Touch-Enabled Input Devices for Controlling Virtual Environments

Taylor Edmiston * **Adam Golden** * **Adam Meily** *
Thomas Wischgoll *

* *Computer Science and Engineering, Wright State University.*

Abstract: The benefits of using virtual environment display technology is the familiarity of the user with the modalities of that environment providing a very intuitive access to models or data sets represented by using this technology. Various different styles of input devices are typically used for such virtual environments, ranging from standard game-pads to high-end commercial devices like an A.R.T. flystick2. These devices work great for operations such as selection or navigating the scene. Whenever more sophisticated dialog-based input is required, these devices typically rely on traditional 2D metaphors projected into the virtual environment. The use of tablet devices can provide a significantly more natural input-paradigm under these circumstances. This paper describes the deployment of a standard Android tablet device that interfaces with a virtual environment over the wireless network. The tablet device was tested using traditional CAVE-type display configurations and wall-type display systems using various different 3D stereoscopic technology including active stereo and passive stereo.

1. INTRODUCTION

Virtual environment display technology can provide very intuitive access to models or data sets visualized by the display system through the use of motion models close to the real world thereby providing a familiar control metaphor to the user. In order to allow the user to freely move around in front of or within the display system wireless input devices are typically used coupled with a tracking system to allow the display environment to be aware of the location of the user and the input devices. This as well provides very intuitive input modalities, in which a user can grab parts of the scene and move them around just like in a real world environment. Due to the lack of traditional input devices, such as keyboard and mouse, however, traditional input and selection modalities tend to be more awkward to use. Many virtual environment programming environments, such as Oliver Kreylos' Vrui, provide tools for selecting items from a menu to change modalities. In order to do this, a menu pops up within the virtual environment. This menu is usually located directly in front of the user as determined via the tracking system. The user can then use a tracked input device to select an item from the menu. A wand-like metaphor is used in which a line extends out of the input device and the item with which this line intersects is selected. This input paradigm tends to be slow and awkward for the user to use which gets even worse when the user is required to make textual input as every single key would have to be selected in this way. In order to provide more suitable input devices for virtual environments, tablet devices can be used. These devices are relatively small in size and provide a very intuitive touch-based input paradigm in which the user can easily make selections or input text via an on-screen touch keyboard. For this project, an Android-based Toshiba 7-inch tablet was chosen with a sleeve that provides an elastic band for one of the user's hand to slide through. Hence, the device can be easily attached to the user's hand without limiting the user's movements in any way. The tablet then connects to the master node controlling the virtual environment via a secured wireless network connection. From that point on, it can exchange any data with the master node. For example, the master node could initiate a request for input. Instead of opening up a menu within the virtual environment as traditionally done, the menu would now pop up on the tablet from where the user can make a selection via the touch interface. Similarly, the master node could request a file to be selected. The tablet would then open a file dialog box in which the user can browse the directory hierarchy as found on the master node to identify the required file. As can be seen from these simple examples, a tablet-based interface provides a significantly more intuitive interface that is easier to use as a result of the touch-based input modalities. This scenario could of course be expanded in which textual input is required which is handled via the tablet just as easily. Similarly, the tracking system could be expanded such that it keeps track of the tablet's position and orientation. In that mode, it could completely replace any other input device. With its touch interface any arrangement and configuration of buttons can be supported on a tablet making it significantly more versatile than traditional input devices at the same time. Overall, the use of a tablet within a virtual environment can increase the ease of use as well as the versatility of the input modalities significantly over existing approaches.

2. RELATED WORK

Immersive virtual environments require two major hardware components. First, display technology is required that allows a user to view in 3D. Second, 3D suitable input devices are required that do not bind the user to a certain location but instead allow for maximal freedom of movement of the user. For the display, there are typically

a few technologies used. Head-mounted displays (HMDs) (Sutherland [1968], Fisher et al. [1986], Chung et al. [1989]) consist of two small screens mounted into a device that the user wears similar to a helmet such that the two screens are placed in front of the user's eyes. Since the device is equipped with two individual screens, different images for the left and right eye can be easily displayed resulting in a 3D effect experienced by the user. One advantage of head-mounted displays is that some can be used as see-through devices for augmented reality systems (Rolland and Fuchs [2000]). HMDs are now migrating into the gaming market in form of the Oculus Rift, a max-anticipated 110-degree field-of-view HMD that is now slowly getting available.

Other display types (Cliburn [2004], Pape and Anstey [2002], Czernuszenko et al. [1997]) rely on glasses that hide the left image from the right eye and vice versa. This allows for a majority of displays to be used. Often times large projection walls are used which can be configured as a large wall-type display or a CAVE-like environment. Two different types of glasses are used in combination with these displays: active and passive. With passive glasses, polarization is used to ensure that the left image can only be seen by the left eye and the other way around. For projection displays, two projectors are typically used where a polarization filter with different polarization is placed in front of each projector. The glasses then only let light pass through that has the appropriate polarization so that each eye only sees the image generated by one of the projectors. Nowadays, even some TFT-based monitors are becoming available that work with passive polarization glasses.

Active stereo glasses need to be synchronized with the display in such a way that ensures that the right image is only seen by the right eye and vice versa. Typically, the system displays the images for the left and right eye in an alternating fashion and activates and deactivates the glasses for the left and right eye in the active stereo glasses accordingly. The advantage of this type of glasses is that they work with many different display types, such as projection displays, CRT screens, or plasma displays. However, they do not work with many TFT screens since they, too, use polarization filters for displaying an image so that the active stereo glasses filter out the light entirely all the time.

Recently, auto-stereo displays were developed that are available at reasonable prices that can be used as displays for virtual environments. The advantage of this type of display is that it does not require the user to wear any glasses. Typically, barrier screens are used so that the light of half of the pixels gets directed more towards the left and the other half more towards the right. This way, one half of the pixels are only visible by one eye, whereas the other half can be seen only by the other eye, assuming the user is located somewhat centered in front of the display. As a result, this limits the mobility of the user to some extent.

As input devices, different wand or stylus devices are typically used. Often times, these are tracked either magnetically or optically to determine their position in 3D space without the need of any cabling. More recently, standard game devices are utilized in virtual environments as well which are wirelessly connected to the computer. Wischgoll et al. [2005] discuss the advantages of game controllers for navigation within virtual environments. Dang et al. [2007] studied the usability of various interaction interfaces, such as voice, wand, pen, and sketch interfaces. Klochek and MacKenzie [2006] introduced metrics for measuring the performance when using game controllers in three-dimensional environments. Wilson and Agrawala [2006] presented a technique for entering text using a standard game controller. MacKenzie [1995] discusses various input devices ranging from traditional mouse and keyboard to tracked devices and their properties, such as lag, with repsect to virtual environments. Barfield et al. [1998] tested joystick and SpaceBall as input device and report that the type of input device does not have an influence on the user's sense of presence within the virtual environment. Iphones and Ipads can also be used for navigating through a virtual environment as shown by Kim et al. [2009]. Pakanen et al. [2013] explored different GUI styles for virtual environments displayed on tablet devices. They reported that users preferred GUIs that create a secure and private feeling.

Based on the previously described technology, a visualization of a data set can be presented to a user. In order to navigate through or around a displayed model, the camera location needs to be modified. In general, a camera model describes point of view, orientation, aperture angle, and direction and ratio of motion. A general system for camera movement based on the specification of position and orientation of the camera is presented by Drucker et al. [1992], whereas Gleicher and Witkin [1992] choose an approach where through-the-lens control by solving for the time derivatives of the camera parameters is applied. The concept of walkthroughs in simulated virtual worlds using a flying metaphor has first been explored by Brooks Jr. [1986]. Other commonly applied metaphors for navigation in virtual environments (VEs) such as "eyeball in hand", "scene in hand" and "flying vehicle control" were introduced by Ware and Osborne [1990].

Researchers explored the suitability of immersive display technology for visualization purposes for quite some time. Unfortunately, the high price tag of most display setups results in only a few researchers having access to high-end immersive displays. An overview of the use of virtual reality technology for visualization can be found in the work by van Dam et al. [2000] and Brooks Jr. [1999].

The use of tiled displays has significantly increased lately thanks to prices for display devices coming down. Tiled displays can be built using projection-based displays, standard computer monitors, or large LCD panels. Projector-based tiled displays typically require calibration to make them appear seamless and uniform across the entire image. As shown by Brown et al. [2005] this calibration process can be automated. Thanks to recent advancements in graphics cards, namely ATI's Eyefinity and Nvidia's Surround technology, a single graphics card can drive up to six displays in case of ATI and three when using an Nvidia graphics card. This allows researchers to build tiled displays out of commodity off-the-shelf computers (Nirnimesh et al. [2007]). Thelen et al. [2009] demonstrated a tiled display wall composed out of 50 LCD panels that are driven by 25 computers used for large-scale volume visualization. Renambot et al. [2004] introduced a scalable

environment that can utilize tiled display configurations to provide a virtual high-resolution framebuffer to an application program.

3. DISPLAY SYSTEMS

Commercial systems developed specifically for virtual environments can be fairly costly with price tags around a million dollars. An example of such a system is shown in figure 1, where a visualization of a large-scale vascular structure consisting of 220 million individual vessel segments is visualized in a Barco Ispace. The four screens located on the floor and the three walls are projected on by Barco Galaxy projectors driven by five computers. There is one computer dedicated to each projection surface plus a master node. Each of these computers is equipped with 4 GB of memory, a dual Xeon 2.5 GHz configuration with an Nvidia Quadro FX 5800. In order to provide a fully immersive experience, an optical tracking system from A.R.T. is used for tracking the 3D coordinates of the user's head as well as the input device, an A.R.T. flystick2.

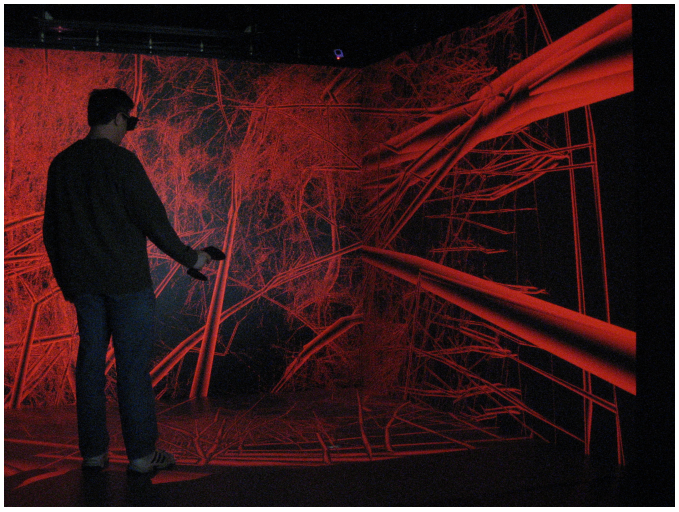

Fig. 1. User experiencing the visualization of a large-scale vascular structure in a Barco I-Space.

Such commercial setups are often cost-prohibitive for many visualization projects. Hence, more cost-effective solutions are needed. At the lowest end of the cost spectrum there are fishtank VR configurations. However, these setups only provide a relatively small amount of immersion due to the limited screen size.

For visualization tasks requiring a larger screen-space, a passive stereo, single-screen projection system can be used. The advantage of a passive system is that glasses are inexpensive and standard projectors can be utilized keeping the overall cost at a minimum. Even though there are 3D capable consumer-grade projectors at reasonable cost available, they typically support 3D at full HD resolution only at 24 Hz. This is mainly a limitation of the current HDMI 1.4 standard. This limitation does not apply to a passive setup since both projectors can run at the standard 60 Hz at full HD resolution, since they do not use 3D modes on the projector. Combined with passive polarization filters that are available starting at $35 this results in a very smooth, flicker-free display. The setup that was used for testing in this paper uses a rear-projection setup so that the user does not block any of the projected image. Since passive stereo is based on polarization, a special polarization-preserving projection screen is required, such as the Da-Lite 3D Virtual Black.

For tracking, a very cost-effective setup was chosen based on Microsoft's Kinect. The Kinect sensor provides full 6DoF data, albeit the directional information is not overly accurate. However, the positional data provided by the Kinect is fairly precise. A dedicated computer running FAAST Suma et al. [2011] captures the data from the Kinect sensor and transmits it to the rendering computer via the VRPN protocol. This provides head tracking and tracking of an input device, for example a Logitech F710 wireless game-pad. Since the directional information is not reliable enough, only positional tracking is used in combination with the input device. However, this still provides for a very intuitive user-experience with full immersion. Other promising devices exist, such as the Leap Motion which is not yet available or the Myo which is on pre-order.

The described passive stereo rear-projection screen configuration provides a comparably low-cost environment with equipment cost of less than $6,000. In order to keep the cost low, a custom-built projection stand was designed consisting of wooden horizontal levels connected vertically by threaded rods. Figure 2 shows the setup of the projectors. The two center levels, where the projectors are sitting on, are very adjustable as they rest on wing nuts that can slide up and down the threaded rods by simply turning them. This provides very fine-grained control over the projected image as all four corners can be adjusted individually so that the overlap between the left and right image produced by each of the projectors can be dialed in very effectively. The linear polarization filters are mounted at a distance of a few centimeters in front of the lens to prevent overheating and melting of the filters.

Figure 3 depicts a user in front of the passive projection screen. The passive stereo glasses with linear polarized filters that match the ones right in front of the projector lenses separate the left and right images. Hence, only the left image is seen with the left eye and the right image

Fig. 2. Set of projectors in a passive stereo configuration with polarization filters mounted in front of the lens.

is only seen by the right eye, unlike the double images that appear in the photograph since no polarization filter was used for taking the picture. The user is tracked by the Kinect sensor, which is located underneath the screen where it sits on top of the tracking computer. The tracked range is rather large ranging from around twelve feet down to just a couple feet in front of the screen, albeit the positional data gets somewhat distorted when getting too close to the screen.

Fig. 3. Tracked user in front of the passive projection screen with the tracking computer and the Kinect sensor on top underneath the screen.

Some visualization tasks may require even more immersion to further embed the user into the data visualized. To provide a good field of view at fairly high resolutions tiled display systems can be utilized. Large-screen LCD displays using LED backlighting are becoming available that have only little depth to them and some of these displays even come with very small bezels making them ideal for tiled display configurations. For example, Samsung's UA55E large-format display has only a small bezel of 3mm. To derive a tiled display configuration using these types of displays that is close to a traditional CAVE-type display, 27 of these displays can be mounted on an aluminum framing system. This results in three walls consisting of a 3×3 tiled configuration per wall (see figureẽffig:dive). The overall walkable footprint within the display system is 12×12 feet^2 with a display height of 87 inches. Hence, it provides a slightly larger area at almost the same height compared to a typical CAVE configuration. Since no rear-projection is required the overall footprint of the entire display with the framing system is only slightly larger (around 13×13 feet^2).

In order to keep the computer setup driving the display close to a traditional CAVE configuration, four computers are used. There is one master node that provides login capabilities and shows a console-type window of the virtual environment. Three slave computers display content on the large-format displays, one dedicated to a single wall each. Obviously, these computers now need to display parts of the virtual environment on 9 displays. For that, these computers are equipped with three ATI FirePro V7900 graphics card combined with an ATI S400 sync card to make sure that all displays show a single image at the exact same time. Three displays are then connected to a single card using ATI's Eyefinity. In the current configuration, there is a dedicated graphics card for each row of displays. This then allows the system to render at all of those three displays utilizing the left/right stereo mode provided by the HDMI 1.4 specification. Note that the full rendering performance of the graphics cards are retained as there is only one rendering step required for a row of three displays and there is a dedicated graphics card available for each of those rows of displays.

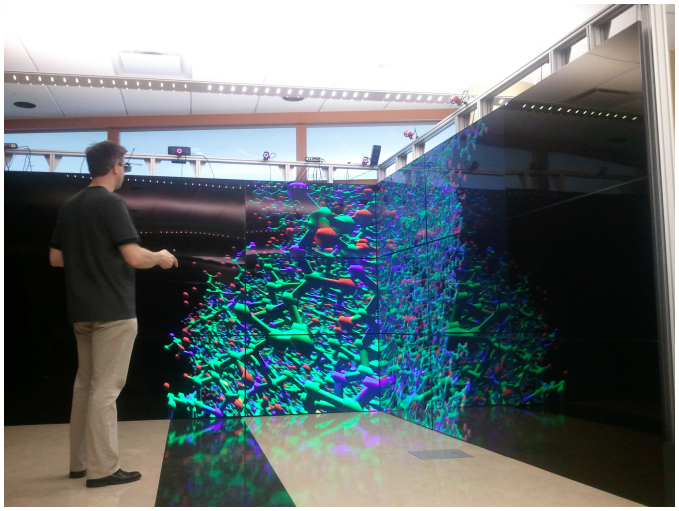

Fig. 4. Display infrastructure providing a fully immersive visualization of a molecular data set.

To provide reliable tracking of head position and input device, NaturalPoint's OptiTrack optical tracking system was chosen. since optical tracking requires line of sight between the cameras and the marker spheres, it is helpful if the marker spheres extend beyond the head to avoid occlusion. In addition, the number of cameras was increased to eleven. This ensures that a sufficient amount of marker spheres is visible at any given point in time with the entire space in between the displays being covered.

3.1 Visualization in Virtual Environments

In order to create a fully immersive virtual environment, more is needed than just interactive rendering. First, a stereo capable display system is needed which can deliver different images for the left and right eye. Second, a tracking system is required to identify the user's current position. Third, a software setup needs to tie all this together to render images mimicking the user's point of view in real time. There are several software packages that assist in creating virtual environments. Aside from commercial packages, such as VegaPrime or CAVElib, free software packages are available as well. Such free software packages are freeVR Sherman [2008] or VRjuggler Bierbaum et al. [2001]. The visualization described in this paper is based on the Vrui toolkit Kreylos [2008] developed by Oliver Kreylos. Compared to the other freely available software packages, Vrui offers more support for a variety of input devices as well as support for multi-threaded and multi-pipe rendering resulting in better rendering performance on more complex cluster-based display configurations. Vrui offers a great deal of flexibility. It can be adapted to

various different types of setups ranging from fishtank VR to full-scale CAVE-type displays. In fact, the same binary can be used and based on the hostname of the computer this binary based on Vrui identifies its settings from a configuration file to match the display system. Once the configuration is set up properly, the rendering algorithm needs to be integrated into the Vrui framework. This is essentially done by adding the rendering routine to the display function of a basic Vrui sample program provided as part of the Vrui distribution following a similar paradigm than most window-managing libraries.

Vrui runs only on Linux and Mac at this point. Most tracking software, however, is only available for Windows. To get around this, a dedicated tracking computer is usually used that interacts with the tracking device. In case of the Kinect sensor, the FAAST software developed at the University of Southern California Suma et al. [2011] is used. NaturalPoint's OptiTrack sensors come with their own proprietary software that is capable of transmitting the tracking data over the network.

All the display systems described in this paper except Barco's Ispace are configured in such a way that the tracking data gets transmitted via the VRPN protocol for both the OptiTrack and the Kinect sensors. The Ispace utilizes A.R.T.'s dtrack protocol in which the tracking computer actively sends the data to the master node. VRPN on the other hand is a passive protocol where the Vrui software requests the tracking data on a regular basis.

Various types of input devices are readily supported by Vrui. The Ispace uses A.R.T.'s flystick2, where the entire input data, such as positional information as well as joystick movements and button presses, are transmitted via the dtrack protocol. All other systems use a Logitech F710 wireless game-pad. If combined with the Kinect, only the positional information is used when tracking the game-pad. For the optical tracking, marker spheres are attached to the game-pad and the input is based on positional as well as directional information. For example, one of the joysticks is tied to a forward motion. If directional information is available, the forward direction is defined by the direction in which the game-pad is pointed. When using the Kinect for tracking, the forward direction is always fixed pointing horizontally into the screen to account for the fact that the directional information is not overly reliable. The use of a tablet device is not natively supported by Vrui. Hence, support for tablets as input devices was added on the client side of the Vrui application.

Displaying content using Vrui is rather straight forward as it follows a similar approach then most graphics packages. Any rendering code needs to be implemented in a display method that Vrui then regularly calls whenever a redraw is necessary. Since some display configurations utilize more than one computer, one needs to be a little careful about using information that is tied directly to a specific rendering process. For example, when using a texture such texture has to be uploaded into each graphics card individually and their identifiers may be different. However, Vrui provides a mechanism that is capable of handling such an environment.

So far, frameworks were developed for displaying on the systems described in this paper that are based on plain OpenGL, OpenSceneGraph, and VTK. OpenGL is directly supported in Vrui as it is based on OpenGL itself. Since OpenSceneGraph and VTK are also based on OpenGL, these can be integrated into Vrui as well. However, both of these graphics packages usually rely on handling the window management and user input themselves. Obviously, Vrui already takes care of both of those two items. Hence, a workaround is required that utilizes these graphics packages but makes them render into an existing OpenGL context. In case of OpenSceneGraph, this can be done relatively easily by creating a viewer instance in which the OpenGL settings defined by Vrui are recreated followed by a traversal of the scene graph. For this, the current OpenGL modelview and projection matrices are retrieved as well as the viewport and directly written into the OpenScenceGraph viewer's settings. For VTK, it is slightly more complicated as it requires the use of multipass rendering. While it does not actually require several render passes, it uses a vtkRenderPassCollection to force VTK to render into an existing OpenGL Drawable.

Based on these frameworks, one can benefit from most software packages that rely on OpenGL, OpenScenegraph, or VTK. For example, the Delta3D game engine based on OpenSceneGraph is readily supported by using a variant of the OpenSceneGraph framework. Figure 4 shows a rendering of a molecular structure based on VTK. Similarly, one can tie into the additional functionality provided by VTK for visualization.

Since different frameworks were developed based on VTK and OpenSceneGraph, a wide variety of applications can be supported by all the described display configurations. Obviously, virtual worlds can be created by importing realistic environments. When using OpenScreneGraph, virtual models can, for example, be imported from Google's 3D Warehouse to create realistic renditions similar to Google Earth. Such environments are frequently used by researchers from psychology to study people's behavior or use it for training purposes.

4. USER INPUT VIA TABLET DEVICE

The Vrui toolkit already provides mechanisms for input modalities via pop-up menus and even dialog boxes. Since in most sophisticated virtual environments the user is tracked, the system always knows where the user is and which way he or she is looking. This allows the Vrui toolkit to place the pop-up menus and dialog boxes in front of the user within the virtual environment. A laser pointer-like tool allows the user then to select items from the menu or within the dialog box similar to a mouse cursor. Hence, this input metaphor is a blend between 2D mouse-style input with flat widgets and a 3D input paradigm. As a result, this may not be as intuitive to the user and depending on where the menu or dialog box appears and what the size of it is parts of it may not fall within the display area. This can make this input mechanism somewhat difficult to use at times.

In order to remedy this blend of 2D and 3D mechanisms, truely 2D input devices can be introduced into the virtual environment. Tablets and mobile phones are a great success thanks in part to their touch-interfaces, making them very intuitive to use. The touch-sensitive display

surface in fact combines display and input device in the same area. This ensures that any menu or input dialog is shown in its entirety on the screen. With the touch-sensitive interface, selecting entries from a menu or making changes or selections within a dialog box are very intuitive.

Since the user wears the tablet on the hand at all times when interacting with the virtual environment, a smaller 7-inch tablet was chosen, namely a Toshiba AT1S5, with a sleeve that comes with a hand strap so that it can be attached directly to the user's hand. The Android software was developed using Eclipse 3.7.2 (Indigo) with the Android Development Tools (ADT) version 20.0.3. The minimal supported version of the Android operating system is 2.2 (Froyo). The tablet connects directly to a wireless network to communicate with the virtual environment using standard network sockets. Any communication is encoded using XML. This allows the virtual environment to enable certain dialog elements on the tablet and at the same time the tablet can send commands and input events directly back to the virtual environment.

As an example, the tablet allows the user to select a specific model into the virtual environment. These models are stored as files on the master node running the virtual environment. The master node sends the directory structure and files in the current directory to the tablet where they are displayed. The user can then make a selection among those. If a directory gets selected the tablet requests the list of files for that newly selected directory which are returned over the network. Once the user selects a file, the virtual environment opens the model represented by that file to show it within the virtual environment. In case more than one computer are used for rendering using a cluster configuration, the model file needs to be distributed from the master node to the display nodes. Vrui provides a multicast mechanism for that purpose in which the master node can simultaneously send the file name to the other nodes so that they can open and display the model file synchronously on all displays involved. Figure 5 shows the tablet running the software to serve as an input device for virtual environments. On its display, it lists all files and directories of the currently selected directory on the server running the virtual environment. Model files selected will then appear within the virtual environment similar to figure 6 running on the tiled display configuration for virtual environments.

Similarly, other input dialogs as well as menus can be displayed on the tablet screen from which the user can select. The position of the device can be incorporated into the selection process, for example to select points or objects, since some form of tracking system is typically used to achieve the immersive effect within the virtual environment.

The tablet was tested with various different 3D stereoscopic technology, including active stereo and passive stereo (both circular and linear polarization was tested). The content displayed on the tablet was still visible with any of the tested 3D glasses.

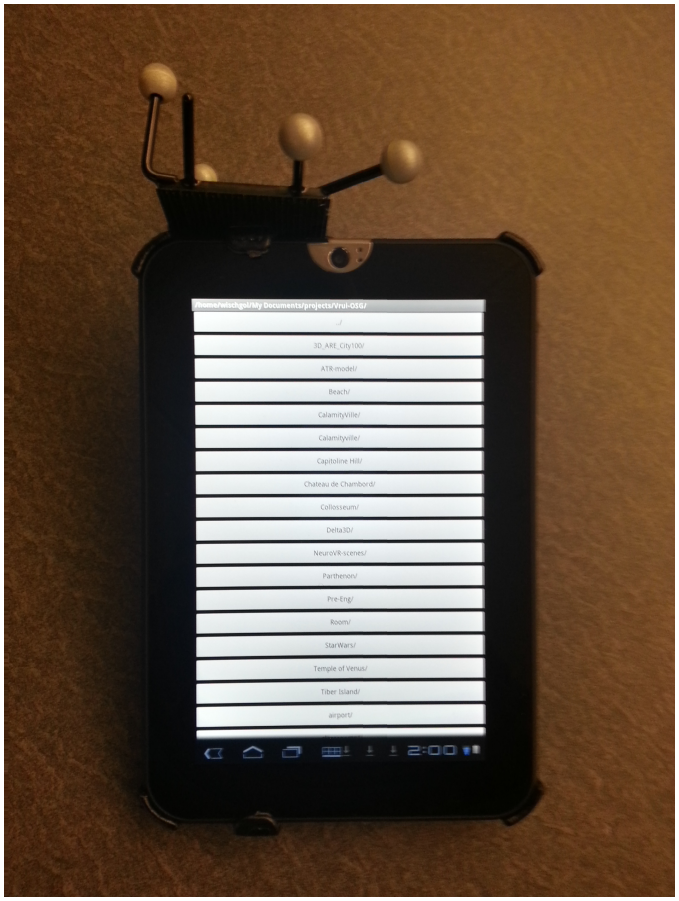

Fig. 5. Tablet device including attached marker spheres for tracking showing the current list of files to choose from.

5. CONCLUSION

This paper demonstrates the utility of using a standard tablet device for input when using virtual environments. Instead of integrating input dialogs within the virtual environment, the traditional 2D display of the tablet device seems considerably more suitable. The additional benefit of proving a touch-based input interface makes performing input commands and changing settings even easier. Overall, tablet devices can be a very useful addition to traditional 3D display technology suitable for virtual environments. Albeit a 7-inch tablet device was used for testing, the software runs on any Android device that runs version 2.2 (Froyo) or newer allowing it to run on, for example, Android phones as well.

6. ACKNOWLEDGMENTS

The authors would like to thank the department of Computer Science and Engineering and the College of Engineering and Computer Science for providing financial support for this project. The authors gratefully acknowledge the support of this project by the AFRL DoD Supercomputing Resource Center and the DoD HPCMP's User Productivity Enhancement, Technology Transfer, and Training (PETTT) Program (Contract No: GS04T09DBC0017 through High Performance Technologies, Inc.). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing

Fig. 6. Display infrastructure providing a fully immersive visualization virtual scenario.

the official policies or endorsements, either expressed or implied, of U.S. Air Force Research Laboratory or the U.S. Government.

REFERENCES

Woodrow Barfield, Kevin M. Baird, and Ove J. Bjorneseth. Presence in virtual environments as a function of type of input device and display update rate. *Displays*, 19(2):91–98, 1998.

A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira. Vr juggler: A virtual platform for virtual reality application development. In *Virtual Reality, 2001. Proceedings. IEEE*, pages 89–96. IEEE, 2001.

F. Brooks Jr. Walkthrough - A dynamic graphics system for simulating virtual buildings. *Proceedings SIGGRAPH Workshop on Interactive 3D Graphics*, pages 9–21, 1986.

F.P. Brooks Jr. What's real about virtual reality. *IEEE Computer Graphics and Applications*, 19(6):16–27, 1999.

M. Brown, A. Majumder, and R. Yang. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):193–206, 2005.

J. C. Chung, M. R. Harris, F. P. Brooks Jr., H. Fuchs, M. T. Kelley, J. W. Hughes, M. Ouh-Young, C. Cheung, R. L. Holloway, and M. Pique. Exploring virtual worlds with headmounted displays. In *Proceedings SPIE Conference, Non-holographic True Three-Dimensional Display Technologies*, pages 42–52, Jan 1989.

Daniel C. Cliburn. Virtual reality for small colleges. *J. Comput. Small Coll.*, 19(4):28–38, 2004. ISSN 1937-4771.

Marek Czernuszenko, Dave Pape, Daniel Sandin, Tom DeFanti, Gregory L. Dawe, and Maxine D. Brown. The immersadesk and infinity wall projection-based virtual reality displays. *SIGGRAPH Comput. Graph.*, 31(2): 46–49, 1997. ISSN 0097-8930.

Nguyen Thong Dang, Monica Tavanti, Ivan Rankin, and Matthew Cooper. A comparison of different input devices for a 3d environment. In *ECCE '07: Proceedings of the 14th European conference on Cognitive ergonomics*, pages 153–160, New York, NY, USA, 2007. ACM. ISBN 978-1-84799-849-1.

Steven M. Drucker, Tinsley A. Galyean, and David Zeltzer. Cinema: a system for procedural camera movements. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 67–70. ACM Press, 1992. ISBN 0-89791-467-8.

S. Fisher, M. McGreevy, J. Humphries, and W Robinett. Virtual environment display system. In *Workshop on Interactive 3D Graphics*, pages 77–87, 1986.

Michael Gleicher and Andrew Witkin. Through-the-lens camera control. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(2):331–340, July 1992.

J. Kim, D. Gracanin, K. Matkovic, and F. Quek. iphone/ipod touch as input devices for navigation in immersive virtual environments. In *Virtual Reality Conference, 2009. VR 2009. IEEE*, pages 261–262, 2009.

Chris Klochek and I. Scott MacKenzie. Performance measures of game controllers in a three-dimensional environment. In *GI '06: Proceedings of Graphics Interface 2006*, pages 73–79, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society. ISBN 1-56881-308-2.

Oliver Kreylos. Environment-Independent VR Development. *Lecture Notes in Computer Science*, 5358:901–912, 2008.

I. Scott MacKenzie. *Input Devices and Interaction Techniques for Advanced Computing*, pages 437–472. Oxford University Press, 1995.

Nirnimesh, P. Harish, and P. J. Narayanan. Garuda: A scalable tiled display wall using commodity pcs. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):864–877, 2007.

Minna Pakanen, Leena Arhippainen, and Seamus Hickey. Studying four 3d gui metaphors in virtual environment in tablet context. visual design and early phase user experience evaluation. In *ACHI 2013, The Sixth International Conference on Advances in Computer-Human Interactions*, pages 41–46, 2013.

Dave Pape and Josephine Anstey. Building an affordable projective, immersive display. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, pages 55–55, New York, NY, USA, 2002. ACM. ISBN 1-58113-525-4.

L. Renambot, A. Rao, R. Singh, B. Jeong, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, A. Spale, C. Zhang, et al. Sage: the scalable adaptive graphics environment. *Proceedings of WACE*, 9(23):2004–09, 2004.

Jannick P. Rolland and Henry Fuchs. Optical versus video see-through head-mounted displays in medical visualization. *Presence*, 9(3):287–309, 2000.

B. Sherman. Freevr: Virtual reality integration library, 2008.

E. Suma, B. Lange, A. Rizzo, D. Krum, and M. Bolas. FAAST: The flexible action and articulated skeleton toolkit. In *IEEE Virtual Reality*, pages 247–248, 2011.

I.E. Sutherland. A head-mounted three-dimensional display. In *Proc. the Fall Joint Computer Conference*, pages 757–764, 1968.

Sebastian Thelen, Joerg Meyer, Achim Ebert, and Hans Hagen. Giga-scale multiresolution volume rendering on distributed display clusters. In *HCIV'09 Proceedings of the Second IFIP WG 13.7 conference on Human-computer interaction and visualization*, pages 142–162, 2009.

Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, and Rosemary M. Simpson. Immersive vr for scientific visualization: A progress report. *IEEE Computer Graphics and Applications*, 20(6):26–52, 2000.

Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *Computer Graphics*, 24(2):175–183, March 1990. ISSN 0097-8930.

Andrew D. Wilson and Maneesh Agrawala. Text entry using a dual joystick game controller. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 475–478, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7.

Thomas Wischgoll, Elke Moritz, and Joerg Meyer. Navigational aspects of an interactive 3d exploration system for cardiovascular structures. In *IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2005)*, pages 721–726, 2005.