

# Display Systems for Visualization and Simulation in Virtual Environments

Thomas Wischgoll; Wright State University; Dayton, OH

## Abstract

Standard desktop setups, even with multiple monitor configurations, only provide a somewhat small view on the data set at hand. In addition, typical mouse and keyboard input paradigms often result in less user-friendly configurations, especially when it comes to dealing with 3D data sets. For simulation environments in which participants or users are supposed to be exposed to a more realistic scenario with increased immersion, desktop configurations, such as fishtank VR, are not necessarily a viable choice.

This paper aims at providing an overview of different display technology and input devices that provides a virtual environment paradigm suitable for various different visualization and simulation tasks. The focus is on cost-effective display technology that does not break a researchers budget. The software framework utilizing these displays combines different visualization and graphics packages to create an easy-to-use software environment that can run readily on all these displays without changing the software thereby providing easy-to-use software frameworks for visualization and simulation.

## Introduction

Common desktop configurations are often not very suitable for some visualization or simulation tasks. There are tasks that require higher resolution displays compared to what can be provided in a desktop setup or more immersion. Also, mouse and keyboard are not designed with 3D interaction in mind. As a result, workarounds have to be used to make these 2D input devices work within a 3D visualization or simulation environments. Using display configurations designed for virtual environments naturally provides a more intuitive input paradigm as these display setups are developed to mimic a more natural environment that the user is already familiar with. This is an advantage some visualization or simulation tasks can take advantage of as well.

Traditional virtual environments, such as original CAVE-type setups [11], use projection-based display configurations. These display types are typically rather expensive so that they are available to only very few visualization or simulation researchers. If higher resolutions are required for the visualization or simulation task tiled projector configurations can be used. However, these drive up the cost even further. On the other hand, similar display configurations can be achieved nowadays by using standard LCD panels. Since LCD panels are available that still provide 3D stereoscopic display capabilities, a tiled display configuration can still provide similar immersive effects than CAVE-type displays. Figure 1 shows an example of such a setup in which a user explores a molecular structure in a fully immersive environment. The advantage of this display configuration is that it is considerably less expensive than typical CAVE-type displays.

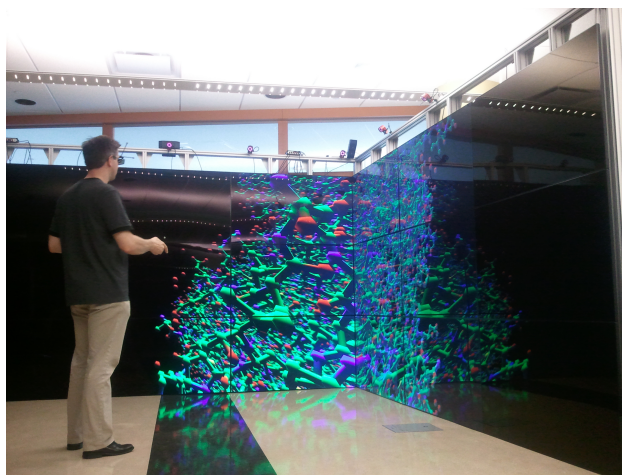


Figure 1. Display infrastructure providing a fully immersive visualization of a molecular data set.

The goal of this paper is to discuss low-cost display configurations suitable for immersive virtual environments that can be utilized for different visualization and simulation tasks. Basic open-source frameworks are deployed to provide a common interface to different visualization tools and libraries to make it as easy as possible for a visualization researcher to develop their algorithms as well as running immersive simulations. The structure of this paper is as follows. First, a summary of related work is provided followed by a description of low-cost display systems suitable for immersive visualization. Then, examples for visualizations taking advantage of these displays are listed. Finally, the results are discussed and some future work is listed.

## Related Work

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. Xia et al. [7] provide an overview of different systems for virtual environments as well as haptic input devices. The book by Earnshaw et al. [15] provides additional details about virtual reality systems. For the display, there are typically a few technologies used. Head-mounted displays (HMDs) [34, 17, 8] 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. The resolution of head-mounted

displays can be as low as only 800 by 600 pixels. Higher resolution head-mounted displays are available but significantly more expensive. For example, the Open Source Virtual Reality (OSVR) HMD has a resolution of 2160×1200 for their second generation development kit. One advantage of head-mounted displays is that some can be used as see-through devices for augmented reality systems [30]. Devices currently in development following that paradigm are Microsoft's HoloLens and Magic Leap.

Other display types [9, 26, 12] 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 required 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 and TV sets are becoming available that work with passive polarization glasses. Pastorelli et al. [27] describe a low-cost, projection-based display system that utilizes polarized glasses.

Active stereo glasses work similar to TFT screens. Two polarization filters are used with LCD crystals in between that can be activated to block all incoming light. The glasses then 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.

In order to avoid the need for glasses, auto-stereo displays [24] 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 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. [40] discuss the advantages of game controllers for navigation within virtual environments. Dang et al. [13] studied the usability of various interaction interfaces, such as voice, wand, pen, and sketch interfaces. Klochek et al. [19] introduced metrics for measuring the performance when using game controllers in three-dimensional environments. Wil-

son et al. [39] presented a technique for entering text using a standard game controller.

Based on the previously described technology, a visualization of a data set can be presented to a user. For virtual simulations, some of these devices can increase the amount of immersion significantly. 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 in [14], whereas Gleicher et al. [18] chose 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 [2]. 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 [37].

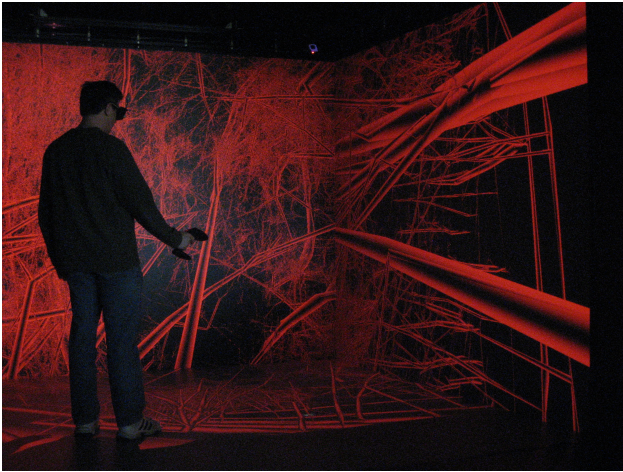
Researchers explored the suitability of immersive display technology for visualization purposes for quite some time. Unfortunately, the high price tag of most display setups results in only a few researchers having access to high-end immersive displays. Fortunately, with the advent of VR-type devices geared for the consumer market (mostly HMDs) prices are coming down. The possibility of using more cost-effective solutions for fully immersive display technology makes it more attractive to apply these to the area of visualization and simulation. An overview of the use of virtual reality technology for visualization can be found in the work by van Dam et al. [36] and Brooks [3]. In his book, Chaomei Chen [6] discusses the utilization of virtual environments for information visualization and provides effective examples. The availability of specifically head-mounted displays at relatively low cost is pushing this technology more and more into the consumer's hands, mostly for gaming. McMahan et al. [23] compare display fidelity and interaction fidelity for that type of application.

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. [4] 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. This allows researchers to build tiled displays out of commodity off-the-shelf computers [25]. Thelen et al. [35] 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. [29] introduced a scalable environment that can utilize tiled display configurations to provide a virtual high-resolution framebuffer to an application program.

## Display Systems

Commercial systems developed specifically for virtual environments can be fairly costly with price tags of up to a million dollars. An example of such a system is shown in figure 2, 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.



**Figure 2.** User experiencing the visualization of a large-scale vascular structure in a Barco Ispace.

Such commercial setups are often cost-prohibitive for many visualization and simulation projects. Hence, more cost-effective solutions are needed. At the lowest end of the cost spectrum are fishtank VR configurations. However, these setups only provide a relatively small amount of immersion due to the limited screen size. Larger display sizes can be achieved using 3D-capable TVs. Since the HDMI 1.4 specification provides 3D-capable modes that are typically used by TV manufacturers, the computer driving the TV merely needs to be configured to recreate the input for such 3D modes. For example, figure 3 shows a setup using a 65-inch Sony 4K TV combined with a NaturalPoint V120:Duo optical tracking system. The TV uses a passive stereo mode in which the left and right images are alternated line by line. While this reduces the overall number of pixels that can be displayed per eye to  $3840 \times 1080$ , it preserves the full 60Hz frame rate per eye. In addition, the passive stereo glasses allow for a flicker-free 3D experience at a very low cost per pair of glasses. The overall cost for such a configuring is around \$5,300, which includes the two computers driving the display and tracking system.

For visualization or simulation tasks requiring an even 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 HDMI 1.4 standard. Support of higher refreshrates via HDMI 2.0 requires high-end graphics hardware and projectors, such as Nvidia's 9th series GTX cards. This limi-

tation 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 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 FFAST [33] 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, a Logitech F710 wireless gamepad. 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.

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 4 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.** Rendering of a vascular structure using a 3D TV combined with optical tracking.



Figure 5 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 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.

In order to achieve a higher resolution to provide the user with more detailed information, tiled setups are common. In order to minimize the issue that arises from the bezels interrupting the images, 4K screens that are becoming quite affordable seem like a good choice. A 2x3 configuration of Sony's 50-inch 4K TVs is shown in figure 6. These TVs already come with a built-in polarization filter to support stereo images so combined with a tracking system this configuration provides similar capabilities as the previously described projection screen, albeit at a significantly higher resolution. As input devices, a gamepad can be used. In addition, Logitech's wireless rechargeable touchpad T650 can serve as a very intuitive input device as well. This touchpad supports multi-touch with up to four fingers being tracked simultaneously. With this capability, smartphone-type interaction can be realized, including single touch rotation and two-finger pinch zoom. Since most people are already very familiar with the input paradigm of their smartphone this makes for a very intuitive input device.

Some visualization tasks may require even more immersion to further embed the user into the data visualized or the simulation. There are different ways to create such an environment. One way is to use head-mounted displays (HMDs). In order to provide a cost-effective solution, consumer-grade head-mounted displays are entering the market that are geared toward watching 3D bluray movies. However, these types of displays still provide full 3D capabilities. For example, the Sony HMZ-T1 supports the 720p top-bottom HDMI mode at 60 frames per second therefore providing



**Figure 4.** Set of projectors in a passive stereo configuration with polarization filters mounted in front of the lens.



**Figure 5.** Tracked user in front of the passive projection screen with the tracking computer and the Kinect sensor on top underneath the screen.



**Figure 6.** A user in front of the 4K tiled display wall.

full 720p resolution per eye at 60Hz. When using a head-mounted display the tracking system has to provide directional information in addition to the position, i.e. full 6 degree-of-freedom (DoF) data. However, the Kinect sensor is not capable of reliably producing directional information as it derives the direction from the skeleton by considering the previous attachment point within the skeletal structure. This is not precise enough for use with a head-mounted display. Instead a more reliably tracking system, such as the NaturalPoint OptiTrack could be used. This requires the use of reflective markers which have to be mounted on the head-mounted display. For better tracking results, these markers should extend beyond the user's head to ensure optimal tracking results while the user moves around. This is why the standard marker configuration that comes with the OptiTrack system shown in figure 7 was replaced with a custom-made configuration that sticks out further shown in figure 8, which depicts a user wearing the head-mounted display. The marker spheres mounted to the top front of the display are tracked by the OptiTrack system using the red cameras in the top left corner of the image mounted to the aluminum framing. Similar to the head-mounted display, reflective markers are attached to the input device (figure 9 to provide 3D positional and directional input capabilities for full 6 degree-of-freedom input.

There are other head-mounted displays available nowadays



that even come with their own tracking systems, such as the Oculus Rift and HTC Vive, for an even lower cost alternative. While the Oculus Rift is supported in Linux, Linux support for the HTC Vive is still not fully available. However, first tests with the HTC Vive in our lab seemed very promising.



**Figure 7.** The Sony HMZ-T1 head-mounted display with reflective markers attached for tracking.



**Figure 8.** User wearing a head-mounted display. Head position and input device are tracked via the optical tracking system mounted on the aluminum framing (top left).

Obviously, head-mounted displays provide a great advantage in that the user can look in any direction and still be presented with

the proper view onto the virtual environment. However, the resolution is somewhat limited with most devices. 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 3x3 tiled configuration per wall. The overall walkable footprint within the display system is 12x12 feet<sup>2</sup> 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 13x13 feet<sup>2</sup>).

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 which is required for active stereo. 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 to 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.

To provide reliable tracking of head position and input devices, 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



**Figure 9.** Logitech gamepad with reflective markers attached for tracking.

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.

### **Visualization and Simulation 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 [32] or VRjuggler [1]. Based on these packages further software development is needed to handle the management of the actual VR content. One example for this is the work by Pavlik et al. [28] who provided a framework utilizing OpenSceneGraph [5] using VR Juggler.

The visualization and simulation framework described in this paper is based on the Vrui toolkit [21] 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 FFAST software developed at the University of Southern California [33] 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 gamepad. If combined with the Kinect, only the positional information is used when track-

ing the gamepad. For the optical tracking, marker spheres are attached to the gamepad 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 gamepad 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. Some systems also have touch-capabilities either via a touchscreen or Logitech's T650 to provide touch-based rotation and pinch-zoom capabilities.

Displaying content using Vrui is rather straight forward as it follows a similar approach than 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 OpenSceneGraph viewer's settings. For older versions of 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. Later versions of VTK (version 6.2 and higher) actually support rendering into an existing OpenGL context making this step significantly easier.

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 1 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 OpenSceneGraph, 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.

So far, several projects in our group were ported to take advantage of the immersive display technology. Vascular renderings can help investigate the geometric configuration of that structure. This can be very useful for vascular structures that were modeled based on statistical assumptions to ensure that the model generated is accurate and that there are no flaws in the algorithms that created the model. An example for rendering such large-scale models is shown in figure 2 where special rendering techniques are used that deploy a geometry shader to compute billboard representations of the individual vessel segments followed by a shader program to make the billboards look like tapered cylinders. Similarly, vascular structures extracted from CT scans can be displayed to search for geometric configurations indicating a diseased state or high risk areas, such as the one shown in figures 5. From the experience so far, such immersive renderings can make it easier for the user to navigate the structure as navigation is more natural in that it tries to mimic the real-world navigational model in which the user moves around and the perspective from the current point of view is recreated by the display.



**Figure 10.** User exploring a volume rendering of a CT angiogram data set in an immersive environment.

Similar to rendering vascular structures, volumetric data, such as the 3D CT scans the vascular models were extracted from, can be directly displayed. An example is shown in figure 10 showing a volume rendering of a CT angiogram of a porcine torso. The volume rendering was done using the Vrui-based volume renderer Toirt Samhlaigh developed by Patrick O’Leary. By taking advantage of the navigational capabilities of the immersive environment, it becomes very easy for the user to quickly identify the appropriate perspective to identify critical sections of the volumetric data set.

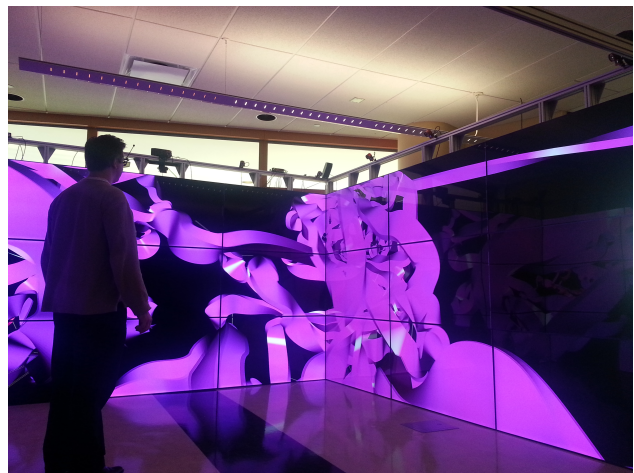
Another example where immersive visualization can help investigate complex structures is the visualization of molecular structures. Figure 1 shows an example that uses VTK to derive a rendering of the molecular structure. There is a multitude of software available already that is suitable for visualizing molecular structures. Figure 11 shows a user navigating a molecular structure within the DIVE environment based on a Vrui version of Oliver Kreylos’ Protoshop software [20]. While it is capable

of running in all the described environments, the DIVE configuration particularly shows its strengths in that it is very capable of reproducing the high detail and the fine-grained structures of the molecular visualization thanks to its high resolution.

In applications such as flow visualization, immersive virtual environments can be very helpful in defining the initial condition for streamlines or streamsurfaces. Since the input devices are typically tracked, as is the case in all the configurations described in this paper, identifying 3D locations takes just a click of a button with the input controller positioned at the desired location. Sweeping motions can similarly be used for initiating streamsurfaces. As is the case for the other examples, such immersive setups allow for more intuitive use in terms of view positions and navigating the data. We are currently in the process of porting the FAnToM software package [38] to support the Vrui environment and therefore all display systems described in this paper. This will provide a vast variety of visualization algorithms applicable to flow visualizations. Particularly the DIVE display setup can be especially useful as a result of its large display surfaces that can depict the data at higher resolution compared to many other available systems.

There is a myriad of other visualization applications that could benefit from immersive display technology. Some Information Visualization applications may be able to benefit from the higher resolutions provided by the DIVE display setup as well as the full immersion. By reducing the overall price point, the entrance fee for researchers that want to apply this intuitive technology will be considerably reduced thereby increasing the proliferation of fully immersive display systems.

Our simulation framework builds on Open Source software packages, including *SmartBody* [31] and *OpenSceneGraph* (OSG). Figure 12 outlines the architecture with all of its major components. This provides a very flexible tool that can explicitly incorporate geometric models to form a complex, computationally accessible representation of the experienced environment. Our suite of generic models make the process of generating a scenario more efficient. The software framework is designed to ingest a configuration file that describes the entire scenario in such a way that it specifies which models to incorporate into the



**Figure 11.** User exploring a molecular rendering based on VRProtoshop.



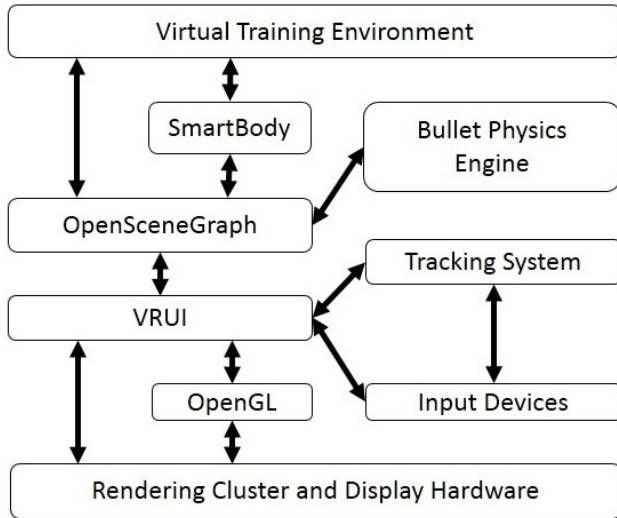


Figure 12. Architecture of the simulation environment.

scenario. In addition, geometric changes can be made to these models, such as scaling, rotating, and translating, so that they can be adjusted for size differences and freely arranged to form the scenario. This enables us to easily incorporate available models, models that we generated ourselves, or thirdparty generated models. Additional parameters can specify whether the object to be generated is supposed to be supported as a physical object the user can move around and whether physics need to be applied. In the latter case, the physics engine *bullet* [10] is applied for a fully capable physics model, such that dropped objects fall to the ground or one can knock over other objects that then behave physically correctly.

Characters can be included in the configuration file such that a model file representing that character combined with a motion model for that character are specified. The Open Source toolkit SmartBody then can generate a high fidelity, geometric representation of that character. This toolkit also provides the ability to animate the character actions fully. A collision engine assures that the characters avoid objects and each other to obtain a more realistic representation overall. The characters can perform gestures as well as include facial animation. Overall, this provides for a great framework for generating such virtual scenarios which automatically supports all the display systems described in this article.

### Application Examples

We intend to utilize the capabilities of our virtual display systems for various different application areas in visualization and simulation. Some visualization examples are listed in the previous sections already. In addition, we successfully used the DIVE system for a comparative visualization of molecular models in which the user was able to look at 27 models for direct comparison with fully tracked interaction based on view direction and input devices [22].

On the simulation side, these display systems can be used for training of personnel. For example, such systems could be deployed for medical training such that people are trained certain

procedures or behaviors. Preliminary results were reported that there is an increase in retention using virtual scenarios compared to traditional approaches [16].

Similarly, a virtual shopping environment can be created in which a study participant is asked to indicate when he or she takes notice of specific items within the environment while moving automatically or autonomously through the environment. Figure 13 shows an example of such a simple virtual environment. The resulting data is important for an ongoing, NSF-funded research project that studies product placement and legibility issues. As such, display systems like the Barco Ispace and the DIVE system are more appropriate as in this study peripheral vision is an important factor. These systems can provide a field of view of over 180 degrees, whereas head-mounted displays can only deliver up to 140 degrees. As for studying legibility issues, higher resolution displays are likely more appropriate, which is why we will run our experiments using the DIVE system.

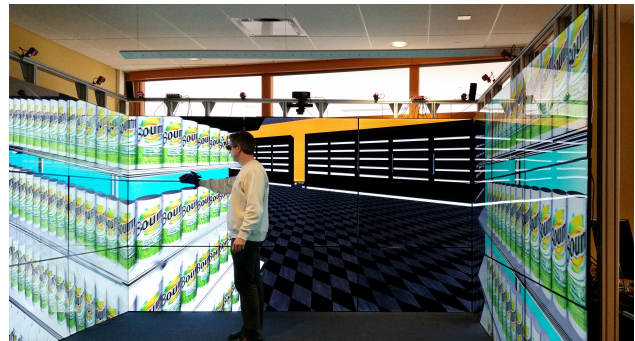


Figure 13. Virtual shopping environment.

### Discussion and Future Works

Obviously, the different display systems described in this paper have their advantages and disadvantages when used for visualization and simulation. The table in figure 14 lists their capabilities and overall price points. Except for Barco's Ispace, all systems were specifically designed with achieving a configuration that provides the best utilitarian value. This is particularly important for many research projects as the use of fully immersive display technology for the purpose of visualization is often not pursued due to the high price tag of commercial systems.

Display configurations based on 3D TVs are reasonably priced. Using flat panel TVs results in fairly crisp picture quality. Note that with most 3D capable TVs the resolution gets cut in half. This is the case for the Sony TV described earlier but also for Samsung TVs utilizing the checkerboard pattern for 3D. The price point can be further reduced when using the Kinect sensor for tracking.

The projection-based display is capable of driving larger screen sizes. On the downside of that, the image is not as bright and crisp when compared to the TV setups. Since it is using passive stereo, polarization filters have to be used which further reduce the brightness of the image. On the plus side, the passive configuration uses two projectors thereby doubling the overall light output of the system. In addition, the passive stereo configuration allows for flicker-free stereoscopic vision. The track-

Display system	Size	Resolution (per eye)	Tracking system	Input capabilities	Price (including computers)
3D TV	49 inch diagonal	4 Megapixel	Kinect	position tracked, 2 joysticks 14 buttons	\$3,200
3D TV	65 inch diagonal	4 Megapixel	optical (2 cameras)	fully tracked, 2 joysticks 14 buttons	\$5,300
Passive screen	102 inch diagonal	2 Megapixels	Kinect	position tracked, 2 joysticks, 14 buttons	\$6,000
4K tiled screen	150 inch diagonal	24 Megapixels	Kinect	position tracked, 2 joysticks, 14 buttons	\$10,000
HMD	N/A	0.9 Megapixel	optical (3 cameras)	fully tracked, 2 joysticks, 14 buttons	\$5,000
Oculus Rift	N/A	1.3 Megapixel	optical/accelerometer	fully tracked, 2 joysticks, 14 buttons	\$600
D.I.V.E.	144×144×87inch <sup>3</sup>	27 Megapixels	optical (11 cameras)	fully tracked, 2 joysticks, 14 buttons	\$120,000
Barco Ispace	120×120×90inch <sup>3</sup>	4.4 Megapixels	optical (4 cameras)	fully tracked, 1 joystick, 5 buttons	\$750,000

Figure 14. This table lists the capabilities of the display setups and their tracking systems.

ing based on the Kinect is fairly reliable and does not require any additional modifications, such as adding marker spheres. It also handles changes in users very well: as soon as the current user steps out of the visible area, the Kinect picks up the next person which is then used for tracking from that point on.

The advantage of the head-mounted display is clearly the fact that it provides full 360 degree display capabilities: no matter which way the user looks or moves the system is capable of recreating the proper point of view assuming that the tracking system covers a sufficiently large area. The downside is the limited resolution and narrower field of view with most head-mounted displays, especially if the goal is to find a reasonably priced solution. Also, only one user is able to see what is displayed which may be undesirable in some applications. In addition, the nature of head-mounted displays completely replaces what the user can see. As a result, no real-world objects can be introduced unless these objects are fully tracked themselves and virtual representatives are integrated into the visualization.

The Display Infrastructure for Virtual Environments (DIVE) was designed to resemble a typical CAVE configuration. Since the specification of any flat panel TV does not allow for horizontal installation, no floor was realized, though. As a result, the display configuration provides a 270 degree surrounding view for full immersion. Since the bezels are very small, they are not intrusive at all and do not seem to reduce the visual quality. Compared to the Ispace, the resolution is considerably higher and the large-format displays provide significantly more brightness so that the overall image quality is significantly improved over conventional projection-based systems. In order to further compare Ispace and DIVE, a small user study was performed in which ten users (two female and eight male age 18 through 35) were asked to find different objects in different virtual environments followed by a short questionnaire. Overall, the participants were slightly faster in finding the objects in the DIVE setup. This may be attributed to the increased resolution provided by that configuration. Also, two participants noticed issues with nausea when using the sys-

tems but stated that the effect was less in the DIVE compared to the Ispace. Further testing has to be done; but it seems from our experience that the DIVE setup is a worthy replacement of traditional CAVE configurations, especially considering the significant reduction in price.

In our experience, virtual environments often provide easier and more intuitive access to the visualization techniques and simulations for the untrained user. Especially, the tracked input devices provide easier ways of selecting positions or areas in 3D, picking up an object, or navigating through 3D space. In accordance with the pros and cons of the described display systems, the utilitarian value depends on the visualization or simulation task at hand. Those tasks that require detailed and fine-grained visualizations or a high-detailed immersive environment can benefit from the increased resolution provided by, for example, the DIVE configuration. People who have used the DIVE setup so far were quite impressed with the high resolution provided by the displays as well as with the improved brightness, which is considerably higher compared to, for example, Barco's Ispace.

In the future, more thorough user studies will be needed to fully judge on the benefits of the display technologies for different visualization and simulation tasks. The author will setup a web site to provide access to more detailed descriptions of the display configurations as well as the software frameworks described in this paper.

### Acknowledgment

The author would like to thank the College of Engineering and Computer Science and the Department of Computer Science and Engineering at Wright State University for providing the funding for the display technology.

### References

- [1] 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.
- [2] F. Brooks Jr. Walkthrough - A dynamic graphics system for simulating virtual buildings. *Proceedings SIGGRAPH Workshop on Interactive 3D Graphics*, pages 9–21, 1986.
  - [3] F.P. Brooks Jr. What’s real about virtual reality. *IEEE Computer Graphics and Applications*, 19(6):16–27, 1999.
  - [4] 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.
  - [5] Don Burns and Robert Osfield. Open scene graph. In *Proceedings of the IEEE Virtual Reality*, volume 265, 2004.
  - [6] Chaomei Chen. *Information visualisation and virtual environments*. Springer Science & Business Media, 2013.
  - [7] Yufang Cheng, Cheng-Li Huang, and Chung-Sung Yang. Using a 3d immersive virtual environment system to enhance social understanding and social skills for children with autism spectrum disorders. *Focus on Autism and Other Developmental Disabilities*, 30(4):222–236, 2015.
  - [8] 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 head-mounted displays. In *Proceedings SPIE Conference, Non-holographic True Three-Dimensional Display Technologies*, pages 42–52, Jan 1989.
  - [9] Daniel C. Cliburn. Virtual reality for small colleges. *J. Comput. Small Coll.*, 19(4):28–38, 2004.
  - [10] Erwin Coumans. Bullet physics library, 2012.
  - [11] Carolina Cruz-Neira, Daniel J Sandin, and Thomas A DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993.
  - [12] 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.
  - [13] 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.
  - [14] 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.
  - [15] Rae A Earnshaw. *Virtual reality systems*. Academic press, 2014.
  - [16] S. Farra, E. Miller, N. Timm, and J. Schafer. Improved training for disasters using 3-d virtual reality simulation. *Western Journal of Nursing Research*, 35:655–671, 2013.
  - [17] S. Fisher, M. McGreevy, J. Humphries, and W Robinett. Virtual environment display system. In *Workshop on Interactive 3D Graphics*, pages 77–87, 1986.
  - [18] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. *Computer Graphics (SIGGRAPH ’92 Proceedings)*, 26(2):331–340, July 1992.
  - [19] 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.
  - [20] O. Kreylos, N. Max, and S. Crivelli. Protoshop: Interactive design of protein structures. In *Proceedings of CASP5-Fifth Meeting on the Critical Assessment of Techniques for Protein Structure Prediction*, pages A213–A214, Pacific Grove, California, 2002.
  - [21] Oliver Kreylos. Environment-Independent VR Development. *Lecture Notes in Computer Science*, 5358:901–912, 2008.
  - [22] Matthew Marangoni and Thomas Wischgoll. Comparative visualization of protein conformations using large high resolution displays with gestures and body tracking. In *Visualization and Data Analysis*, pages 939712–1–939712–12, 2015.
  - [23] Ryan P McMahan, Doug A Bowman, David J Zielinski, and Rachael B Brady. Evaluating display fidelity and interaction fidelity in a virtual reality game. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):626–633, 2012.
  - [24] A Neil. Autostereoscopic 3d displays. *Computer*, 8:32–36, 2005.
  - [25] Nirmimesh, 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.
  - [26] 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.
  - [27] Emiliano Pastorelli and Heiko Herrmann. A small-scale, low-budget semi-immersive virtual environment for scientific visualization and research. *Procedia Computer Science*, 25:14–22, 2013.
  - [28] Ryan A Pavlik and Judy M Vance. Vr jugglua: A framework for vr applications combining lua, openscenegraph, and vr juggler. In *2012 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, pages 29–35. IEEE, 2012.
  - [29] 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.
  - [30] Jannick P. Rolland and Henry Fuchs. Optical versus video see-through head-mounted displays in medical visualization. *Presence*, 9(3):287–309, 2000.
  - [31] Ari Shapiro. Building a character animation system. In Jan Allbeck and Petros Faloutsos, editors, *Motion in Games*, volume 7060 of *Lecture Notes in Computer Science*, pages 98–109. Springer Berlin / Heidelberg, 2011.
  - [32] B. Sherman. Freevr: Virtual reality integration library, 2008.
  - [33] E. Suma, B. Lange, A. Rizzo, D. Krum, and M. Bolas. FFAST: The flexible action and articulated skeleton toolkit. In *IEEE Virtual Reality*, pages 247–248, 2011.
  - [34] I.E. Sutherland. A head-mounted three-dimensional display. In *Proc. the Fall Joint Computer Conference*, pages 757–764, 1968.
  - [35] Sebastian Thelen, Joerg Meyer, Achim Ebert, and Hans Hagen. Giga-scale multiresolution volume rendering on dis-



- tributed display clusters. In *HCIV'09 Proceedings of the Second IFIP WG 13.7 conference on Human-computer interaction and visualization*, pages 142–162, 2009.
- [36] 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.
- [37] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *Computer Graphics*, 24(2):175–183, March 1990.
- [38] Alexander Wiebel, Christoph Garth, Mario Hlawitschka, Thomas Wischgoll, and Gerek Scheuermann. FAnToM - lessons learned from design, implementation, administration, and use of a visualization system for over 10 years. In *Refactoring Visualization from Experience (ReVisE) 2009*, 2009.
- [39] 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.
- [40] 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.

## Author Biography

*Thomas Wischgoll received his Master's degree in computer science in 1998 from the University of Kaiserslautern, Germany, and his PhD from the same institution in 2002. He was working as a post-doctoral researcher at the University of California, Irvine until 2005 and is currently an associate professor and the Director of Visualization Research at Wright State University. His research interests include large-scale visualization, flow and scientific visualization, as well as biomedical imaging and visualization. In the area of vector field visualization, Dr. Wischgoll completed the topological analysis of vector fields by developing an algorithm that detects closed streamlines, a missing link between branches of a topological skeleton. In the realm of biomedical engineering, he developed a visualization system that facilitates the analyses of large-scale vascular models of a heart represented geometrically by several hundred million polygons. The models are derived from CT scans and feature a simulated flow inside the blood vessels. Dr. Wischgoll developed methodologies for analyzing such volumetric data and extracting quantitative measurements at very high accuracy for further analysis. His research work in the field of large-scale, scientific visualization and analysis resulted in more than thirty peer-reviewed publications, including IEEE and ACM. Dr. Wischgoll is a member of ACM SIGGRAPH, IEEE Visualization & Graphics Technical Committee, and the IEEE Compute Society.*