draft

An Approach towards Semantics-Based Navigation in 3D City Models on Mobile Devices

Jürgen Döllner, Benjamin Hagedorn, Steffen Schmidt

University of Potsdam – Hasso-Plattner-Institut, Helmert Str. 2-3, 14482 Potsdam, Germany
Phone ++49 331 5509 170
Fax ++49 331 5509 179
Email {juergen.doellner, benjamin.hagedorn, steffen.schmidt}@hpi.uni-potsdam.de

Abstract

This paper outlines a novel approach for user navigation in complex virtual 3D city models on mobile devices. Users navigate within the virtual 3D city model by sketching navigation commands in the perspective view on the mobile client. The sketches are sent to the server, which reprojects the sketches onto the 3D scene, interprets these sketches in terms of navigation commands, and sends the resulting video-encoded image stream to the mobile client. This approach allows us to provide interactivity for complex virtual 3D city models on resource and bandwidth limited mobile clients. A high degree of usability is achieved because users can trigger complex navigation commands in a task and goal oriented way taking advantage of the navigation properties and affordances inherent to elements of geovirtual environments.

Keywords: Mobile Mapping, Navigation Systems, Geovisualization, Geovirtual Environments, Virtual Reality.

1 Motivation

Virtual 3D city models represent urban spatial and geo-referenced data by 3D geovirtual environments (GeoVE) that include terrain models, building models, vegetation models as well as models of roads and transportation systems. In general, these models serve to present, explore, analyze, and manage these urban information spaces and, therefore, constitute a major user-interface paradigm for 3D geoinformation systems.

An increasing number of applications and systems incorporate virtual 3D city models as essential system components such as for facility management, logistics, security, telecommunication, disaster management, location-based services, real estate portals as well as entertainment and education products. Consequently, a large number of potential users and usages require an efficient and effective mobile access to virtual 3D city models and their contents.

We present a novel solution for accessing virtual 3D city models on mobile devices. The user controls the navigation within the virtual 3D city model by navigation command sketches drawn directly on the view-plane of the mobile client (Fig. 1). The sketches are sent to the server, which reprojects the sketches onto the 3D scene correlating the sketches to scene objects, interprets these sketches in terms of navigation commands, and sends the resulting video-encoded image stream to the mobile client. That is, the mobile client enables users to specify and retrieve step-by-step created video sequences that correspond to their navigation intentions.

Fig. 1: Sketching the navigation command “look around” (left). Sketching the navigation command “walk along the path and, finally, look at the indicated building” (right).

2 Related Work and Challenges of Mobile 3D City Models

Mobile applications of virtual 3D city models represent a major and complex research challenge due to limited bandwidth and graphics capabilities, restricted interaction capabilities, data standardizations and distribution techniques, and digital rights issues.

2.1 Mobile 3D Rendering

In 3D computer graphics, numerous rendering techniques are available to cope with complex virtual environments, including discrete and continuous multi-resolution geometry and texture representations, view-frustum culling, occlusion culling, imposter techniques, and scene-graph optimizations (Akenine-Möller and Haines 2002). Virtual 3D city model visualizations require an efficient management of large-scale texture data, e.g., for aerial photography and building facades (Buchholz and Döllner 2005), and level-of-detail management for large heterogeneous 3D object collections (Davis et al. 1999) and 3D terrain surfaces (Döllner et al. 2000). Although these rendering techniques enable real-time rendering of complex 3D scenes, they generally cannot be transferred directly on mobile devices due to limited computational resources and power.

One principal approach to efficient *mobile 3D rendering* consists in the adaptive, progressive, and compressed transmission of 3D graphics data to mobile clients. For example, Royan et al. (2003) describe client-server architecture for mobile 3D virtual city visualizations based on a progressive and hierarchical representation for GeoVEs. The server pre-computes multi-resolution representations of terrain models and building models, and progressively sends data about visible areas to the mobile clients. The clients allow users to interact with the 3D city model (e.g., virtual walk-throughs, fly-overs, etc.). However, the limited 3D graphics acceleration on today’s mobile devices makes it difficult to implement fully featured 3D rendering techniques for virtual 3D city models. Furthermore, the implementation is complicated due to the broad variety of hardware and software solutions for mobile 3D graphics (e.g., OpenGL ES, Mobile 3D Graphics API for J2ME).

Another principle solution consists in *server-side 3D rendering* and the progressive, compressed transmission of image sequences. For example, Cheng et al. (2004) investigate a client-server approach for visualizing complex 3D models on thin clients applying real-time MPEG-4 streaming to compress, transmit, and visualize rendered image sequences. They identify the MPEG-4 encoding speed as bottleneck of client-server 3D rendering, and devise a fast motion estimation process for the MPEG-4 encoding process.

2.2 Mobile 3D Interaction

To achieve a high degree of usability, mobile applications require goal-oriented and task-oriented interaction techniques that take into account the specific restrictions of mobile devices, e.g., no mouse, no desk, or one-handed operation. For this reason, approaches for automating user interaction are crucial for effective mobile user interaction. Of course, these approaches are also faced with the general problems of navigating in virtual worlds (Russo et al. 2000).

A critical task in applications of GeoVE represents the process of navigation, "whereby people determine where they are, where everything else is, and how to get to particular objects or places" (Jul and Furnas 1997). Navigation as the primary interaction can be distinguished into three kinds, naive search, targeted search, and exploration (Darken and Sibert 1996) and serves to explore, analyze, and gather geoinformation as well as to trigger object-specific interaction. To do this, users generally move the virtual camera or an avatar through the GeoVE. This way the user builds up a mental model of the GeoVE by forming linear maps and combining them to spatial maps (Ingram and Benford 1995). Wernert and Hanson (1999) incorporate task-based constraints on the navigation parameters (e.g., viewer position and orientation) to enable the designer of GeoVE "to provide extra assistance to keep the user's explorational wanderings and attention focused on the task objectives".

Common navigation controls for GeoVE include world-in-hand controls, fly-over controls, and virtual trackballs. Burtnyk et al. (2002) introduce a general approach of facilitating navigation in GeoVE based on explicitly designed navigation spaces using integrated spatial and temporal controls. Buchholz et al. (2005) describe a concept of smart and physically-based navigation techniques, controlling the user's movement similar to an assistance system preserving users from being disoriented or getting lost in the GeoVE. It constrains the movements to be inside the GeoVE, hinders collisions with buildings, controls the gaze direction at the terrain borders, and facilitates the switch between navigation modes. For mobile applications, semantics-based navigation control can integrate and extend these concepts.

Igarashi et al. (1998) develop an intuitive approach for specifying navigation commands: The user draws the intended navigation path as a curve on the view plane. This path is mapped to the 3D scene and determines the 3D path the avatar moves along. This way, the user can specify not only the final position, but also the route and the camera direction at the goal with a single stroke. Our approach also has been motivated by the metaphor-aware 3D navigation technique (Russo et al. 2000) and specialized for virtual 3D city models.

2.3 Standardization and Distribution

Applications of virtual 3D city models also suffer from a lack of data standards and flexible distribution techniques. Virtual 3D city models frequently are implemented as graphical models without explicitly modeling semantic and topological relations. Therefore, they can almost only be used for visualization purposes but not as a data basis for higher-level functionality such as simulations, analysis tasks, or spatial data mining. The limited reusability and interoperability inhibits the broader use of virtual 3D city models. CityGML represents a first XML and GML-based format for storing and exchanging virtual 3D city models (Kolbe et al. 2005), which also represents semantic and thematic properties, taxonomies and aggregations.

With respect to distribution, a complete delivery of city model data would result in massive data transfers. Even if only a part of a complex virtual 3D city model is required (e.g., view-dependent multiresolution selections), the costs for geometry and texture data for high-quality photorealistic models typically exceed current and future transmission capabilities.

2.4 Digital Rights Management

Protecting the contents of 3D city models is one of the most critical aspects of real-world business models underlying 3D city model applications (Döllner 2005). The transmission of raw city model data or derived detailed 3D graphics data imply severe drawbacks for copyright issues and controlling usage and distribution. For this reason, we transmit only video sequences but no raw data to the mobile clients.

3 Sketch-Based Navigation Commands

3.1 Real-Time Interaction vs. Selective Interaction

Common navigation techniques allow users to control their movement within the virtual environment in real-time. For mobile devices, however, real-time 3D rendering of virtual 3D city models is not practically possible due to limited computation resources and bandwidth as well as the non-stable data transmission. In contrast to the real-time user reaction that characterizes most games taking place in virtual environments, real-time 3D interaction is not crucial for many applications and systems of virtual 3D city models because exploration and analysis tasks performed by users are based on a selective, targeted access of spatial information. That is, the delay between issuing interaction commands on the mobile device and the execution of the commands is acceptable and corresponds to the expectation of the user.

3.2 Concepts of Sketch-Based Navigation

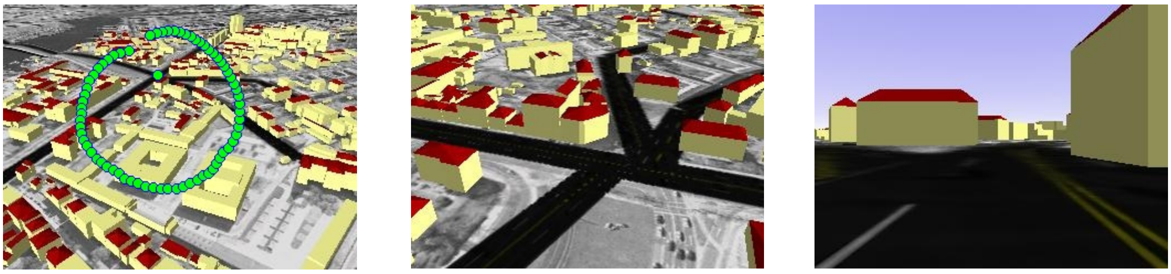
In our approach, navigation commands are graphically specified in the perspective view of the virtual 3D city model on the mobile client, e.g., drawing a line along a street, pointing to a building or the sky. The sketches are correlated with the objects of the GeoVE by reprojecting the sketched shapes onto the 3D scene. The sketch-based navigation commands are interpreted based on the semantics of sketch-correlated objects and their inherent navigation affordances.

We distinguish between three types of information used for interpreting the commands:

- *Spatial context*: The spatial context refers to the virtual location to which the sketch is aligned or associated. For example, the user can draw a path along a street or mark a specific building.
- *Temporal context*: The temporal context refers to the order in which the user composes the sketch elements. For example, as first step the user draws a path along a street, and then marks the building.
- *Sketch geometry*: The elements include points, lines, and polygons drawn in the perspective view. They can be grouped and interpreted by higher-level sketch geometry such as circle-like paths or u-like paths.

The sketches can be differentiated into *location-aware sketches* and *gestures*. Location-aware sketches refer to a spatial context, whereas gestures do not. From a technical point of view, gesture recognition requires large, screen-wide drawings for correct identification. For example, a circle gesture cannot be drawn close to the corners of the screen. Gestures are known from computer games, from several navigation-aware applications, or as utility programs that can be used for desktop interaction.

We allow for concatenating and building up a temporal context for the navigation command sketches. For high usability and consistency of the user interface it must be considered that sketches might a) represent a place to go to or a path to go along, b) mean one ore more points to gaze at, or c) conclude both, place and direction of view. By combining gestures with other sketches we can introduce spatial context to gestures, too (Fig. 2).



a) b) c)

Fig. 2: The user sketches a point on a junction and adds a circle-like gesture as secondary input (a). In the resulting animation, the camera sinks down to the marked junction (b) and then performs a turn-around (c).

3.3 Sketch-Based Navigation Vocabulary

A first collection of spatial and temporal contexts together with sketch elements is illustrated in Table 1. Gabbard (1997) points out that "when assessing metaphors for navigation and locomotion in VEs, it is important to consider mappings of integral navigation and locomotion components to metaphor gestures or mechanisms." The sketch-based navigation commands within their spatial and temporal context provide such mechanisms. For example, drawing a single, straight path along a street object indicates, "walk along the street". A circle-like (close or nearly closed) path drawn on the terrain surface indicates "look around" using a drawn point as camera position. A path drawn along a street with a final indicated u-turn indicates, "walk along to the end of the street, turn around, and walk back".

Table 1: Overview of sketch-based navigation commands.

Name & Context	Example Sketch	Navigation Sketch	Navigation Action
Point-House		Point on a building.	Finding shortest path to the building, going there, and looking at the building.
Point-Roof		Point on a building's roof.	Flying up to the roof, placing the camera on top, and looking around.
Curve-Street		Curve or polygon on a street.	Walking on street and looking back finally.

Curve-Street, Point-House		Curve on a street and point on a building.	Walking along the street and looking at the building finally.
Point-Ground, Point-House		Point on the ground and point on a building.	Flying to the marked ground point and looking at the building finally.
Point-Street, Circle Gesture		Point on the street and circle-shaped gesture.	Flying to the marked ground point and looking around.
Point-Sky		Point on the sky.	Soaring above ground for overview.

4 Client-Server Architecture for 3D Visualization

The presented approach has been implemented based on a client-server architecture outlined in Fig. 3. We assume that the 3D city model is hosted on the server and that the mobile devices efficiently support encoded video streams.

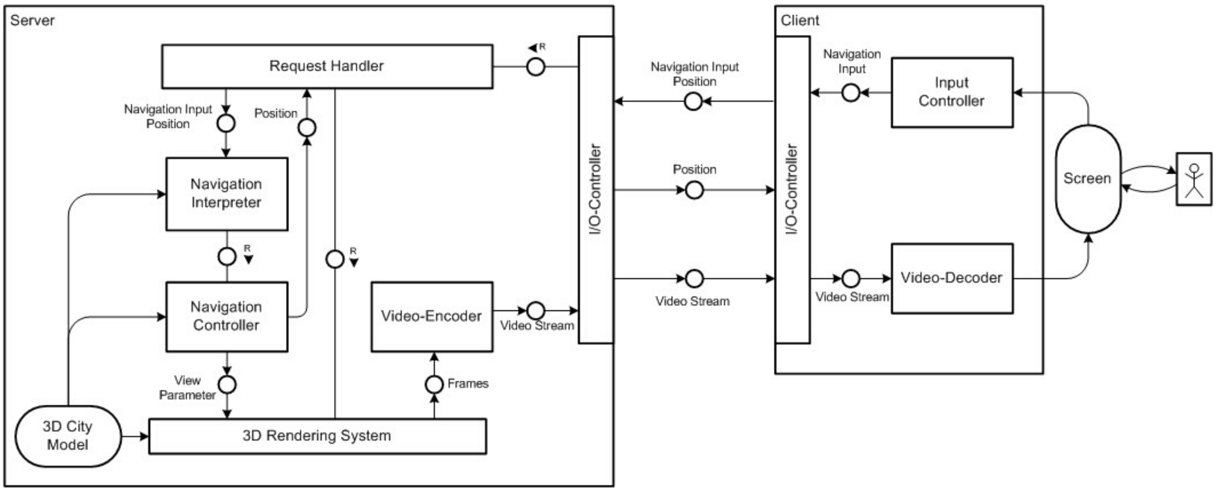


Fig. 3: Client-server architecture of our system for mobile access to virtual 3D city models.

4.1 Server System

The server system is responsible for handling requests sent from the clients, for interpreting and controlling sketch-based navigation commands, for 3D rendering, and the video-stream encoding. It provides a web service interface to the virtual 3D city model. The clients can communicate with the server by exchanging SOAP messages. The interface supports three main operations:

- *GetCapabilities*: Provides the service metadata including information about the used streaming protocol and available start positions. The clients call the operation at the beginning of the communication.
- *GetStartPosition:* Renders and transmits an image of the start position. This image provides the spatial context for the user's first navigation inputs.

- *GetMotion:* Interprets sketch-based navigation commands and initiates the rendering of the camera animation. Because the server is stateless, it has to reconstruct what the user of the client saw while drawing the sketches. Therefore, the request contains the final camera position of the preceding request. If the user stopped before the end of the animation for a new input, the client's position can be determined by the start position and inputs of the previous navigation and the point in time the user stopped the video. Both camera positions, at the beginning and the end of navigation, are included in the response message.

The *navigation interpreter* detects the semantics of sketch-correlated scene elements and identifies the classes to which hit objects belong. For a sketch that has more than one input point we determine which object type occurs most frequently such as in the case of a path on a street whereby not all of the input points are placed exactly. The *navigation controller* calculates the resulting animations (Christianson et al. 1996; Mackinlay et al. 1990). Special navigation controllers use a navigation network geometry that provides paths that can be used to walk along. The position to look at a house is determined as the nearest point on such a path element. For an effective 3D overview for the rise-to-the-sky navigation we provide a map of view directions as introduced by Hanson and Wernert (1997). It allows us to determine a suitable direction to look at from a specific point to gain as much spatial information as possible. The current implementation is based on height-defined landmarks. The *rendering component* encodes the animation frames into an MPEG-4 video stream (Cheng et al. 2004; Noimark and Cohen-Or 2003) by the *video encoder*. The resulting video stream is transmitted to the client immediately using a standard streaming protocol. So, the client can start the video playback as soon as possible.

We have implemented and tested a server that uses DIME attachments to deliver the video to the client. The DIME standard is similar to MIME and defines a way to send arbitrary binary data along with SOAP messages. Because the data is sent in chunked data blocks, it allows starting the streaming of the produced video while the rendering process has not been completed. Instead of DIME, any other streaming protocol could be used. In this case, the server's response message must include the parameters necessary to connect to the streaming protocol or server.

4.2 Client System

The thin client system does not contain application logic, it only needs capabilities for receiving and playing the MPEG-4 video streams, capturing the user input and sending and receiving SOAP messages. While receiving a video stream, the client simultaneously decodes and displays the video. Most mobile devices provide built-in support for these tasks. For drawing new navigation sketches, the user can wait for the end of the video or he can stop it at any point of time.

The client records the user inputs as a set of 2D points representing the screen coordinates of the navigation sketches. The temporal context of the input can be determined by the drawing order of the points and the classification of single sketches.

5 Conclusions

We have presented an approach towards semantics-based navigation control for mobile virtual 3D city models. A high degree of usability is achieved because sketch-based navigation commands allow users to trigger effectively complex navigation intentions taking advantage of the navigation properties and affordances inherent to elements of geovirtual environments. In addition, it is perfectly suited for the input devices and usage situation of mobile devices where generally no mouse and no desktop can be assumed. From a technical perspective, the presented approach allows mobile applications to provide users interactive access to complex 3D city models including high-resolution 3D terrain geometry, 3D building geometry, and textures exceeding several hundreds of GB of storage. In particular, the server can be optimized for processing large-scale 3D city models using high-end computer graphics hardware, whereas only multimedia capabilities are required from the mobile client.

In our future work, we will address general sketch-based interaction commands and visual feedbacks about the automated navigation. We also plan to include in the video stream meta-information about the scene and its objects.

References

T. Akenine-Möller and E. Haines, "Real-Time Rendering", 2^{nd} Ed., A K Peters, 2002.

H. Buchholz, J. Bohnet, and J. Döllner, "Smart and Physically-Based Navigation in 3D Geovirtual Environments", IEEE Information Visualisation, London, 2005.

H. Buchholz and J. Döllner, "View-Dependent Rendering of Multiresolution Texture Atlases", Proc. IEEE Visualization, IEEE CS Press, 2005, to appear.

N. Burtnyk, A. Khan, G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach, "StyleCam: Interactive Stylized 3D Navigation using Integrated Spatial & Temporal Controls", Proceedings of the ACM Symposium on User Interface Software and Technology, 2002, pp. 101-110.

L. Cheng, A. Bhushan, R. Pajarola, and M. El Zarki, "Real-Time 3D Graphics Streaming using MPEG-4", BroadWise '04, San Jose, CA, USA, July 2004.

D.B. Christianson, S.E. Anderson, L. He, D.H. Salesin, D.S. Weld, and M.F. Cohen, "Declarative Camera Control for Automatic Cinematography", Proceedings of AAAI '96 (Portland, OR), 1996, pp. 148-155.

R.P. Darken and J.L. Sibert, "Wayfinding Strategies and Behaviors in Large Virtual Worlds", Proceedings of ACM SIGCHI 96, 1996, pp. 142-149.

D. Davis, W. Ribarsky, T.Y. Jiang, N. Faust, and S. Ho, "Real-Time Visualization of Scalably Large Collections of Heterogeneous Objects", Proceedings of IEEE Visualization 1999, pp. 437-440.

J. Döllner, "Constraints as Means of Controlling Usage of Geovirtual Environments", Cartography and Geographic Information Science, 2005, 32(2):69-79.

J. Döllner, K. Baumann, and K. Hinrichs, "Texturing Techniques for Terrain Visualization", Proc. IEEE Visualization 2000, IEEE CS Press, 2000, pp. 207-234.

J.L. Gabbard, "A Taxonomy of Usability Characteristics in Virtual Environments", Technical Report, Virginia Polytechnic Institute and State University, 1997.

A.J. Hanson and E.A. Wernert, "Constrained 3D Navigation with 2D Controllers", IEEE, Proceedings on the Conference on Visualization, 1997, pp. 175-183.

T. Igarashi, R. Kadobayashi, K. Mase, and H. Tanaka, "Path Drawing for 3D Walkthrough", 11th Annual Symposium on User Interface Software and Technology, ACM UIST'98, San Francisco, November 1998, pp.173-174.

R. Ingram and S. Benford, "Legability Enhancement for Information Visualization", Proceedings of IEEE Visualization 1995, pp. 209-216.

S. Jul and G.W. Furnas, "Navigation in Electronic Worlds: A CHI 97 Workshop", ACM SIGCHI Bulletin, Vol. 29, No. 4, October 1997, pp. 44-49.

T.H. Kolbe, G. Gröger, and L. Plümer, "CityGML - Interoperable Access to 3D City Models", International Symposium on Geoinformation for Disaster Management, 2005.

J.D. Mackinlay, S.K. Card, and G.G. Robertson, "Rapid Controlled Movement Through a Virtual 3D Workspace", Proceedings of SIGGRAPH '90, August 1990, 24(4):171-176.

Y. Noimark and D. Cohen-Or, "Streaming Scenes to MPEG-4 Video-enabled Devices", IEEE Computer Graphics and Applications, Jan/Feb 2003, 23(1):58-64.

J. Royan, C. Bouville, and P. Gioia, "PBTree - A New Progressive and Hierarchical Representation for Network-Based Navigation in Urban Environments", Proc. Vision Modeling Visualization (VMV 2003), pp. 299-307.

C. Russo dos Santos, P. Gros, P. Abel, D. Loisel, N. Trichaud, and J.-P. Paris, "Metaphor-Aware 3D Navigation", InfoVis'2000, IEEE Symposium on Information Visualization, Salt Lake City, October 2000, p. 155.

E.A. Wernert and A.J. Hanson, "A Framework for Assisted Exploration with Collaboration", Proc. IEEE Visualization, 1999, pp. 241-248.