Center for Cyber-Physical Systems: Immersive Visualization and Simulation Environment

Thomas Wischgoll*
Wright State University

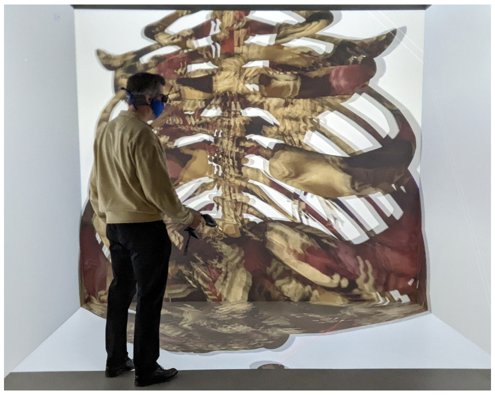

Figure 1: Anatomic visualization in a CAVE-type display system.

ABSTRACT

Immersive display systems in the form of head-mounted displays or full-size, walkable display systems can provide a very intuitive environment for a multitude of applications. Such applications include exploration, simulation for training, experimental studies to learn about people's behavior, and many more. Similarly, large-scale high-resolution display systems can also be very effective in data exploration and visualization. These systems can also be fully immersive. This paper describes the infrastructure available at Wright State University with its advantages and disadvantages and discusses some of its use cases as well as its setup and administration.

Index Terms: Computing methodologies—Computer graphics—Graphics systems and interfaces—Virtual reality; Computing methodologies—Computer graphics—Graphics systems and interfaces—Mixed / augmented reality

1 INTRODUCTION

The visualization and simulation infrastructure at Wright State University is supported by the Appenzeller Visualization Laboratory and the Immersive Visualization and Animation Theater. These two laboratories serve the common goal of making a variety of display systems available to the university. The Appenzeller Visualization Laboratory is more focused on the research side with some teaching components whereas the Immersive Visualization and Animation Theater provides students with 24/7 access to fully immersive display capabilities.

Right from the inception of this infrastructure, it was important to provide access to a variety of diverse display systems as different applications require different parameters. This diverse configuration also allows for direct comparison of different display environments to identify the most suitable one for a specific application or to interconnect display systems for a collaborative environment [6].

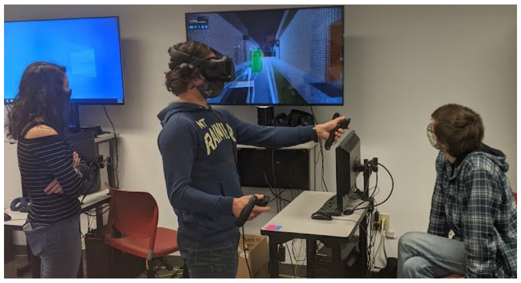

Figure 2: Students using an HTC Vive Eye head-mounted display with their custom-developed software also showing on the screen in the back.

This paper outlines the various display systems and environments available at Wright State University and discusses both the hardware and the software aspects to administer and support these environments in the following sections.

2 HARDWARE ENVIRONMENTS

To support the various activities in our laboratories, a diverse variety of display systems are installed. This ranges from desktop systems and head-mounted displays to large wall displays and walkable CAVE-type systems.

Head-mounted displays (HMDs) provide a cost-effective way to provide immersive display technologies to students and researchers. In our laboratories, the HTC Vive Eye HMDs and HP mixed reality devices are available. Figure 2 shows students exploring their custom-developed software using one of the head-mounted displays. We also utilize the Magic Leap One augmented reality devices. These stand-alone devices, similar to the Microsoft Hololens, provide an overlay image on top of the real world suitable for augmented reality applications. We have successfully used these devices for nursing education in which overlay images of fully animated organs and other internal structures are displayed on the traditional manekins [10, 12]. Another application was for assisting surgeons to visualize fractured ribs through the skin based on a CT scan during

*e-mail: thomas.wischgoll@wright.edu

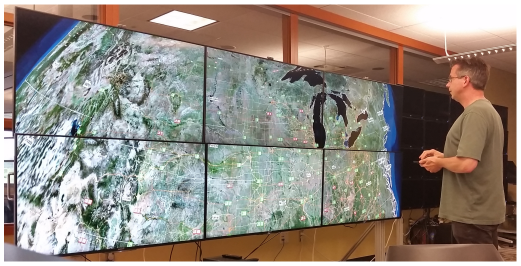

Figure 3: High-resolution tiled display system showing a GIS application based on OpenStreetMap.

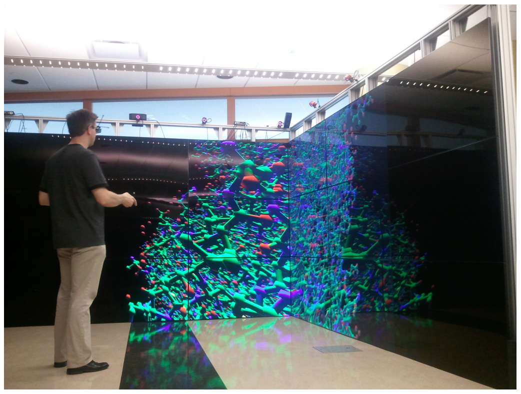

Figure 4: Molecular visualization using the DIVE system.

corrective surgery [11, 14].

Different projector-based display systems are available, such as a Barco CADwall and a mobile projection screen with support for passive stereo. For applications that require higher-resolution display environments, tiled configurations are a common way of supporting those applications. One of our configurations uses a 2-by-3 setup comprising Sony's 50-inch 4K TVs [17]. The entire system is driven by a single computer with two graphics cards to allow an easy deployment of standard software packages. To provide intuitive input modalities, this system uses a Logitech touchpad T650 to enable smartphone-style interaction metaphors.

An in-house built system, the Display Infrastructre for Virtual Environments (DIVE) [19], utilizes 27 55-inch full-HD LED-backlit displays with small bezels. Specifically, we used Samsungs UA55E large-format displays as those are commercial-grade displays. Arranged in a 3-by-3 configuration per wall using three walls, the system provides a 12-by-12 feet walkable footprint with a height of 87 inches. Each wall is driven by a single computer with three graphics cards running each of the Samsung displays in the side-by-side HDMI stereo mode. Since active stereo glasses are used for this system, all graphics cards are frame-looked using AMD's FirePro S400 sync cards. This provides a high-resolution display system bright enough to be used in an environment with the lights fully turned on. To interact with the system, a NaturalPoint OptiTrack optical tracking system is used as well as a Logitech wireless gamepad. In addition, our custom pinch glove uses the electronic components of a wireless mouse with contacts on the fingers and thumb wired to where the mouse buttons used to connect to. These pinch gloves are also fully tracked in 3D space using the optical tracking system.

To provide access to a fully immersive walkable display system, we chose the Virtalis ActiveCube to upgrade form the previous Barco I-Space. Our configuration includes a typical 10-by-10 feet footprint with projections on three walls and the floor. Figure 1 shows this system depicting an anatomical model of the rib cage and multiple organs. The system uses Barco's F80 series projectors which are laser-based projectors to avoid continuous bulb replacements. Each side wall uses two projectors resulting in about 2716 by 2716 pixels per wall. The system is combined with an A.R.T optical tracking system with a flystick2 for input.

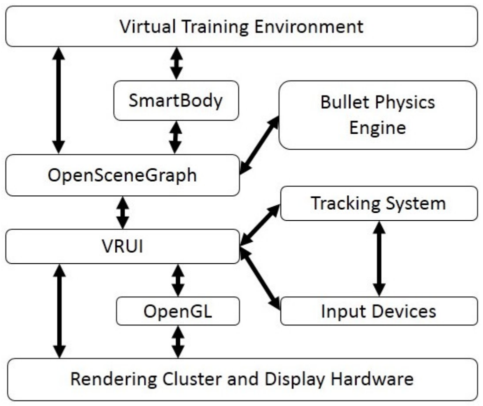

Figure 5: Software framework for rendering geometric content will full physics support.

3 Software Environments

Many of the display systems are powered by Linux or support a dual-boot environment in which Linux and Windows are installed side-by-side. Additional software is used to realize different virtual environments as outlined in the following sections. We have a variety of software packages installed, incuding ParaView [1] and FreeVR [15]. However, the ones listed in the sections below are the ones most frequently used.

3.1 VRUI

One of the libraries we have successfully used to develop more sophisticated software is VRUI created by Olliver Kreylos [7]. This library supports a variety of display configurations and is fully customizable with excellent support for different tiled display configurations run by one or more computers. Support for several input devices, such as gamepads, is also available. Keyboards and mice can be used as input devices as well. Hence, our pinch gloves using the electronic components of a wireless mouse is directly supported already. Multiple 3D tracking systems are supported either natively, such as the Intersense systems, or through network-based protocols, such as VRPN or A.R.T's dtrack. Support for additional input devices can also be added based on the already available drivers. For example, we implemented support for touch-based input devices. This allows us to utilize the Logitech T650 touchpad or any other touchscreen to enable smartphone-style input metaphors, such as pinch-zoom, rotation, or panning. This can provide a very intuitive input mechanism for many applications.

VRUI creates an OpenGL context for all the displays involved in the specific configuration. For simple setups, this may be just a single one. For more complex ones, multiple OpenGL contexts may be created. For example, the CAVE-type display using the Samsung TVs describes earlier renders onto a column of 3 TVs using a dedicated graphics card. Each wall is formed by a 3×3 configuration of TVs so there will be three OpenGL contexts created for each wall. This allows VRUI to fully take advantage of the accelerated 3D support provided by the graphics cards. VRUI then renders into those graphics contexts as needed.

The advantage of providing an OpenGL context is that any OpenGL-based application can be supported. One can use a traditional approach to render content using OpenGL commands to render the required geometry. However, any other OpenGL-based library can be used as well. For example, we also use OpenSceneGraph and have it render any geometric content into the OpenGL context. Any library building onto OpenSceneGraph can be supported in this way as well. Additionally, the bullet physics engine is used to provide a physics model to create realistic animations. Figure 5 shows a schematic layout of all the necessary components for this software setup.

The other big advantage of VRUI is that it is centrally configured with a few configuration files. In our case, these configuration files are located on the server that all of our Linux installation mounts. The parameters for all of our display systems are collectively specified in these configuration files. This then allows VRUI to identify the computer it is running on based on its hostname and grab the configuration for that system. As a result, the same software can then be run directly on all of our display systems without changing the software or even the need for recompiling it. This provides a very elegant software development platform in an educational environment by allowing students to develop on a simpler hardware configuration, for example, a 4K stereo-capable TV with Natural Point's optical tracking system as provided in the Immersive Visualization and Animation Theater. The student can then simply take their software and run it on the DIVE system, the Virtalis ActiveCube, or any other display system immediately without requiring changes to the software.

3.2 Unity

The game engine Unity has gained a lot of popularity within the virtual and augmented reality community. This is mainly due to the fact that many manufacturers support Unity directly for their devices. This is the case for pretty much all head-mounted virtual reality devices, such as the HTC Vive series or HP's mixed reality devices, but also for augmented reality devices, such as the Magic Leap devices or Microsoft's Hololens. In addition, Unity provides a relatively easy entrance to developing virtual and augmented reality software since there is a lot of support for a large variety of devices and controllers as well as additional support for developing 3D environments. Unity can also be used in traditional CAVE-type configurations. MiddleVR provides a commercially available integration for using Unity with CAVE-type displays [8]. An open-source integration was developed at the University of Wisconsin, Madison by Tredinnick et al. [16] that is freely available called Uni-CAVE. Since Uni-CAVE may require more configuration compared to MiddleVR, Davis et al. provide additional tutorials for setting up Uni-CAVE [2]. Once configured, Unity-based software developed for other devices, such as head-mounted displays, can be ported over by importing the content into the new Unity environment. Obvioulsy, all of the interaction mechanisms have to be ported over to the input devices used within the CAVE-type environment.

3.3 Visionary Renderer

Visionary Renderer is a commercial software package provided by Virtalis. Visionary Renderer can be run on a standard desktop computer or within a CAVE-type environment. It also supports head-mounted VR displays. Visionary Renderer is capable of ingesting a variety of different CAD formats natively. These models can then be explored or edited within the virtual reality environment.

The models can be fully animated through lua [5] scripting. This can be used to create simulated environments for training purposes of using some type of equipment or machinery, as just one example. Visionary Renderer provides a very high visual quality and flexibility in configuring the virtual environment.

3.4 VTK

VTK can also be used for creating virtual reality applications, particularly for the purpose of visualizing or exploring different types of data sets. VTK supports the HTC Vive directly in Linux and Windows. It requires SteamVR to be installed.

Another way of utilizing VTK is by combining it with VRUI. VTK can be used to render its content into an existing OpenGL context using the *vtkExternalOpenGLRenderer* class. This then allows us to support all of our display systems directly.

4 ADMINSTRATION

Running a virtual reality laboratory can be challenging from an administrative perspective. On the one hand, augmented and virtual reality devices and display systems tend to be pricy. The advent of modern head-mounted displays has brought down the cost of entry significantly with some headsets starting at just a couple hundred dollars. But many are still a $1000 or higher in price. Augmented reality devices tend to be even costlier and can go for even more than $3000. Walkable display systems are typically even more expensive with many CAVE-type display systems costing hundreds of thousands of dollars or more. However, in our experience, these display systems can be advantageous, especially for novice users as they are not completely detached from the real world as would be the case with a head-mounted display. It is also a little easier for the user to take off the 3D glasses compared to a head-mounted display. So in cases of nausea or dizziness, these types of display systems can be a benefit that may make the higher price tag worthwhile. Another advantage can be field-of-view. Head-mounted displays typically have a limited field of view of 140 degrees or less. The peripheral vision extends beyond that. We have performed studies in our laboratory where covering the full field of view including the peripheral vision was important which is why we chose the DIVE system for those studies [3].

Another issue with directing a virtual reality laboratory is maintenance. In our case, we do not have dedicated techs to support the systems. This makes it important for the systems and the software to require as little maintenance as possible. Based on the design choices we made for both the hardware and software configurations, we were fairly successful in achieving this goal.

4.1 Funding

The initial investment for the laboratories came from the Ohio Third Frontier program to set up the Appenzeller Visualization Laboratory augmented by private donations. These funds allowed us to purchase our first CAVE-type system, a Barco I-Space, and a large display wall, a Barco CADWall. Additional investments from the College of Engineering at Wright State University were used to add the TV-based CAVE-type display system (DIVE) and additional large tiled display walls. Recently, a grant from the Ohio Department of Higher Education funded a replacement for the original CAVE-type system, the Virtalis ActiveCube, and additonal head-mounted virtual and augmented reality displays. Additional grant funding from the U.S. Air Force, U.S. Army, and the National Science Foundation supported the software development of a variety of visualization and extended reality software.

Supporting a set of laboratories at this scale can be challenging. We have been very fortunate to have been able to fund our efforts through a variety of sources as outlined above. The majority of our display systems with the exception of the Barco CADWall and the Virtualis ActiveCube was designed and built in-house. This allowed us to significantly cut down on the cost of these display systems.

4.2 Maintenance

In our experience, Linux has served us well in supporting the diverse set of display environments available in our laboratories. The Windows operating system enforces updates which tend to break things at times. In Linux, we have updates disabled by default, and any updates are applied manually, typically on all systems at the same time. The installation is identical for all systems and ties in with a common server structure where home directories and additional software packages are installed and accessed from all clients driving the displays. This creates a very homogenous environment in which installed software packages become available to all clients immediately through the server. Similarly, the same software can be run on various display systems without requiring any changes using software libraries, such as VRUI. From an administration perspective, this makes this type of setup require significantly less support than Windows or other configurations. The Linux installation very rarely requires us to make any kind of adjustments.

We use a dedicated demo account with many of the software packages that we developed over the years and their configurations. This isolates all the settings and the software from any software development activities allowing us to have working versions of those software packages at all times in case we need to do a quick demonstration for an impromptu visitor.

We also try to find hardware configurations that require minimal maintenance. This is why our updated CAVE-type system, the Virtalis ActiveCube, utilizes laser-based projectors to avoid bulb replacements. This cuts down on maintenance, downtime, and cost all at the same time. Another maintenance issue with projection-based systems can be alignment. All of our systems that use more than a single projector and thus require precise alignment of the images use fixed projection screens made out of glass or plexiglass. This provides the most rigid setup. The Appenzeller Visualization Laboratory is temperature and humidity controlled with its dedicated air conditioning unit to avoid structural movement as much as possible.

The in-house built systems using different types of LCD displays or TVs have been very reliable as well. Over the years, we had to replace just a single TV for the DIVE system. This occurred fairly early on and was still covered under warranty. All the computing equipment driving these displays was all configured and assembled in-house and use high-quality name-brand components. As a result, we had very few hardware failures over the years.

4.3 Software Development

Developing software for augmented and virtual reality applications can be challenging in an educational environment. While for research projects, graduate students are usually developing a lot of code for a variety of software projects. Developing software for teaching purposes can be more challenging, albeit educational research funding is sometimes available for this purpose. However, graduate students typically develop prototype software for their research projects. Once the student graduates, there is a bit of a knowledge drain with respect to that software. As a result, managing larger software projects can be difficult in such an environment.

Providing students with basic frameworks to develop their software can make integrating various components into a larger framework easier. This then still poses the need for integrating the different components to form a larger software package with the additional requirement of testing the final version of the software. Alternatively, the students can contribute directly to a common software repository. Establishing coding standards, testing procedures, and best practices can help to ensure a quality standard of the resulting software.

5 Conclusion

In conclusion, the visualization and simulation environment at Wright State University has been very successful. The students appreciate and enjoy having access to this type of equipment. The laboratories have contributed to a variety of research projects. Especially the larger display systems are met with a lot of excitement from the general population during open-house events as well. These display systems were successfully used for experimental studies [13] and comparative visuslizations [9], for example. Virtual reality and simulations have also been shown to be effective tools for education and training with clear benefits to the learning process. We have used our virtual reality and simulation laboratories for training nurses [10, 12], medical personell [4], and our students [18].

Acknowledgments

The activities outlined in this paper are supported by the National Science Foundation, the U.S Air Force, the U.S. Army, the Ohio Third Frontier program, the Ohio Department of Higher Education, and private donations from Robert C. Appenzeller.

References

[1] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large data visualization. *The visualization handbook*, 717(8), 2005.

[2] C. Davis, J. Collins, J. Fraser, H. Zhang, S. Yao, E. Lattanzio, B. Balakrishnan, Y. Duan, P. Calyam, and K. Palaniappan. CAVE-VR and unity game engine for visualizing city scale 3d meshes. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 733–734. IEEE, 2022.

[3] B. R. Guthrie, P. Parikh, T. Whitlock, M. Glines, T. Wischgoll, J. Flach, and S. Watamaniuk. Comparing and enhancing the analytical model for exposure of a retail facility layout with human performance. In *Proceedings of the 2018 IISE Annual Conference*, 2018.

[4] P. J. Hershberger, Y. Pei, T. N. Crawford, S. M. Neeley, T. Wischgoll, D. B. Patel, M. M. Vasoya, A. Castle, S. Mishra, L. Surapaneni, et al. An interactive game with virtual reality immersion to improve cultural sensitivity in health care. *Health Equity*, 6(1):189–197, 2022.

[5] R. Ierusalimschy, L. H. De Figueiredo, and W. Celes. Lua 5.1 reference manual, 2006.

[6] C. Koehler, A. Berger, R. Rajashekar, T. Wischgoll, and S. Su. Dynamic collaborative visualization ecosystem to support the analysis of large-scale disparate data. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3968–3977. IEEE, 2019.

[7] O. Kreylos. Environment-independent vr development. In *International Symposium on Visual Computing*, pp. 901–912. Springer, 2008.

[8] S. Kuntz. MiddleVR a generic VR toolbox. In *2015 IEEE Virtual Reality (VR)*, pp. 391–392. IEEE Computer Society, 2015.

[9] M. Marangoni and T. Wischgoll. Comparative visualization of protein conformations using large high resolution displays with gestures and body tracking. In *Visualization and Data Analysis 2015*, vol. 9397, pp. 135–147. SPIE, 2015.

[10] S. S. Menon, C. Holland, S. Farra, T. Wischgoll, and M. Stuber. Augmented reality in nursing education–a pilot study. *Clinical Simulation in Nursing*, 65:57–61, 2022.

[11] S. S. Menon, T. Wischgoll, S. Farra, and C. Holland. Medical education and assisted surgery by ar. In *The Campus Alliance for Advanced Visualization (THECAAV20)*, 2020.

[12] S. S. Menon, T. Wischgoll, S. Farra, and C. Holland. Using augmented reality to enhance nursing education. *Electronic Imaging*, 2021(1):304–1, 2021.

[13] P. Parikh, B. R. Guthrie, T. Whitlock, M. A. Glines, T. Wischgoll, and J. Flach. Validation of models to estimate exposure in a retail layout using a 3d virtual environment. *Industrial and Systems Engineering Research Conference (ISERC)*, 2018.

[14] T. Sensing, P. Parikh, C. Hardman, T. Wischgoll, and S. S. Menon. Augmented reality headset facilitates exposure for surgical stabilization of rib fractures. In *16th Annual Acadmic Surgical Congress*, 2021.

[15] W. R. Sherman, D. Coming, and S. Su. Freevr: honoring the past, looking to the future. In *The Engineering Reality of Virtual Reality 2013*, vol. 8649, pp. 47–61. SPIE, 2013.

[16] R. Tredinnick, B. Boettcher, S. Smith, S. Solovy, and K. Ponto. Unicave: A unity3d plugin for non-head mounted vr display systems. In *2017 IEEE Virtual Reality (VR)*, pp. 393–394. IEEE, 2017.

[17] T. Wischgoll. Display systems for visualization and simulation in virtual environments. *Electronic Imaging*, 2017(1):78–88, 2017.

[18] T. Wischgoll. XR-based workforce develop in the southwestern region of ohio. *The Campus Alliance for Advanced Visualization (THECAAV21)*, pp. 27–28, 2019.

[19] T. Wischgoll, M. Glines, T. Whitlock, B. R. Guthrie, C. M. Mowrey, P. J. Parikh, and J. Flach. Display infrastructure for virtual environments. *Electronic Imaging*, 2018(1):060406–1, 2018.