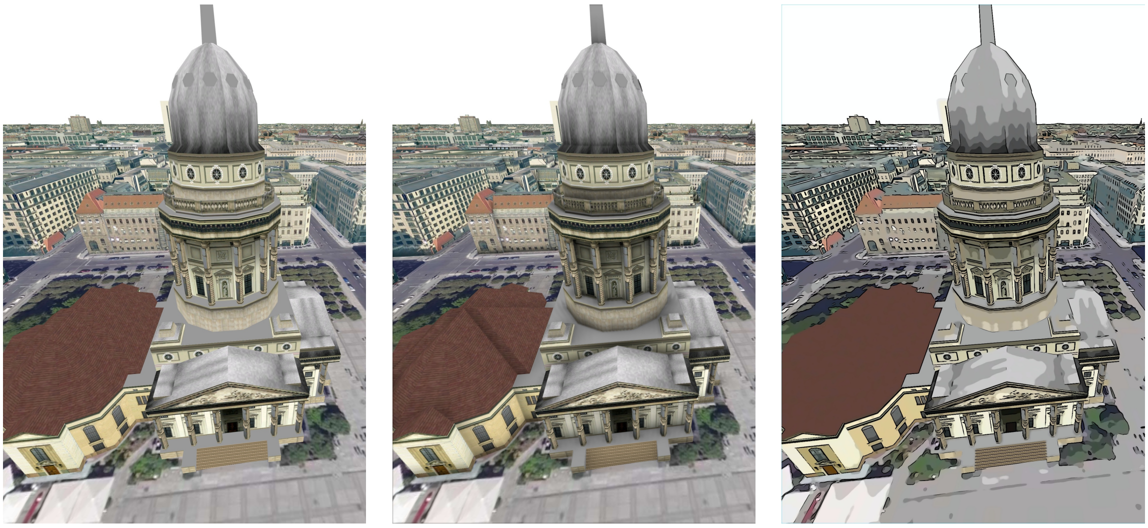
Figure 1: Architecture of a service-based system for 3D geovisualization. The 3D Rendering Service encapsulates the visualization pipeline for geodata and hides away the complexity of dealing with large, complex structured, and possibly distributed data sets.

To provide client applications that are both, universally applicable in different working processes and interactively manageable, an application configuration service is used to setup 3D rendering servers as well as corresponding client applications. Application specific configuration of multi purpose client applications is done remotely. This allows us to provide an individual application configuration, including, e.g., appropriate model data, map data, styling effects or predefined, or generated camera positions, for specific users, roles, or use cases.

4.1 Stylization of 3D Contents

Stylization of 3D maps is one of the main issues of the 3D map generation process. It supports the efficient communication of geoinformation. In our 3D rendering service, we support flexible stylization of 3D map contents. For this it provides a number of options to style the 3D maps. Specific textures can be applied to each model element, e.g., a terrain model can be textured using different 2D maps. Further, different 3D rendering effects can be applied per model element, e.g., a specialized rendering for planned constructions. Additionally, the 3D rendering service offers so called image styles that affect the overall appearance of the 3D scene.

This stylization is implemented as an image-based post-processing executed after the 3D rendering of the scene geometry and textures has been performed. Data from



(a) (b) (c)

Figure 2: Examples for different stylization applied to a 3D scene: a) Without any explicit stylization. b) With a self shadowing effect that improves depth perception c) an image abstraction effect, removing details from surface textures.

different image layers is used together with additional configuration options to configure the image stylization effect, e.g. the id of a scene object to be highlighted by a halo effect. Stylization effects are implemented efficiently using the graphics hardware. Unnecessary copy operations of source image-layers are avoided by reusing them on the graphics card without prior download into the main memory. This way, the graphics hardware of the rendering server is used in a very efficient way for image-based computations, e.g., non-photo-realistic rendering. Examples for image-based styling are depicted in Figure 2.

The image-based stylization has one major advantage compared to conventional techniques for 3D-stylization: The computational complexity does not depend on the model size and complexity. The computational costs are mainly dependent on the resolution of the source and target images and the complexity of the desired effect. Image-based techniques can also be used to visualize thematic data, e.g. through projection of image data or applying color values for certain objects encoding specific data values.

5 Thin-Clients for Map-Like Visualization of 3D Geodata

Integration of 3D visualization of geodata in a variety processes and application areas, e.g., in city planing, infrastructure management, tourism or city marketing, gains more and more importance. In contrast, the problem of provisioning, deployment and scaling 3D visualization applications for a large scale audience (e.g., thousands of concurrent users) in a cost efficient way remains a challenging task. Existing approaches for 3D visualization of geodata often require a client system that is capable of performing complex 3D rendering for massive amounts of data. The server-side requirements and especially the complexity of implementing high-quality rendering techniques for a large number of highly heterogeneous client hardware and software platforms makes it very hard to implement a usable 3D visualization component that can be integrated into third party applications.

Recently, we presented a client application that reconstructs a lightweight representation of a complex server-side model in order to allow and interactive user exploration and free navigation in 3D world [5]. While this kind of application addresses a lot of issues that arise when massive amounts of 3D geodata, such as virtual 3D city models, is used in mobile or web-based visualization applications, the issue of the complexity of 3D navigation in 3D geovirtual environments remains. To address this issue, and to provide a solution for exploring 3D city models in connection with related 2D and 3D application data, we created a map-like visualization that uses image data that can be pregenerated while still keeping the connection between underlying 3D geometry and the semantic data they represent. Therefore we use parts of the framework presented in Section 4, in particular preprocessing and rendering services, to generate tileable views of a 3D dataset. A restriction of user interaction, conceptually to a motion in parallel to the earth's surface and a modification of the distance between surface and camera, makes it possible to create sets of image tiles for applications that cover a specific spatial region. In contrast to earlier clients, which where using a conventional perspective projection for image generation, this application works with an orthographic projection. In this way, it is possible to generate we can create a specific image for each numbered spatial region (image tile). This way, each tile does exclusively contain a certain part of the overall model geometry. A script is used to generate all tiles for a spatial region, in our case the region covered by the city model of Berlin, in several discrete zoom levels (discrete distance steps for surface-camera distance).

The files for image tiles are organized in conformance to the Tiles Map Service, as a quasi standard for providing a simple access scheme for tiles georeferenced image data. The corresponding client application, depicted in Figure 3, requests tiles that are needed for the current zoom level and visible spatial region.

Besides the efficient visual presentation of 3D geodata, linkage of the visual entities and their underlying geodata plays a major role for visualization solutions to be applicable in every day working processes. Therefore, images containing object identifiers for an color image tile are generated. They can be used to get object identifiers for specific positions given in pixel coordinates for single tiles. The rendering service instance is then able to map these identifiers back to the original identifier in the source dataset. Besides these capabilities for data access, id images can be utilized for rendering effects on client side, e.g., highlighting of single objects.

6 Conclusions and Future Work

In this report I presented a service-based framework for 3D portrayal of 3D geodata implementing the 3D visualization process. Using the extended image generation capabilities of the 3D Rendering service, significant amounts of computational complexity

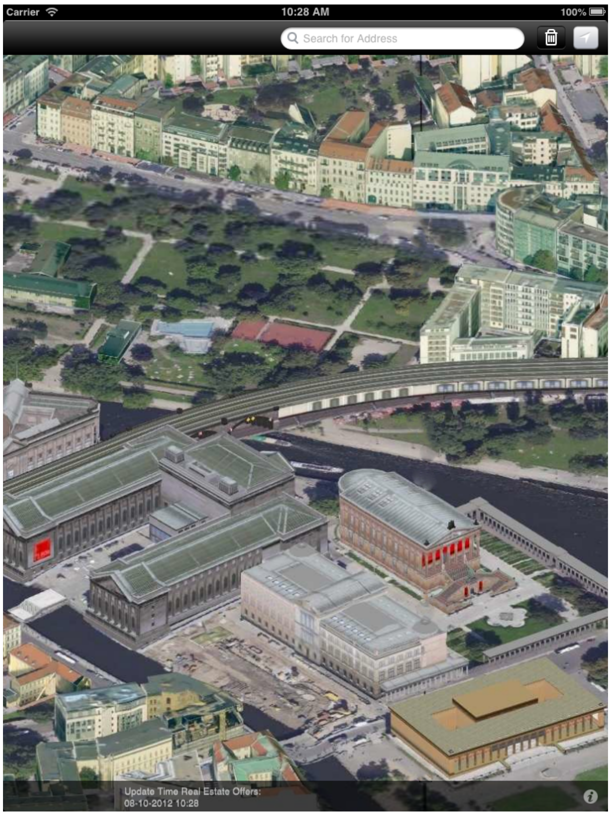


Figure 3: Example for a Web-based Visualization of the virtual 3D city model of Berlin created using the service pipeline. The created map-like visualization can be arbitrarily styled and customized. Underlying image data can be pregenerated in order to serve a large number of clients.

can be implemented on server-side, allowing for a more resource efficient implementation of client applications. Besides interactive, image-based, real-time 3D-clients presented earlier, map-like applications using pregenerated image-data provide further advantages with regard to server-side effort. Generated image-tiles can be served easily for a large number of users, e.g. by using highly scalable cloud storage services.

The next steps in research on our framework for service-based 3D visualization will be to perform performance tests and evaluations for several large-scale datasets in order to reveal potential for further optimizations and extensions to the overall system and explicitly to interactive client applications. Measuring the performance of the service based visualization system will be part of my work during the next month. Both, the performance of the rendering service itself, as well as the performance of the overall image-based visualization system need to be explored in more detail in order to identify possible bottlenecks, arising, e.g., from client accessing the rendering service concurrently.

Further we see, that 3D interaction and rendering techniques that support latency hiding seem promising in order to improve the overall user experience of image-based 3D visualization applications. Within the next term i am therefore going to focus on such techniques in connection with specific rendering technique for that purpose.

References

[1] A. Altmaier and T. H. Kolbe. Applications and Solutions for Interoperable 3d Geo-Visualization. In D. Fritsch, editor, *Proceedings of the Photogrammetric Week 2003*, pages 251–267, Stuttgart, 2003. Wichmann.

[2] J. Basanow, P. Neis, S. Neubauer, A. Schilling, and A. Zipf. Towards 3D Spatial Data Infrastructures (3D-SDI) based on open standards experiences, results and future issues. In *Advances in 3D Geoinformation Systems*, Lecture Notes in Geoinformation and Cartography, pages 65–86, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[3] Y Bishr. Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, 12(4):299–314, 1998.

[4] A Boukerche and RWN Pazzi. Remote rendering and streaming of progressive panoramas for mobile devices. In *Proceedings of the 14th annual ACM . . .*, pages 691–694, 2006.

[5] Juergen Doellner, Benjamin Hagedorn, and Jan Klimke. Server-based rendering of large 3d scenes for mobile devices using g-buffer cube maps. In *Proceedings of the 17th International Conference on 3D Web Technology*, Web3D ’12, pages 97–100, New York, NY, USA, 2012. ACM.

[6] Gerhard Gröger, Thomas H. Kolbe, Claus Nagel, and Karl-Heinz Häfele. OpenGIS City Geography Markup Language (CityGML) Encoding Standard Version 2.0.0. Technical report, Open Geospatial Consortium Inc., 2012.

[7] R. B. Haber and D. A. McNapp. Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In *Visualization in Scientific Computing*, pages 74–93. IEEE, 1990.

[8] B. Hagedorn. Web view service discussion paper, Version 0.6. 0. *Open Geospatial Consortium Inc*, 2010.

[9] David Koller, M Turitzin, and M Levoy. Protected Interactive 3D Graphics via Remote Rendering. *Transactions on Graphics (TOG)*, 2004.

[10] Fabrizio Lamberti and Andrea Sanna. A streaming-based solution for remote visualization of 3D graphics on mobile devices. *IEEE transactions on visualization and computer graphics*, 13(2):247–60, 2007.

[11] D. Pajak, R. Herzog, E. Eisemann, K. Myszkowski, and H.-P. Seidel. Scalable Remote Rendering with Depth and Motion-flow Augmented Streaming. *Computer Graphics Forum*, 30(2):415–424, April 2011.

[12] Gianluca Paravati, Andrea Sanna, Fabrizio Lamberti, and Luigi Ciminiera. An open and scalable architecture for delivering 3D shared visualization services to heterogeneous devices. *CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE*, 23(11):1179–1195, 2011.

[13] Clemens Portele. OpenGIS Geography Markup Language (GML) Encoding Standard, Version 3.2.1, July 2007.

[14] Arne Schilling and T.H. Kolbe. Draft for Candidate OpenGIS Web 3D Service Interface Standard, Version 0.4.0, 2010.

[15] Peter Schut. OpenGIS Web Processing Service, Version 1.0.0, 2007.

[16] P.A. Vretanos. OGC Web Feature Service Implementation Specification, 2005.

[17] Andrew Wessels, Mike Purvis, Jahrain Jackson, and Syed (Shawon) Rahman. Remote Data Visualization through WebSockets. *2011 Eighth International Conference on Information Technology: New Generations*, pages 1050–1051, April 2011.