High-Quality Video Generation for Thin Clients - An Application for Image-Based 3D Portrayal Services

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

1 Introduction

3D geodata, such as virtual 3D city models, is currently the basis for a broad range of applications in different areas, such as city marketing or urban planning. The increasing availability, volume, and quality of 3D geodata raises the complexity of 3D visualization applications using such data. Service-based 3D portrayal approaches, either image-based or based on geometry and texture streaming services, provide solutions for interactive exploration of large 3D data sets. But, even if the challenge of provisioning visualizations of large scale 3D models to heterogeneous hardware and software platforms was addressed in recent years, the complexity of navigation remains challenging for individual users. The communication of specific intentions (e.g., a positive impression of certain planning projects in an urban context) in connection with 3D geoinformation demands for specially designed camera paths, rendering effects (e.g., effects for highlighting, focus and context visualization [13], or real time ambient occlusion [1]), and transitions between specific sets of camera and scene configurations. Due to this complexity in operating a visualization system for end users, production of video presentations of large 3D models are still a usual way of communicating spatial information. Still images are not well suited to communicate sufficient amounts of spatial information since “the acquisition of spatial knowledge, essential for wayfinding, is primarily based on direct environmental experience, which is usually gained via movement“ [5]. The advantage of preproduced video presentations is its compatibility (virtually any computing platform used by end users is able to play common video formats), the simplicity in playback (in terms of necessary hardware, software, and network resources for their transmission and playback), and the assured appearance of the 3D scene contained in the video presentation.

In recent years, several approaches have been presented for interactive presentation of 3D geodata in several scenarios, in stationary (e.g., on desktop PCs or notebooks) and also in mobile scenarios (interactive 3D visualization on hand-held devices, such as tablet PCs or mobile phones). Nevertheless, producing distributable artifacts for communication of 3D geodata, specifically video presentations, are currently mainly a manual, and therefore, resource intensive task. Generation of video presentations usually includes involves two major steps: a) creation of planned flythrough sequences and b) post production and cutting of the assembled video presentation. An overview over the usual process can be found in Figure 1.

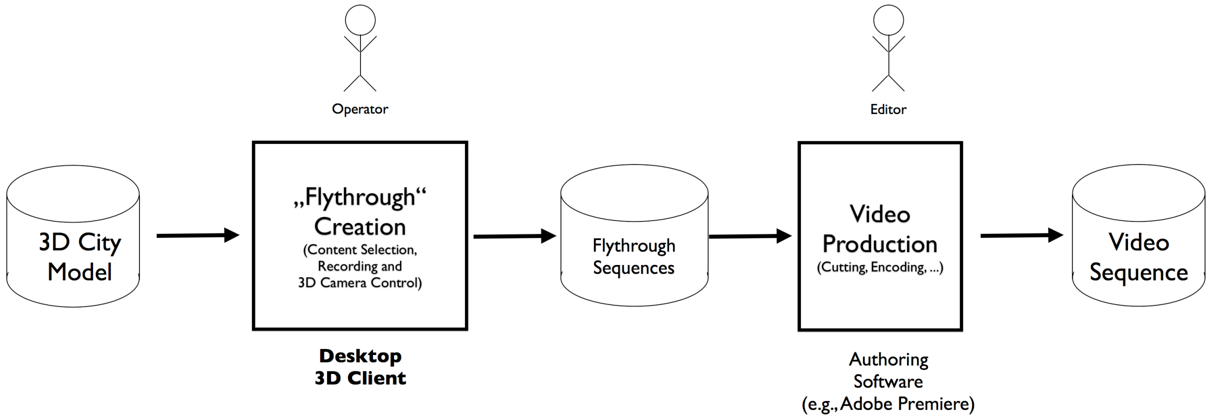

Figure 1: Overview over a conventional process for generating video presentations of 3D data. There is a lot of manual work involved. The virtual camera is controlled manually during video capturing for flythroughs. This makes the hole generation process costly and hard to repeat even if there are just minor changes in settings or scene content.

Since both of the two production steps involve high degrees of manual work, a repeated production with only minor changes in the 3D scene (e.g., an updated city model or a change in planning models) is an expensive task. We propose to automate the overall generation process by providing a web service that implements this process. The service takes a video description through its interface and automatically creates video presentations using image-based portrayal services, namely Web View Services (WVS) [6]. The video descriptions can be stored and reused, which facilitates to recreate videos, e.g., with equal cutting and camera path but different versions of the underlying data. Further, if underlying services are improved (e.g., new rendering techniques that provide an improved visual quality for 3D renderings) it is very easy to recreate video presentations with little or no costs, since no additional manual work for video clip recording or post production is necessary. In this way, we provide a high-level service that is able to provide additional value for owners of 3D city model data and proof that these Web View Services provide an important part of 3D geodata infrastructures [2].

The remainder of this report is organized as follows. Section 2 provides an overview over the work related work. Section 3 describes the designed system, its components, and the current implementation status. Finally, an outlook of planned future work concludes this report in Section 4.

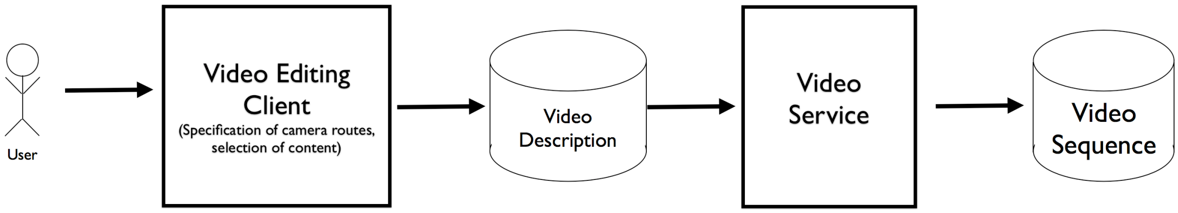

Figure 2: Overview over an optimized process for generating video presentations of 3D data. Video descriptions are created using a dedicated client application that also triggers the video rendering process. Descriptions can be stored and reused to automatically create video presentations using, e.g., extended or updated data or improved image-based portrayal services.

2 Related Work

Previously, we presented solutions to distribute 3D geoinformation to different devices and platforms (focused especially on web browsers and mobile devices) in homogeneous quality [3, 10]. Thes systems are based on a Web View Service [6] as image-based 3D portrayal service encapsulating processing, management, and 3D rendering of massive amounts of 3D geodata. Thin clients query interactive these services and provide partial reconstructions of the 3D scene on the client side. This way, users are able to interactively explore 3D geovirtual environments, which is important to gain understanding of the 3D information. Nevertheless, the systems introduce so far are missing a possibility to create artifacts that can be distributed easily. Currently, the only articfacts that can be created using these client applications are static image. For the work presented in this paper, we utilize the technology that has been developed before to build a video editing client that can be used to define video presentation descriptions using a variety of client platforms.

Currently the topic of automatic, distributed generation of video presentations from 3D data has primarily been addressed in the area of medical visualization. Iserhardt-Bauer et al. introduce a system that encapsulates rendering of datasets acquired from CT scanners [9]. They encapsulate hardware rendering behind a service interface in to enable users to analyze the scan data without having to have specialized, powerful graphics hardware available in end user devices. Unless our approach, the do not support composite video presentations that are assembled from different, specifically user designed camera path sequences and auxiliary sequences containing text or images. The approach supports a standardized camera path only that cannot be customized. Further the approach is limited to the very specific data generated by CT scans. In contrast, our approach provides an abstraction over the underlying data by using service-based rendering for image generation. Roßler et al. extended this approach by using a conventional GPU-based PC cluster [12]. As like our architecture, they use a kind of manager process to perform dispatching and other supporting tasks.

Further, there are several approaches that are primarily targeted at bringing high-end graphics to low-end devices. E.g., Lamberti et al. perform remote rendering of complex 3D objects, e.g., containing several millions of voxels or textured polygons, and stream the result encoded as video to clients [11]. This approach leaves the configuration of camera parameters to clients, which are able to adjust the camera settings interactively. It is designed to support interactive rendering of video streams and does not provide a service interface for definition of complex video presentations as our approach does.

3 A Service-Based System for Automated Generation of Video Presentations

Generating video presentations of large scale 3D datasets is a resource intensive task. We system described in this section allows to decompose the task of video generation into reusable service components for definition of video presentations, image generation from 3D data, and video clip generation. The system is based on Web View Services (WVS) for image generation introduced earlier [7] that is currently in a standardization process within the Open Geospatial Consortium (OGC). Since we want the video service to be as responsive also during the generation process for single video presentations, we created an asynchronous process to generate video presentations. We separated the generation process into two services components: a) A frontend evaluating and validating service requests and b) a video rendering component that implements the more computationally expensive task of video clip generation. In this way, clients do not receive a video presentation as response to their service request, but the generation status is signaled using a callback mechanism (either via email messages or calls to a remote URL).

3.1 Service Components

There are five main components within the system (see Figure 3 for an overview of the framework architecture):

Video Editing Client The client application provides the user interface to specify video sequences including scene contents, camera paths, and the transitions between them. This way, we focus the user interaction on the one tasks of defining the complete video presentation instead of distributing it into a clip recording and a postproduction phase. Since the representation of video presentations in terms of the service interface (see Section 3.2 for details) does not necessarily correlate with the process of defining such presentations, the client application is built upon a cross platform library providing utility functions, e.g., for creating camera paths or defining transition screens. The client is able to store the video projects in a database allowing for reproduction and adaption of videos that have been created once.

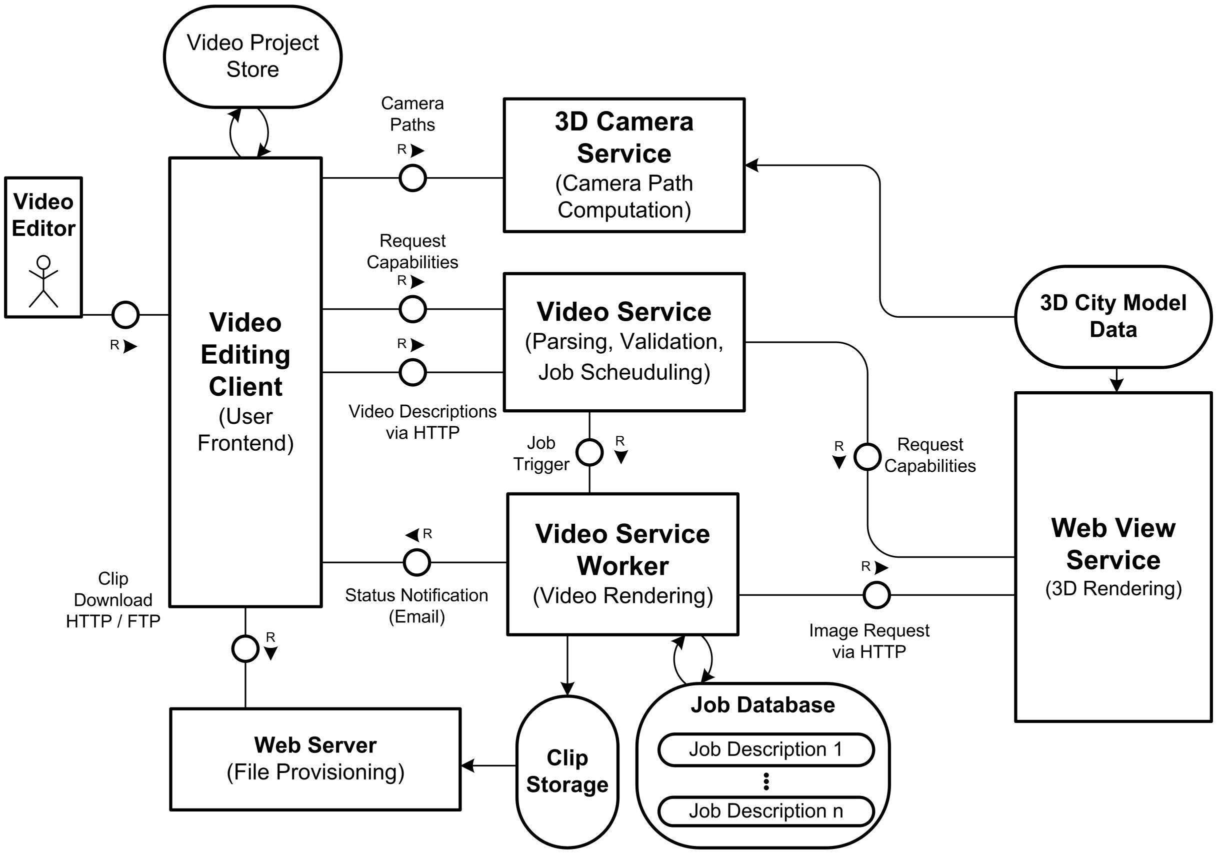
Video Project Store
Video Editor
Video Editing Client (User Frontend)
Camera Paths
R
3D Camera Service (Camera Path Computation)
Request Capabilities
Video Service (Parsing, Validation, Job Scheuduling)
Video Descriptions via HTTP
Job Trigger
3D City Model Data
Request Capabilities
Web View Service (3D Rendering)
Video Service Worker (Video Rendering)
Status Notification (Email)
Image Request via HTTP
Clip Download HTTP / FTP
Web Server (File Provisioning)
Clip Storage
Job Database
Job Description 1
Job Description n

Figure 3: Components of the service-based system for video clip generation from 3D city model data. A video editing client is used to determine single clips, their camera paths (supported by 3D Camera Services) and scene content, and their interconnection. Project descriptions are stored in a database for easy altering and reproduction of video presentations. A Video Service instance utilizes Web View Services to retrieve single video frames and combines them to video sequences.

3D Camera Service 3D camera services are utility services in 3D geodata infrastructures encapsulating the complexity of 3D camera path computation in complex 3D environments such as virtual 3D city models [10]. The Video Editing Client can utilize such services in order to support users in defining visually appealing camera transitions. A 3D camera service is able to deliver proposals for such camera paths, which can be customized by video editors in order to fulfill their navigation intentions.

Web View Service Web View Services encapsulate image generation from 3D data. Each service provides portrayal capabilities for a specific set of data or for a specific spatial region. By utilizing this service-based approach for rendering of 3D geodata, the video service can be implemented without having to deal with the complexity of geodata visualization. Several WVS instances can be used in one video presentation, enabling the combination of different data providers, each providing its own WVS instance with its own content. This allows to use 3D geodata for visualization purposes, without transferring the geodata itself, which might be confidential or their access might be subject to a charge.

Video Service The Video Service serves as a frontend providing the service interface described in Section 3.2 to service consumers. Requests are parsed and validated by this component. If a valid request was issued, a job for the Video Service Worker is created and a confirmation is sent back to the service consumer. The video service can also implement further application logic for authentication of service consumers as well as functionality for billing.

Since video frame generation, especially for longer running video presentations, generates a large amount of load to underlying portrayal services. Here, a Video Service has the possibility to schedule video generation jobs to specfic low traffic hours. Depending on the priority of the video service request, a generation task can be scheduled immediately or can be postponed to be processed later when enough service capacity is available.

Video Service Worker The worker performs the video rendering from single frames that have been previously fetched from a WVS (in case of images of a 3D scene) or generated by the service itself (e.g., overlay screens containing text or images). For camera paths, the camera position and orientation is interpolated for every frame from given camera paths. If two adjacent camera path sequences exist, the service is able to connect the two paths so that no visible jumps in camera paths are visible. Videos are often designed to communicate certain intentions. The worker service supports this by providing possibilities to show intermediate screen containing text and images as well as screen overlays that are shown over 3D flythrough sequences. Such overlays usually contain logos (e.g., for project specific logos or contractor logos). The frames or the necessary parts of them are rendered locally by the Video Service Worker. Since all 3D rendering techniques are implemented as web-based portrayal service, a worker implementation does not need hardware acceleration or any special hardware setup. This way, a Video Service Worker can be deployed easily, also as cloud-based service.

3.2 Service Interface

We define a model for describing video presentations that is used for generating through a `GetVideo` request allows to specify complex composite video presentations (see Figure 4 for an object model). A `Video` has certain settings that are valid for the overall presentation: The resolution of the video and a string defining the encoding (codec) to be used. Further there are default settings for a video presentation that can be overwritten by single sub elements of the document (e.g., specific point definitions or camera path sequences). The spatial reference system (SRS) and the default scene contents (defined as layer in WVS interface terms) are examples for such properties. A video presentations consists of one or more video `sequences`, that are single parts of the video presentation with an assigned duration. We distinguish two types of video sequences: `CameraPath` sequences are flythrough video clips that are based on a camera path, that can be defined either explicitly by providing path geometry or implicitly, e.g, by defining camera tasks like rotation around a scene object.

The other kind of sequences are `Screen`s that either show textual content or images for a certain duration. `Sequence` objects area connected by `SequenceTransition` definitions. Here, the service needs to expose the types of sequence transitions it supports. There are two types of sequence transitions: a) Image based transitions, such as different blending techniques that can connect all types of video sequences and b) `CameraPathTransitions` that interpolate camera paths in order to connect them without having non continuous camera paths. The transitions are assigned implicitly to sequences using correlating indices.

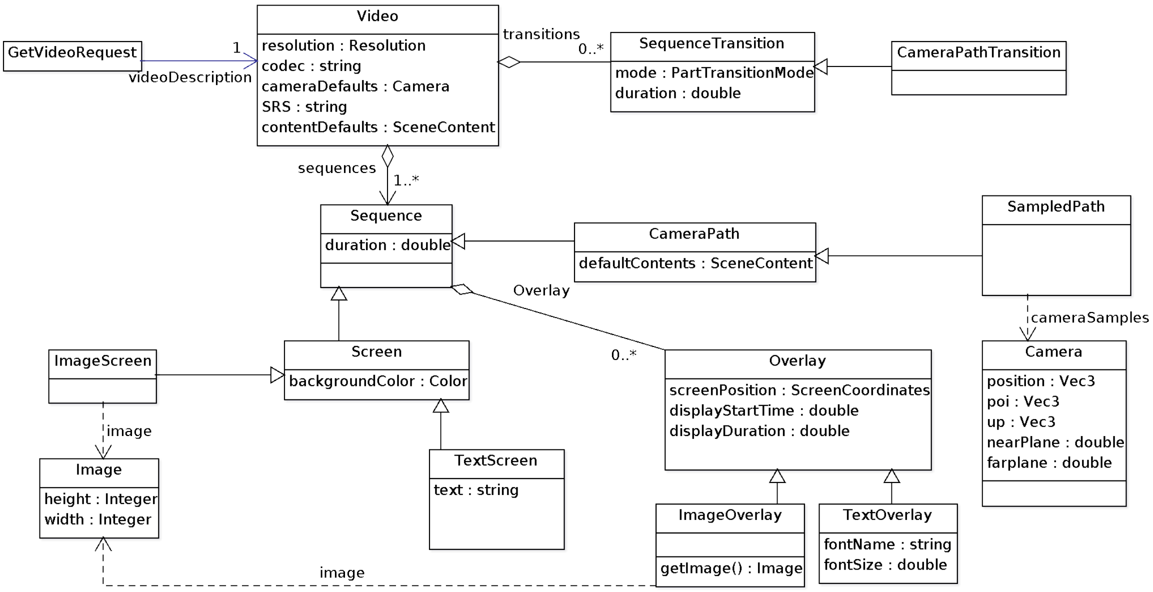

Figure 4: Excerpt of the object model for describing video presentations in a service request. A video contains one or more sequences, which may either be a camera path through a 3D scene or a screen (either an image or text). Two sequences are connected through sequence transitions that configure the blending properties and the interpolation between two adjacent camera paths.

Video generation requests are currently encoded using the JavaScript Object Notation (JSON).

3.3 State of the Work

Currently the implementation of the video generation system is work in progress. We implemented a first version of the video generation process including a basic Video Service and a Video Service Worker.

The next steps we are currently working on is to implement a platform independent client library that defines a process for defining video service descriptions. Our client is currently build for tablet PCs based on the iOS operating system. Its visualization component is following an image based approach introduced earlier based on the WVS we are running [3]. The base data we are currently working with is the virtual 3D city model of the city of Berlin, which is one of the largest, highly detailed, and fully textured 3D city models that are available world wide. At the moment, the GetCapabilties operation of the video service is not yet implemented, so the service is not yet self descriptive. The supported content layers are always dependent

4 Conclusions and Future Work

The implementation of a video production process that supports nearly arbitrary WVS image sources massively facilitates the generation of video presentations in the context of virtual 3D city model. The amount of manual effort, especially for regeneration of video presentations due to updated data, is reduced significantly through the automatic, repeatable generation of video presentations. Using image-based portrayal services to encapsulate complex 3D geodata management and 3D rendering provides a major advantage since a video service can be implemented in a generic manner using the well defined WVS interface specification, which is going to be an OGC standard for image-based portrayal of 3D geodata.

Currently image-based styling and post processing of frames is performed separately for each WVS that is queried for images. This can cause an inhomogeneous appearance, since the is no standard way of implementing and configuring image-based rendering effects, such as global illumination [4]. Therefore, as proposed by Hildebrandt [8], image-based styling could be externalized to a separate styling service that processes single frames using the same implementations for image-based styling for all sources.

The client for specification of 3D video presentation descriptions offers further possibilities to explore 3D interaction techniques that are specfically designed to assist users in specifying scene contents, 3D camera paths, and sequence transitions. Further, the client implementations as well as the video service implementation can be further generalized to configure themselves according to `GetCapabilties` documents delivered by WVS instances and the other services involved in the process.

References

[1] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-Time Rendering*. A. K. Peters, Ltd., Natick, MA, USA, 3rd edition, 2008.

[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] J. Doellner, B. Hagedorn, and J. 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.

[4] W. Engel, editor. *GPU Pro 4: Advanced Rendering Techniques*. Routledge Chapman & Hall, 2013.

[5] Nathan Gale, Reginald G. Golledge, James W. Pellegrino, and Sally Doherty. The acquisition and integration of route knowledge in an unfamiliar neighborhood. *Journal of Environmental Psychology*, 10(1):3–25, March 1990.

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

[7] 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.

[8] D. Hildebrandt. Towards Service-Oriented, Standards- and Image-Based Styling of 3D Geovirtual Environments. In *Proceedings of the 5th Ph.D. Retreat of the HPI Research School on Service-oriented Systems Engineering*, pages 133–147, 2010.

[9] S. Iserhardt-Bauer, P. Hastreiter, T. Ertl, K. Eberhardt, and B. Tomandl. Case study: Medical Web Service for the Automatic 3D Documentation for Neuroradiological Diagnosis. In *VIS '01 Proceedings of the conference on Visualization '01*, pages 425–428, Washington, DC, 2001. IEEE Computer Society.

[10] J. Klimke, B. Hagedorn, and J. Doellner. A Service-Oriented Platform for Interactive 3D Web Mapping. In M. Jobst, editor, *Service-Oriented Mapping*, pages 127–139, Vienna, 2012.

[11] 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.

[12] Friedemann a. Röß ler, Torsten Wolff, Sabine Iserhardt-Bauer, Bernd Tomandl, Peter Hastreiter, and Thomas Ertl. Distributed Video Generation on a GPU-Cluster for the Web-Based Analysis of Medical Image Data. In Kevin R. Cleary and Michael I. Miga, editors, *Medical Imaging 2007: Visualization and Image-Guided Procedures*, pages 650903–650903–9, March 2007.

[13] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 3rd revise edition, 2012.