

Towards Efficient Camera Interaction in Service-based 3D Geovirtual Environments

Jan Klimke

jan.klimke@hpi.uni-potsdam.de

Service based geovisualization helps to enable high quality 3D visualization with minimal requirements for clients regarding computational power or rendering capabilities. While interaction with 2D maps is quite well understood, interaction in 3D is more challenging due to its additional degrees of freedom that have to be controlled. So called smart interaction techniques can help a user to master this 3D complexity using standard 2D input devices. To do so, additional information about the 3D environment, which is used for visualization, is needed. Such data is not per se available in many of the applications of service-based 3D visualization. This report proposes to encapsulate parts of the interaction process into service components that can be deployed independently from client applications. This lowers the complexity of client implementations, that are possibly running on a small handheld devices, so data access and computation necessary for interaction can be externalized to faster systems, which may also have a faster, more reliable network connection. Such a decoupled paradigm enables the deployment of service-based visualization applications depending on the client device's capabilities regarding computational power or data access (access rights or networking bandwidth).

1 Introduction

Since "a 3D world is only as useful as the user's ability to get around and interact with the information within it" [12] interaction components should play a major role in development of systems using geoinformation for 3D visualization. In a 3D virtual environment camera manipulation is the primary interaction since the specification of camera parameters defines what is visible to a user and therefore defines a user's context. Six degrees of freedom (namely 3D camera position (3 DOF) and 3D orientation (3 DOF)) have to be controlled to specify the position and orientation of a virtual camera. These additional variables, compared to interaction with 2D user interfaces, make camera interaction in 3D geovirtual environments (3D GeoVEs) a complex task for users. While the capabilities of standard input devices for today's applications, such as mouse, keyboard or touch sensitive surfaces, provide adequate means for controlling 2D user interfaces (such as map-based applications or conventional window-based ones), they do not allow a direct manipulation of all the parameters that are necessary for 3D po-

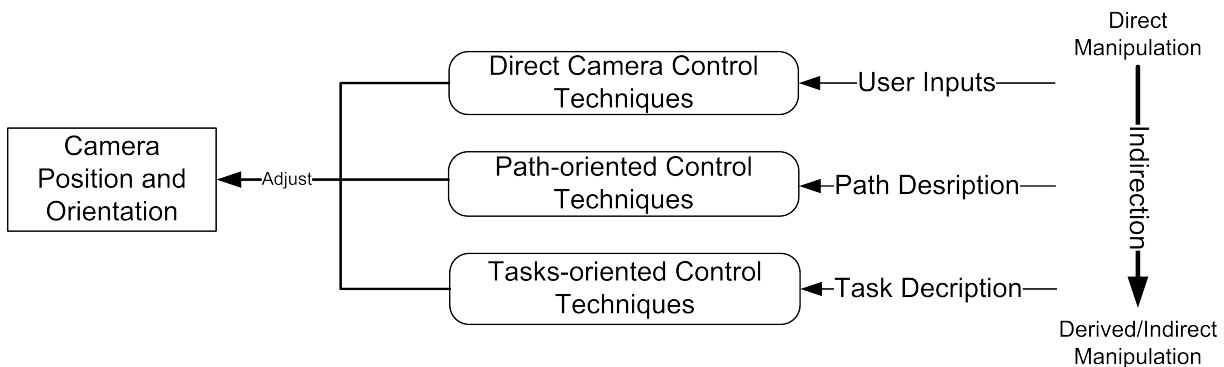


Figure 1: Classification of camera interaction techniques regarding their indirection from direct manipulation of camera parameters.

sitioning and orientation of a camera view in virtual 3D space. Hence, an intelligent mapping from user input to camera configurations is needed. Approaches for smart (assisting) camera control for 3D GeoVEs can use semantics of the underlying spatial model (e.g., 3D city model) to provide higher-level interaction to users, which reduces the number of feedback cycles needed to position and orient the virtual camera. Additionally such techniques try to avoid distracting or disorienting views of a virtual camera by applying constraints to its parameters.

Service-based portrayal for 3D GeoVEs allows for thin client applications by hiding the complexity of geodata handling, processing and large parts of the rendering from clients [8]. In this way, also clients with constraint capabilities regarding data access, memory, data processing, rendering and input (e.g., mobile phones as client devices) can be used for running 3D geovisualization applications. To support the arefore mentioned smart interaction techniques for camera interactions, such clients need service-side support for computation of camera view parameters and camera paths. To improve efficiency of interaction, camera navigation techniques have to be designed and selected with special regard to such limitations [4].

In this report a concept for separating the camera control process from the service-based portrayal itself is described. Therefore we provide a concept of how to decompose a camera control process into single, possibly distributed service components, each performing a specific task in the interaction process. Well defined interfaces for each service component facilitate reuse of existing functionality and can additionally serve for a more efficient prototyping of interaction techniques. The concept for a service-based support for interaction applications is a first step in my research towards future applications for SOA-driven 3D geovisualization.

The remainder of this report is organized as follows: Section 2 will give a short overview of work that motivates this paper. Afterwards, in Section 3 our approach for distributing camera control functionality is presented. Section 4 will summarize this paper and gives an outline of future research activities towards future service-based 3D geovisualization systems.

2 Related Work

Smart interaction techniques, also called assisting navigation techniques, help a user to perform a task and avoiding disorienting or distracting camera positions and orientations. There are several types of camera control techniques regarding the level of indirection from user input to camera parameters (Fig. 1). The notion of camera task describes an intended of camera movement and position. In general, users can be supported better, if a higher level specification of desired camera tasks is possible.

Semantics contained in virtual 3D city models, as type of geovirtual environment, can be used to evaluate user input and derive camera control tasks [7]. A defined camera control task (e.g., "show me that building", "guide me along this road") allows for generation of a camera path with respect to quality criterions, e.g., collision avoidance, retaining a user's orientation inside the 3D GeoVE, perceived smoothness of camera animation, or keeping certain points of interest visible. To enable comprehensible camera paths, approaches exist that are using basic physical models for camera control. For example inertia effects like acceleration and deceleration lead to a perceived better camera motion [2].

Assisting camera control techniques, which use semantics of the underlying model, involve additional requirements regarding data management and computational power since geodata is typically massive, heterogeneous and distributed. Especially thin clients are restricted in hardware (mobile devices like phones) or in software (thin, browser-based clients). This is a problem when 3D geovisualization applications are running on such devices. Nevertheless, through the high abstraction level from the user input, assisting camera navigation techniques seem to be promising especially for use cases that demand for a relatively low number of interaction cycles to reach a specific goal, e.g. performing a desired camera animation. A low count of interaction cycles between user and application is especially favorable for service-based systems because of the inevitable network delays during requests, which slow down service-based applications. Since mobile devices tend to have a possible unreliable network connection, this effect plays an important role if end-user applications are running on such devices.

Handling geodata for visualization is a complex task that demands for special high performance hardware (e.g., graphics adaptors, large main memory, CPU) to produce high quality visualizations. The heterogeneity of hardware components and the financial and configuration effort to deploy those hamper the implementation of such high quality visualizations. Since geodata itself is often not freely accessible, another point concerning geovisualization applications is data access and rights management. Owners of geodata do not want to grant full access to their data, but might want to enable its usage for visualization. Using the paradigm of service orientation for geovisualization applications can hide the complexity of rendering or sophisticated rights management from client applications by encapsulating steps of the visualization pipeline. The data management encoding and processing part of the service chain is already sufficiently standardized by the Open Geospatial Consortium (*OGC*) (i.e., Web Feature Service [13] for data access, CITYGML [6] or GML [10] for data encoding, and Web Processing Service [11] for data processing).

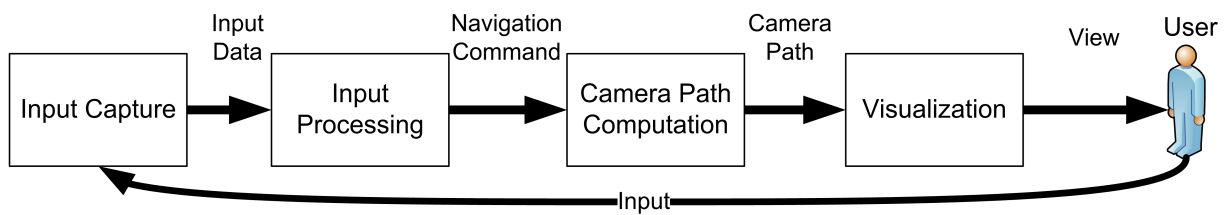


Figure 2: Separation of a camera interaction process into tasks together with the results generated by each step.

For portrayal of geodata, a 2D portrayal service has been specified by the OGC and is widely adopted. This Web Map Service (WMS) [3] creates map images from geodata. Currently, there is no corresponding OGC standard for 3D portrayal. Two active approaches towards OGC standards for 3D portrayal services are currently existing in the OGC community: The Web 3D Service (W3DS) [1] and the Web View Service (WVS) [8]. The approaches differ in the type of data they provide. While a W3DS provides display elements in a computer graphic coordinate system, a WVS provides layers of scene views encoded as images. This leads to different requirements for presentation clients needed to use such services. While a W3DS demands for 3D rendering capabilities on the client side for image synthesis, this is not necessary for WVS clients since, because this service performs server-side image synthesis.

Supporting navigation in 3D GeoVEs has not been a central issue during the development of current portrayal standard candidates. The W3DS draft standard does not include any facilities to support camera control on the server side, because the client is expected to handle all issues concerning user interaction and rendering. The WVS standard draft defines an optional *GetCamera-Operation* that returns a good camera definition for a set of 2D pixels. Here, the definition of what is a good camera perspective for the actual application is defined by the WVS implementation. So a very fundamental support for camera control is included in this standard, and can be used by thin clients for exploration of 3D GeoVEs .

3 The Camera Interaction Process

To support camera control for service-based visualization environments the process of interaction needs to be analyzed and decomposed into actions that can be performed by conceptually independent components. A clear definition of an interaction process in service-based visualization systems helps to identify the interfaces and data for communication of services. The components of such an interaction support system can be distributed. Depending on the capabilities and demands of client devices and applications, the distribution of the components could be dynamically adjusted.

We divide four tasks for camera control (Fig. 2). The input capturing task of an interaction process is done by the client device running an application. A variety of sensors can deliver input values for 3D camera control techniques, e.g., (series of) touch events for tangible surfaces, ui button events or keyboard events. Further, mobile

often include additional sensor hardware, such as GPS receivers, gyroscopes or accelerometers. So they are able to capture different information that describes a user's current context. This information can be used to influence the interaction with a 3D application to support a user by adjusting to his current context, especially with regard to mobile applications.

Raw input data can be preprocessed to provide a more high level input for the following steps. The input preprocessing step can involve, for example, smoothing input values, recognition of shapes from series of positions. The result of this step is a *navigation command* that encodes a camera task to perform by the animation of virtual camera.

The camera path computation step executes a navigation command by creating a camera path, which consist of one or more sets of parameters defining a virtual camera's position and orientation. The path computation itself may use additional data sources, for example, to provide collision avoiding camera paths. Therefore it has to use the same geometries that is currently used for visualization to be able to perform the necessary computations to comply with the data that is currently viewed.

The visualization step concludes a camera interaction loop. It applies the generated camera path to the image synthesis stage. Image generation itself may be done using geovisualization services or implemented independently at client side. A camera service itself is not bound to thin clients. It may also be used by thick oder medium clients to provide smart camera paths.

4 Service Support for Camera Interaction

The single steps of a camera interaction cycle presented above can be supported using service instances. Which types of services can be used and how they could communicate is depicted in Fig. 3. As described in the previous section, navigation commands are recognized from user inputs. The necessary operations for command recognition can possibly involve additional data to provide commands that rely on semantics of objects included in the current scene view.

As shown in Figure 1, a user's inputs do not have to influence the parameters of the virtual camera directly. They can also describe a higher-level task to be executed by the camera. Tasks to perform are, e.g., following a route, inspect a building, or taking an overview position for parts of the scene. To allow this kind of indirection from user inputs, a command recognition service can be used to extract navigation commands. Each type of navigation command can have specific parameters, e.g., a "move to" command can have a feature identifier as target description.

Conceptually a 3D camera service is able to compute a camera position or animation for executing one type of navigation command. Therefore a registry for such camera service is introduced which manages metadata of camera services and therefore helps to find the right service-endpoint that is able to execute the command which was recognized from user input. Camera navigation techniques compute camera parameters, respectively camera animations, according to criteria, defined by the type and implementation of a technique. Examples for such criteria are:

Collision Avoidance A navigation technique may be capable of creating a camera path for animation that does not intersect with other scene geometry. There are several strategies for collision avoidance for camera path computation. Collision avoidance can be either guaranteed, best-effort or there can also be no collision avoidance at all.

Orientation Camera navigation techniques can implement orientation preserving features. For example, a technique may try to optimize the visibility of landmarks to provide orientation support to users.

Task Task specific techniques compute camera paths that are specialized in performing a task such as object inspection, overview tours or routing the camera to a defined endpoint.

3D Camera services encapsulate camera navigation techniques. They use navigation commands, which have been computed in preceding steps, alongside with command specific parameters, e.g., current camera position or a point in space as target for the camera animation, and return a camera path, which consist of one or sets of camera parameters that describe at least a camera's position and orientation. The path computation itself may use additional data, either originating from the visualization service (e.g., for image-based techniques such as the one presented by McCrae et al. [9]) or from a WFS to provide certain quality features such as, for example, collision avoidance or a guaranteed visibility of certain points of interest.

5 Summary and Outlook

This report presented a first step towards service support for camera interaction in service-based 3D geovisualization systems. Here our central point is the definition of a service-supported interaction process using different processing stages.

The future research towards more user friendly service-based 3D geovisualization applications is seen in the following areas:

Definition of quality criteria for camera animations The definition of criteria for camera positions and animations is not trivial. Quality criteria might depend on several factors defined by the type of application using these camera positions, the context of a user and especially on the current task to be performed by a 3D visualization system. To judge the quality of camera positions and respectively animations, a computable definition of the quality of a scene view is necessary. A numerical description of the quality of a scene view may then be optimized by a technique that computes camera paths. A promising approach could be, for example, using methods from visual analytics in 3D city models, as presented by Engel and Döllner [5] in connection with image analysis to rate a specific scene view regarding the quality criteria.

Camera navigation techniques New approaches for creating the camera animation itself will help to generate camera animation paths that are flexible enough to include a variety of parameters into the camera path computations. For example,

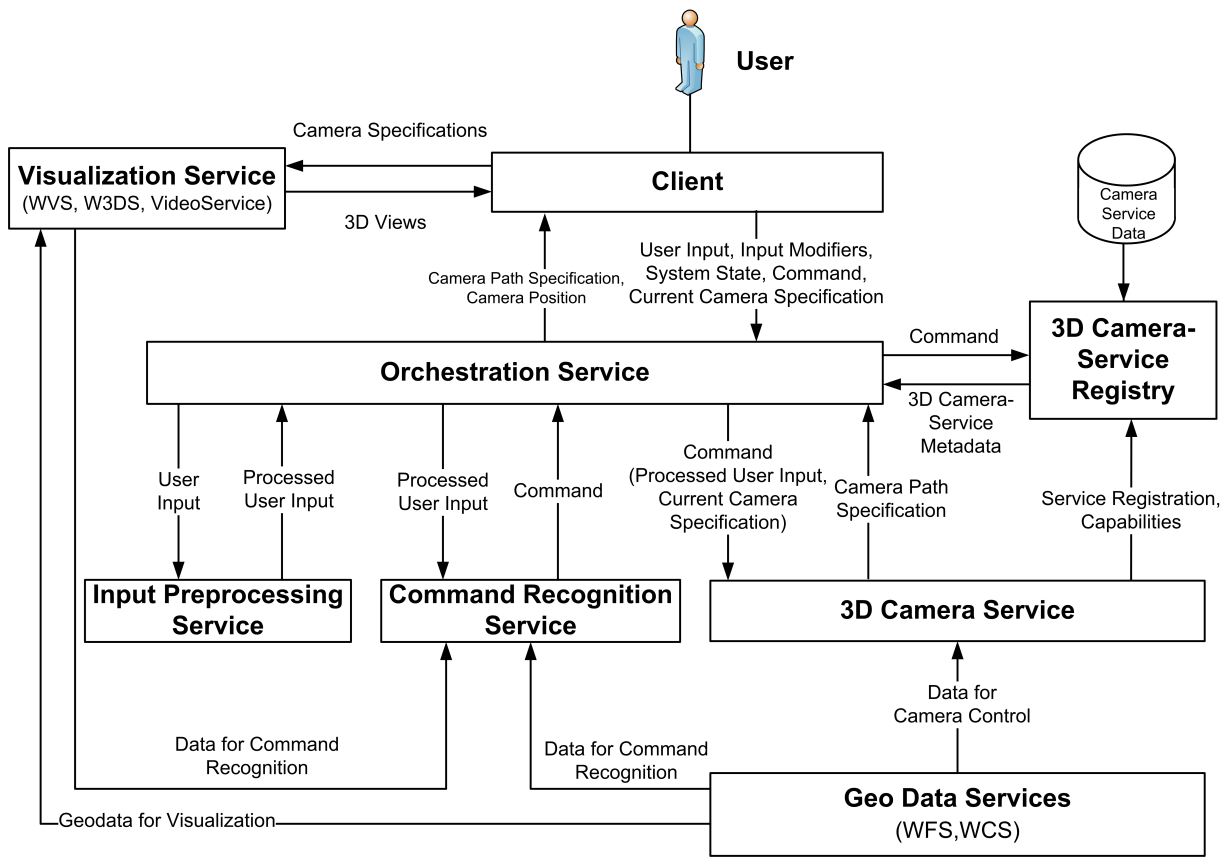


Figure 3: Abstract component architecture and data flow of a geovisualization system using service-based camera navigation

to provide the mentioned context-sensitive camera animations for mobile applications, the available parameters, such as user position, device orientation and acceleration, influence the camera animation. One promising approach for such calculation is using a physics engine to simulate forces that affect the camera and to create a camera animation. Here, questions for my further research may include the type of physical forces (e.g., spring forces, force fields, flow simulation etc.) that can be used to create a camera animation.

Visual feedback for camera navigation Visualization for current or future actions a camera animation will include is essential to keep a user oriented and also to communicate recognized navigation commands to a user of a service-based visualization system. The research question here is how to introduce meta elements into the visualization that communicate clearly the navigation intention that a camera animation executes. Especially in the case of indoor visualization this is an open question in the research community.

Definition of camera control intentions In which way can a user describe complex camera navigation tasks using conventional input devices. This is important to allow reliable recognition of navigation commands and to provide task oriented navigation techniques. Here the central question is, how can semantic data, which is

available when using geodata for visualization, be used to derive such intentions in 3D from actions that are performed on the 2D camera view plane.

Camera services Definition, assessment and implementation of the service-based camera control remains a challenging task. Especially performance considerations together with an analysis of the application potentials and restrictions will show how such a system can be applied. The definition of exchange formats and camera service meta-information for the services introduced in Section 4 is another part of work to be done to enable a loosely coupled system and a camera service registry.

Multi-touch input devices The mapping of the input of tangible displays, which are increasingly used in end-user hardware, to camera navigation intentions is a question that will be in focus. The goal to achieve highly interactive visualizations of massive 3D geodata on small devices demands for more research in this specific direction.

References

- [1] 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. Springer, 2008.
- [2] H. Buchholz, J. Bohnet, and J. Döllner. Smart and Physically-Based Navigation in 3D Geovirtual Environments. In *9th Int. Conf. on Information Visualisation (IV'05)*, pages 629–635. IEEE, 2005.
- [3] J. de la Beaujardiere. OGC Web Map Service Interface, 2004.
- [4] F. Declé and M. Hachet. A Study of Direct Versus Planned 3D Camera Manipulation on Touch-based Mobile Phones. In *Proc. of the 11th Int. Conf. on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '09*, page 1, New York, USA, 2009. ACM Press.
- [5] J. Engel and J. Döllner. Approaches towards visual 3d analysis for digital landscapes and its applications. *Digital Landscape Architecture Proceedings 2009*, pages 33–41, 2009.
- [6] G. Gröger, T. H. Kolbe, A. Czerwinski, and C. Nagel. OpenGIS City Geography Markup Language (CityGML) Encoding Standard Version 1.0.0, 2008.
- [7] B. Hagedorn and J. Döllner. Sketch-Based Navigation in 3D Virtual Environments. In *Proc. of the 9th Int. Symp. on Smart Graphics*, volume 5166 of *Lecture Notes in Computer Science*, pages 239–246, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

- [8] B. Hagedorn, D. Hildebrandt, and J. Döllner. Towards Advanced and Interactive Web Perspective View Services. In *Developments in 3D Geo-Information Sciences*, pages 33–51, Berlin/Heidelberg, 2009. Springer.
- [9] J. McCrae, I. Mordatch, M. Glueck, and A. Khan. Multiscale 3D Navigation. *Proc. of the 2009 Symp. on Interactive 3D Graphics and Games - I3D '09*, page 7, 2009.
- [10] C. Portele. OpenGIS Geography Markup Language (GML) Encoding Standard, Juli 2007.
- [11] P. Schut. OGC Web Processing Service, 2007.
- [12] D.S. Tan, G.G. Robertson, and M Czerwinski. Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, volume pp, pages 418–425. ACM New York, USA, 2001.
- [13] P.A. Vretanos. OGC Web Feature Service Implementation Specification, 2005.