Freehand interaction with large-scale 3D map data

Take down policy
If you believe that this document breaches copyright, please contact cris.tau@tuni.fi, and we will remove access to the work immediately and investigate your claim.

FREEHAND INTERACTION WITH LARGE-SCALE 3D MAP DATA

Vamsi Kiran Adhikarla[1,2], Paweł Woźniak[3], Attila Barsi[1], Dave Singhal[1], Péter Tamás Kovács[1,4]

[1]Holografika, Baross u. 3. H-1192 Budapest, [2]Pazmany Peter Catholic University, Faculty of Information Technology, Prater u. 50/a, Budapest, [3]t2i interaction lab, Chalmers University of Technology Gothenburg, Sweden, [4]Department of Signal Processing, Tampere University of Technology, Finland.

ABSTRACT

In this paper, we present our method and apparatus to visualize and interact with large scale 3D map data on a 3D light-field display in real time. 3D map data are streamed over Internet to the display in real-time based on request sent by the application. On the user side, data is processed and visualized on a large-scale 3D light field display. We present a method to efficiently interact with the visualized 3D map using freehand gestures. We use a Leap Motion sensor that supports sufficient refresh rate for our current application and is able to accurately track and acquire information on the user hand position. This information is further processed to support natural and fast interaction. We have also developed a method to automatically to smoothly adjust the map's plane to the screen plane of the display. The visualization and interaction method is scalable and allows the exploration of large-scale 3D maps down to the street level.

Index Terms — 3D-TV, 3D Interaction, HoloVizio, Human computer interaction, Leap Motion, Light field.

1. INTRODUCTION

Finding optimal methods of interacting with geographical data is an established research problem within Human-Computer Interaction (HCI).While map datasets are becoming more and more detailed and novel scanning methods enable creating extensive models of urban spaces, enabling effective ways of accessing geographical data is of utmost importance.

Maps have been extensively studied in the field of Computer-Supported Collaborative Work (CSCW) and recognized as an example of a boundary object – an artifact that enables communities to share information spaces. A large variety of systems for exploring maps were developed [1]. Maps come in different forms (e.g. paper, desktop, web, mobile) and all of these formats have different advantages. Digital and analog plans and maps play an important role in how the space around us is created, used and developed [2]. Maps help us understand and navigate the landscape around us and reflect on our activities and the environment around us [3].

A significant amount of past research that investigated providing effective map services to the public has been conducted. Bouvin et al. compared the affordances of different map types [4]. Nurminen [5] showed that infrastructure is in place to deliver a mobile 3D map experience over 3G networks.

Morrison et al. suggested using augmented reality to effectively interact with maps. MapLens [4] explores 3D maps with augmented reality lenses and shows benefits to user experience compared to a standard 2D map. Seager and Fraser [5] studied how users rotate maps to fit the surroundings and concluded that physical rotation is the most effective method. Here, we try to build on this affordance using a holographic displays system. Holographic displays can be efficient in manipulating objects as demonstrated by Hilliges et al. [6]. We try to combine the lessons learned in map interaction and interacting with holographic displays.

In this paper we introduce a system which deals with a highly complex, multi-resolution 3D model of city space. The system allows the users to effectively interact with the model while viewing it on a holographic screen. The interfaces can capitalize on the affordances of 3D light-field visualization and empower users to effectively explore a massive city model. Our ultimate goal is to establish design guidelines for systems that effectively harness the power of holographic displays to promote sense making and a deeper understanding of the world around us. The rest of the paper is organized as follows: in Section 2, we present few details on the map used for visualization. The main contributions of the work, light field visualization and interaction are presented in Sections 3 and 4 respectively. Results are presented in Section 5 and conclusions are drawn in Section 6.

2. LARGE-SCALE 3D MAP

The software that allows real-time streaming and rendering 3D map data on variety of 2D devices, as well as sample 3D map data have been developed and made available for research by myVR software [7]. The myVR mMap SDK is using a client-server model for downloading map data over the Internet. The C++ interface of API provides direct access to high level functions such as: querying the depth from the virtual camera at a particular pixel location, getting the position of virtual camera in latitude, longitude and height etc., and also allows real-time streaming and rendering of 3D model of a map. Inside the API, most of the communication is carried out using JSON (Java Script Object Notion) queries.

The mMap SDK uses composite for rendering the 3D map, with each composite consisting of several layers (e.g. we can have a layer that will render vector data, a layer that renders aerial data, and a layer that renders any Point Of Interests (POI's)). A typical map contains several composites and each composite and it's corresponding layers are enabled to receive JSON queries and return information back to the calling application. Every layer is assigned a priority when created and the order of rendering layers is based on the assigned priority. If two layers have the same priority, the layer that is first created gets the highest priority. The SDK is well optimized to eliminate unnecessary redrawing. We have built on a sample map viewer application which allows the exploration of potentially infinite map data, streams the multiresolution map data and displays it using Level Of Detail techniques. This application, which supported only mouse based interaction, have been amended with the described interaction techniques and the 3D display specific optimizations.

3. VISUALIZATION ON LIGHT-FIELD DISPLAYS

Real-time visualization on light-field displays requires rendering the given scene from many viewpoints that correspond to the characteristics of the specific light-field display [8,11]. One way to achieve this is using the HoloVizio OpenGL wrapper [12].

As the mMap SDK relies on OpenGL for rendering 3D maps, combining it with the OpenGL wrapping approach was the most straightforward approach to visualize it on a light-field 3D display.

The OpenGL wrapper library intercepts all OpenGL calls and sends rendering commands over the network to the cluster driving the light-field display as well as modify related data (such as textures, vertex arrays, VBOs, shaders etc.) on the fly to suit the specifications of the actual light-field display. The wrapper is designed in such a way that its operation is completely transparent to the client application producing the scene and it requires no modification of the client application.

In this case however, we had control over the application, thus we could exploit the additional OpenGL features provided by the wrapper library, through which we supply with additional semantic information related to the currently rendered scene and adjust visualization in 3D space accordingly. This additional information is the distance between the viewpoint and the ground of the 3D city model. What constitutes as ground level is specific to the model semantics and not deducible from the OpenGL command stream. When mapping the application's Region Of Interest (ROI) to the light-field display's ROI we map the ground level to be slightly behind the display's screen ensuring that buildings are in focus. The displacement by which we push the ground level behind the screen has been experimentally determined for different scale levels, as we have found that the same amount does not always work well. This mapping is recalculated for each frame as the viewpoint moves around so that even in the case of abrupt ground level changes (e.g. hilly terrain) buildings will stay focused on the screen plane.

4. MAP INTERACTION DESIGN

Designing interaction involves defining a set of interface controls to navigate through the map. As the data is visualized on a light field display with a smooth and continuous horizontal parallax, the spatial relation of the objects in the scene e.g., buildings in the map, are properly maintained similar to the real world (see Figure 1).

A natural way of interaction using free hand and without requiring the user to wear any additional gloves or trackers will be physically-realistic. An important concern while designing the interaction with a map is that the user should not get lost and should be able to find/navigate to places of interest or a familiar location easily. This requires a precise apparatus and method to control the positioning of virtual camera. Furthermore, as described before, the streaming and light field visualization is done on the fly without using any pre-rendered animations or images. Thus the interaction process should be fast enough to manipulate heavy light field data. Once the interaction control messages are acquired, rendering is performed in real-time on the light-field display's rendering cluster.

Microsoft Kinect can be used to track user hands. Kinect works fine starting from a given distance from the sensor and is mainly used to detect big gestures. Although it is possible to detect the hand gestures by precisely positioning the sensor and carefully considering the acquired information, tracking minute hand movements accurately is quite imprecise and error prone.
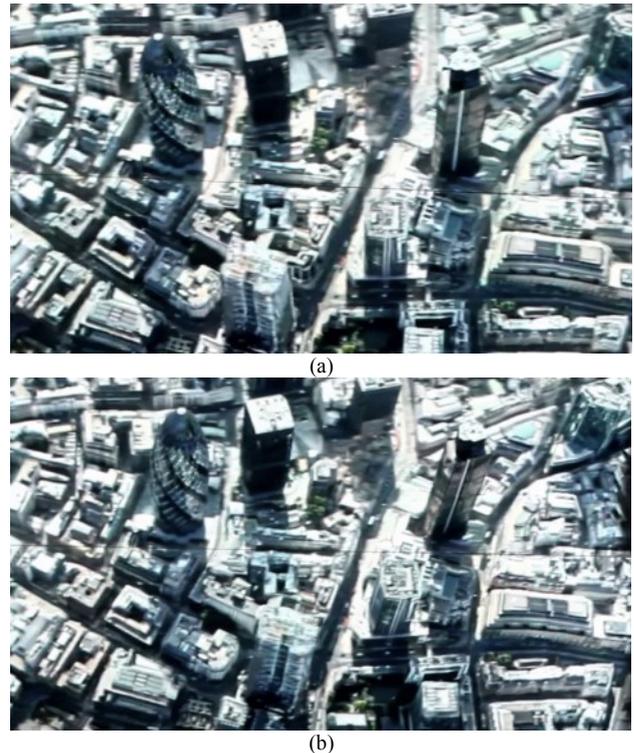

(a)

(b)

Figure 1. Sample 3D map on a light field display. (a) & (b) shows the identical light field as seen from different viewing positions

Leap Motion Controller [13] is a motion sensing device introduced by Leap Motion Inc. which fits very well with our interaction needs. The device is proven to be relatively inexpensive, precise and is useful in tracking the hands and finger inputs than any existing free hands interaction devices. The device has a high frame rate and comes with a USB interface. The device provides a virtual interaction space of about one sq. meter with almost 1/100th millimeter accuracy. The Leap Motion Control SDK provides access to abstract data such as number of hands and fingers sensed, their location, stabilized palm position etc. A direct access to few set of gestures such as circle, swipe, key tap and screen tap is also provided. Using the SDK, it is more convenient to design and program any user defined gestures.

Designing interaction gestures in a way that is obvious for untrained users can be a very complicated task. The main factor of concern is the complexity and familiarity of a gesture, as it directly effects the learning time. On one hand easily detectable gestures such as open hand, closed hand may not be very intuitive and require additional attention from the user. On the other hand more intuitive gestures used to interact with real world objects e.g., lifting/grabbing, could be more complex and often cannot be precisely defined within a given group of users. For us the main challenge is to bring the best trade-off between the complexity of the gesture and its intuitiveness i.e., the gestures should be very easy to learn and also should be precisely detected within a given amount of time to support real-time interaction.

4.1 Leap Motion mouse emulation

The 3D map application also supports mouse interaction with computer mouse. A straightforward idea is to use Leap Motion device to emulate mouse movements. Leap Motion follows right-handed Cartesian coordinate system (see Figure 2).
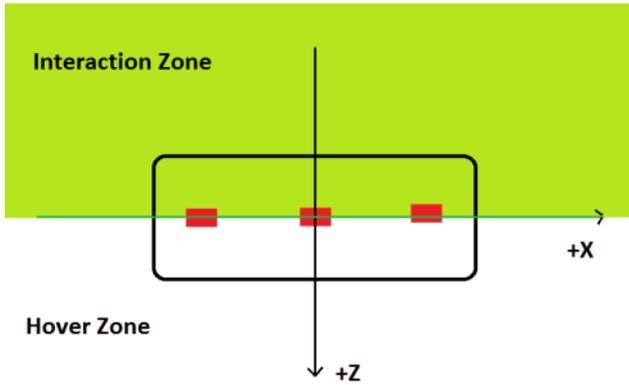
Figure 2. Leap Motion coordinate system (Top view)

A simple way to simulate mouse movements as pioneered by Leap Motion Inc. is to separate the sensed zone into two parts, and determine the navigation action based on which half of the interaction space we are moving. Moving the hand in the positive Z region is analogous to mouse hovering without clicking and moving the hand in negative Z region is analogous to click and drag. General map interaction such as panning and rotating can be isolated with this method, and separated into interaction and hover zones.

Although this approach works, our experiments suggested that user find it difficult to pay attention to their hand position, and also that this approach does not work without prior training. Watching which zone their hand is in distracts the user, and also limits the space available for interaction. Successful implementation of this approach requires initial training to the user to understand and locate the +Z and –Z regions of the device for interaction.

4.2 Gestures for interaction

A contributing factor to the degree of intuitiveness is the increasing usage of mobile smart phones. These devices usually contain maps and the touch interaction for navigating through the map is very well in practice now a days. Designing a set of similar gestures for interaction can make the learning phase shorter, or eliminate it completely, based on prior experience. Generally interaction includes pan, rotate and zoom. The respective gestures are defined in the following sub-sections. All the gestures are active within the valid Field Of View (FOV) of Leap Motion device.

4.2.1 Pan
Panning in horizontal and vertical directions can be done by translating the virtual camera in the opposite direction by a given amount. Panning is achieved using one hand (either left or right). Leap Motion driver runs a separate thread providing the information about the number of hands and the stabilized palm position continuously. As the display, scene and Leap Motion coordinate systems are different, first the axes are normalized and then the relative changes as seen by the Leap are used to update the scene camera. The Leap device faces upside, the direction vectors of the device and the light field display point towards the same direction.

Information on the previous from the Leap is always stored until the next frame. Up on receiving a new frame, the difference in the hand position is calculated and a panning message with respective parameters is dispatched to translate the virtual camera. Figure 3 (a) shows the panning gesture along X axis (top view) and identical gesture along the Y axis is also valid.
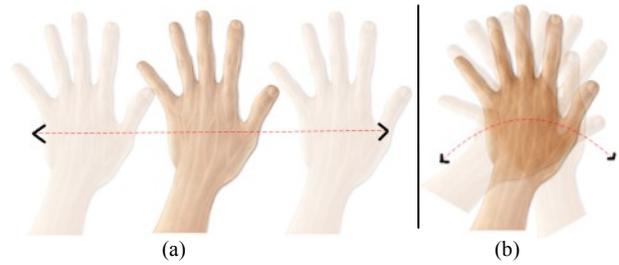

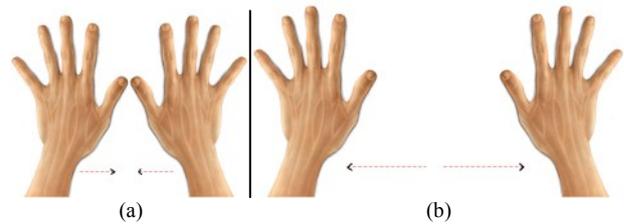Figure 3. One hand pan (a) and rotate (b) gestures

Figure 4. Two hands zoom out (a) and zoom in (b) gestures

4.2.2 Rotate
The rotation gesture is implemented using one hand. The virtual camera is made to mimic the hand rotation. As translation also depends on single hand gesturing, it is needed to isolate the rotate and pan actions carefully for a smooth interaction. A simple approach is to detect the change in the position of a hand in a given time. If the stabilized palm position does not change above a fixed threshold in a given amount of time, rotation event is recorded and the subsequent steps involve assessing the amount of hand rotation in degrees. Similar to translation we rely on the previous frame information to acquire the rotation amount and direction. Figure 3 (b) shows one hand rotation gesture.

4.2.3 Zoom
Zooming mode is activated if two hands are detected within the FOV of Leap Motion. Bringing two hands closer is the gesture for zooming out and taking the hands apart results in zooming in to the map as shown in Figure 4. As the entire application is real-time, it is important to preserve the states of various modes (for e.g., translation at a give zoom, zoom at a give rotation and translation). From the rendering side, the view matrix takes care of the virtual camera positioning and it is also important to check the state of a given mode from the interaction point of view as we only need to update the current state. This is done in a simple way by storing the active times of various modes and checking the time lapse.

Similar to panning and rotation, zooming commands are dispatched considering the current and previous frames from the Leap Motion. Our experiments showed that depending on one immediate before frame temporally provides sufficient and accurate information to detect the change in states and also meets real-time requirements.

5. RESULTS

We tested the presented method using a Leap Motion device and myVR mMap 2.0 SDK on Holografika's HV80C light field cinema display. HV80C incorporates front projection mechanism, driven by a cluster of 10 computers and 80 projection modules providing an effective FOV of 45 degrees [9]. The application runs on a demo computer that controls the rendering cluster. With a 24 Mbps download speed internet

connection, the map data is streamed and rendered at 75 frames per second (FPS). The cameras of Leap Motion generate almost 300 FPS of the raw data from which the information on the user hand(s) position is extracted. The frame rate supported by Leap Motion is much higher than the required frame rate for the application leaving us sufficient space to further processing and filtering the gestures before dispatching the final interaction commands. Kinect acquires depth and color stream at only 30FPS and hence limits the interaction speed.

Figure 5. Interaction with 3D map on HV80C light field display.

We observed that the lag from hand movement to the update of the visualized light field visualization is almost negligible. On the rendering side, a 10 PC cluster that normally drives this display is capable to render in real-time in response to the application's window updates. The whole system is implemented in C++ and runs in real-time (see [10] for video demonstration of our results).

6. CONCLUSIONS AND FUTURE WORK

We presented an approach to visualize and interact with large scale 3D maps on a light field display using Leap Motion device. The key aspects of the presented system are visualization, rendering and interaction. Visualizing the map on light field display brings out natural and realistic depth cues and the buildings appear to be sticking out of the display surface. The OpenGL wrapper allows real-time light field rendering to produce multiple view points after holographic transformation. Leap Motion best suits our purpose for interaction with the 3D map. A set of simple and more intuitive gestures are implemented for real-time interaction. The results show that the proposed system is robust and helps users to navigate through familiar locations on the map without getting lost.

The presented system is first of its kind exploring and combining the latest advances in 3D visualization and interaction techniques. In future, we are planning to conduct user studies for subjective evaluation of the effectiveness of the presented interaction method. We plan to perform initial evaluation of the system in a controlled user study very soon. We expect that the study will result in significant refinements to the gesture suite. We will then proceed with formal evaluation of the system. We will also aim to evaluate the system in an in situ study where users will be empowered to use the city map to explore their own urban space. We wonder how holographic display can impact communities and perform in public settings. We hope to investigate how communities can benefit from the system installed in a public setting and how will this compare to a traditional 2D screen and stereoscopic displays. We hope that our design will inspire further inquiries into how holographic displays can help us develop systems of social significance.

A further idea is to register the spatial 3D spaces of Leap Motion and HoloVizio by means of 3D calibration, similar to the calibration of touch screens. Such a system is highly desirable as can produces realistic effects such as interaction with a particular building in the current scene.

8. REFERENCES

[1] S. Bodker, "A Human Activity Approach to User Interfaces," *Human-Computer Interaction*, vol. 4. pp. 171–195, 1989.

[2] N. O. Bouvin, C. Brodersen, S. Bødker, A. Hansen, and C. N. Klokmose, "A comparative study of map use," in *CHI '06 extended abstracts on Human factors in computing systems - CHI EA '06*, 2006, p. 592.

[3] A. Nurminen, "Mobile, hardware-accelerated urban 3D maps in 3G networks," in *Proceedings of WEB3D 2007 - 12th international conference on 3D wed technology,* 2007, pp. 7–16.

[4] A. Morrison, A. Oulasvirta, P. Peltonen, S. Lemmela, G. Jacucci, G. Reitmayr, J. Näsänen, and A. Juustila, "Like bees around the hive: a comparative study of a mobile augmented reality map," in *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 2009, p. 1889.

[5] W. Seager and D. S. Fraser, "Comparing physical, automatic and manual map rotation for pedestrian navigation," in *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, 2007, p. 767.

[6] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson, "HoloDesk: Direct 3D Interactions with a Situated See-through Display," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 2421–2430.

[7] MyVR software, "mMap SDK," [Online]. Available: http://bit.ly/1hAucMa [Accessed: Apr. 14, 2014]

[8] T. Balogh: Method and apparatus for displaying three-dimensional images, U.S. Patent 6,201,565, EP 0900501, 1997.

[9] Tibor Balogh, Zsolt Nagy, Péter Tamás Kovács, Vamsi K. Adhikarla, "Natural 3D content on glasses-free light-field 3D cinema," *Proc. SPIE 8648, Stereoscopic Displays and Applications*, March 12, 2013.

[10] Holografika, "MyVR Software demonstration on HoloVizio C80 Glasses-free 3D cinema system," [Online]. Available: http://bit.ly/1jE2NZd [Accessed: Apr. 14, 2014]

[11] T. Balogh, "The HoloVizio system," in Proc. SPIE 6055, *Stereoscopic Displays and Virtual Reality Systems XIII*, 60550U (January 27, 2006); doi:10.1117/12.650907.

[12] T. Balogh, P. T. Kovács, Z. Megyesi, A. Barsi, "HoloVizio - True 3D Display System," in *Proc. of 3DTV Conference*, 7-9 May 2007, pp. 1-4.

[13] Leap Motion Motion Controller [Online]. Available: https://www.leapmotion.com/ [Accessed: Apr. 14, 2014]