Software-Enhanced Teaching and Visualization Capabilities of an Ultra-High-Resolution Video Wall

Ramses van Zon
Marcelo Ponce
rzon@scinet.utoronto.ca
mponce@scinet.utoronto.ca
SciNet HPC Consortium, University of Toronto
Toronto, Canada, Ontario

ABSTRACT

This paper presents a modular approach to enhance the capabilities and features of a visualization and teaching room using software. This approach was applied to a room with a large, high resolution (7680×4320 pixels), tiled screen of 13 × 7.5 feet as its main display, and with a variety of audio and video inputs, connected over a network. Many of the techniques described are possible because of a software-enhanced setup, utilizing existing hardware and a collection of mostly open-source tools, allowing to perform collaborative, high-resolution visualizations as well as broadcasting and recording workshops and lectures. The software approach is flexible and allows one to add functionality without changing the hardware.

CCS CONCEPTS

• **Applied computing** → **Education**; **Interactive learning environments**; *Collaborative learning*; • **Hardware** → **Displays and imagers**; • **Human-centered computing** → *Collaborative and social computing devices*; *Scientific visualization*; • **General and reference** → Evaluation.

KEYWORDS

Education and teaching, Virtualization, Scientific Visualization, UHD video wall.

1 INTRODUCTION

In the summer of 2016, the offices of the SciNet HPC Consortium at the University of Toronto in Canada were moved to a new location. Its previous location had a boardroom which also served as

a video conferencing room, as well as the room for training and courses, which would usually be recorded and then posted, and sometimes also broadcast. But because of increasing demand for SciNet's training workshops and courses, the room was almost never large enough. In addition, there never was a proper space for visualizations, as the screens in that room were neither particularly large nor of particularly high resolution.

As we were designing our new office space, we had the vision of a collaborative visualization space with a large, tiled, ultra-high-resolution display. The utilization of ultra high-definition video walls in different scientific fields offers several advantages in many different aspects [7, 9], ranging from higher resolution definition visualizations to sharpening wider fields of view. The same room would also be used as a teaching room, while a separate boardroom across the hall could mirror the session and thus serve as an overflow room. And because SciNet serves a geographically distributed user base, it should be possible for lectures, training sessions and meetings to be recorded and broadcast.

While it was clear from the onset that supporting software would be needed to get all envisioned functionality, the scope of the multi-server software setup turned out to be much larger and involved than anticipated.

The hardware components provide a 100 square feet, 8K capable tiled display in a new teaching/visualization room, which can switch between being driven from a Windows PC server and a dedicated visualization GPU server running Linux. Cameras were installed in the front and the back of the room, with zoom and pan capabilities through a control screen in a lecture console. The same console screen can mirror the large screen. The lecture console has microphones and there are wireless microphones for the room. Sharing of the screens and of sound is possible with the boardroom's two 4K monitors. There are also inputs for laptops, Apple TV and Chrome-cast. In addition, there is a projector screen, and a possibility for later hook up of a standard university teaching station PC, which would allow this room to be used as a classroom by other groups within the University of Toronto.

Not all of the envisioned capabilities were realized out of the box. For instance, the maximum resolution of the PC server or external laptops that could make it through the various hardware layers and cables, is merely 1080 vertical lines. The visualization server, meanwhile, can run in full 8K mode because its output bypasses the hardware interface in charge of multiplexing (i.e., choosing between the projector or video wall and sharing), but that meant it could not be shown on the preview window in the lectern console. The solution contained no component that allowed online streaming of
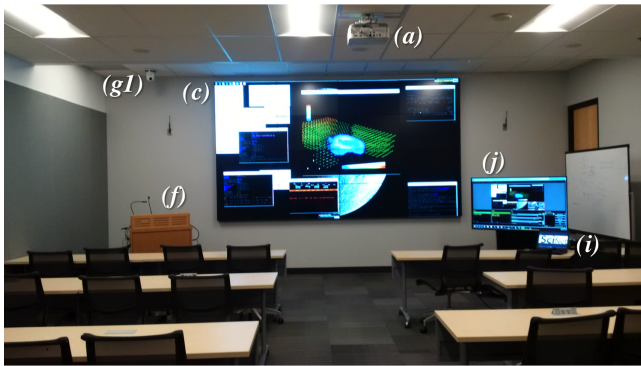
Figure 1: A view of SciNet's teaching-visualization room located at SciNet's headquarters in the University of Toronto St. George campus.

video and sound; this would require another dedicated hardware appliance.

We used a variety of software to enable most of the envisioned capabilities within the given hardware setup. This paper explains the software stack utilized for driving the video wall, and for recording and broadcasting. It ends by describing the several use cases of the current setup. In our experience, vendors of A/V equipment are not necessarily aware of the software possibilities, and the modular, multi-server setup presented here appears to be novel. We hope it can be of help to places that would like to set up a similar facility.

2 HARDWARE COMPONENTS OF THE VISUALIZATION AND TEACHING ROOM

To understand how the software running on the server(s) of the teaching room facilitates the room's functionalities, we will first describe the hardware setup.

Figure 1 shows a picture of the teaching room. A schematic diagram of the hardware layout of the visualization system is shown in Figure 2.

The most visible part of the hardware setup is the video wall system (*c*). This is a Clarity Matrix LCD Video Wall [10] by Planar [11] composed of a 4 by 4 grid of 46 inch displays with a small bezel [12].

The total resolution is 7680×4320 pixels (i.e., 8K or UHD resolution). To supply content to the 16 tiles, they are grouped in units of 2×2 tiles. Each unit acts as a single display; this is accomplished by four hardware splitters (*b*). As a result, the video wall can be driven as if it were a 2×2 tiled display, and that allows the whole video wall to be driven from just one high-performance graphics card with 4 output ports, hosted by a single high-end visualization server, avoiding the need for a cluster to drive the display (in contrast to e.g. Ref. [6]). [1]

In addition to the dedicated visualization server that directly drives the video wall at its full resolution, it can also be driven by a Crestron Matrix Switcher. Because the tiled display is also called a

matrix, we will refer to the Crestron Matrix Switcher simply as the *Switcher*. The Switcher has multiple inputs and outputs, which can be controlled from the console.

From the point of the Switcher, the video wall is just one of the outputs. Other available output devices connected to the Switcher are a projector (*a*) in the teaching room, and the displays in the boardroom. The Switcher can work in a sharing mode as well, where the audio and video from the teaching room are also played in the boardroom, or vice versa.

The switcher system was designed to be HD capable in order to be compatible with many input and output devices, but this does mean it currently cannot redirect 8K or even 4K output to the video wall. This is the reason the visualization server is directly connected to the video wall instead of also being routed through the switcher, but that in turn means the output of the visualization server cannot be shared with the boardroom nor shown on the projector screen (devices which are presently not 8K capable at any rate).

Other possible inputs to the Switcher are the cameras ($g1, g2$), laptops, Apple-TV and Google Chrome-cast, as well as a PC server (*e*) intended for presentations and applications that either requires *Microsoft Windows* or needs to be shown on the projector screen.[2]

3 REQUIRED FUNCTIONALITY

The need for an additional software layer arises from the different kinds of usage that the system needs to support. Among the most common usages are:

(1) High-resolution graphics: Using the visualization server to generate images or animations of data visualizations. This is mostly useful for research and outreach.

(2) Lectures and seminars: The video wall shows presentation material. Optionally, the lecture can be shared with the boardroom, or a second display could show presentation controls. Lectures are still a frequently used educational vehicle.

(3) Recording: like lecturing with added sound capture and optionally including the camera showing the lecturer. Requires post-processing to produce the video to be disseminated online. Such recordings are frequently requested by students, especially in lecture series, where they may have missed some lectures. Recordings can also form the basis of producing an online course or inverted classroom setup.

(4) On-line lecturing (or broadcasting): Instead of recording, the presentation and the camera output need to be combined and made available as an online stream in some standard format. Broadcasting is particularly useful when the audience of a lecture or seminar is geographically dispersed.

(5) Video conferencing: This requires a two-way communication of video and optional screen sharing. It is sometimes desirable to have multiple displays; one for the presentation material and one to see the presenter. Apart from video meetings, this capability is also useful in remote instruction.

(6) Visualization collaborations or demonstrations: One or several people might want to inspect and interact with an 8K visualization on the video wall. In case of a demonstration,

---

[1]The visualization server (*d*) contains two 14-core Intel Xeon CPUs (E5-2680v4) at 2.40GHz, has 256GB of RAM, and an NVIDIA Quadro M400 graphics card, which can handle up to four outputs. The operating system of the visualization server is *openSUSE Leap* 42.2.

[2]The PC server has two 4-core Intel Xeon CPUs (E5-2637v4) at 3.50GHz, with 128GB of RAM, and runs *Microsoft Windows* 10 as its operating system.

recording should also be possible. This can be useful for research, training, and outreach.

In the next section, we will explain how our audio-visual setup that was designed for only some of these functions, could be extended to provide all desired functionality.

4 MODULAR SOFTWARE APPROACH

In order to fully exploit the features and capabilities of the visualization and teaching room, we used a series of software packages to augment the hardware capabilities, which we will describe in this section.

The software solution relies heavily on a (wired) network. Through the network, video and control signals can be exchanged between various servers (e.g. GPU server, PC, laptops, and mobile devices), using software solutions such as VNC[13, 14], OBS[2], and nginx-rtmp[5]. Wireless capabilities are added through a dedicated router (so that connected devices live on the same network and are not blocked by firewall rules).

The first three ways of using the system described in the previous sections, i.e., high resolution graphics, lecturing and recording, require very few additional software tools, as long as lecturing and recording at 1920×1080 resolution is acceptable. The camera feed is accessible from the PC server, on which common presentation software as well as recording software can be run. For recording, we at first installed *Camtasia* [15] (which has useful post-processing features), but later also the free software *ActivePresenter* [1] and the open-source *Open Broadcaster Software* [2].

The *Open Broadcaster Software* also enables the fourth point, online lecturing, except for a final step: to broadcast to the internet, one needs the feed from the software to be pointed to a *streaming server* which can then stream it through a standard protocol. Some streaming servers are available online, but require at least an account. To have our setup be independent of an external site, we instead installed an *nginx*-based media streaming server, the *nginx-rtmp* module (*nginx* version 1.10, *nginx-rtmp* version 1.13) [5], inside a *Ubuntu* 16.04 virtual machine that runs on the PC server using *VirtualBox* [4]. This same module allows streaming and recording at the same time.[3]

Video conferencing or remote participation, the fifth required functionality from the previous section, is possible with the PC server through a variety of proprietary applications, such as Skype, Vidyo and Zoom. This requires no further help from software, unless one wants to have a two-monitor setup, one with slides and one with the presenter, for instance (we will come back to this capability below).

The case of collaborative visualization or demonstrations turned out to be the most involved to get to work. In this case, a group of people might be interactively involved in the visualization, perhaps walking around and inspecting different parts of the displayed graphics, or one person is interacting with the visualization while explaining it to an audience. One might want to zoom in on particular aspects, or change parameters, while looking at the video wall. The challenge is to overcome the restriction that the visualization

server can only be controlled from the console, so one could not walk around and manipulate the graphics. A further complication is that the console faces away from the video wall and doesn't show a preview of the screen, because the console can only show output from the Switcher, which the visualization server bypasses.

To make the setup suitable for such interactive usage, we first start a Virtual Network Computing (VNC) server on the 8K visualization server ("VNC server1" in Figure 2). Because VNC [13] is a protocol to view and interact with a remote screen, where the VNC server captures the screen and sends this to VNC clients on other computers, one can have a tablet running the VNC client showing and controlling the same screen that is showing on the video wall.

The VNC server on the visualization server that we picked for this was *x11vnc* [14], which is capable of server-side scaling on Linux, thus bringing the transfer rate needed to mirror the 8K desktop image down to something that a laptop or tablet can handle over a typical wireless network.[4] The laptop or tablet can then be used to control the video wall without being tethered to the lecture console. A second scaling VNC server ("VNC server2" in Figure 2) on the GPU servers sends a signal to a VNC client on the windows PC that matches the PC's resolution. The Windows PC also captures the camera input, and runs the capturing software, enabling a recording of the event at the lower resolution of 1920×1080, while the video wall showed the full 8K visualization. In addition, the PC server's screen was visible on the lecture console, so lecture-style demonstrations were still possible. See Figure 2 for a detailed illustration.

Finally, the addition of further displays or monitors can be of use in presentations and remote instruction. The hardware was not designed with this in mind, but it is possible to do this in software, such that the system could have a secondary monitor which is, e.g., a 55" screen or a smaller computer monitor (see (*j*) in Figure 1). The first option (an additional monitor) can be useful in the case of having a remote participation or demonstration, where the video wall is used to display the main presentation or participation, whereas the additional screen can be used to show the presenter or other details related to the main visualization. The second setup (using an additional laptop) can be useful as an aid to the presenter, where information about the active presentation in the video wall is visualized privately on the laptop for the presenter, eg. slideshows in presenter mode, or confidential material not to be shared through the videowall.

In order to achieve this, a Windows laptop was added to the setup (*i*), which was connected to the additional display (*j*). We use a software named spacedesk[3]. The spacedesk server portion runs on the Windows server (*e*), while the spacedesk client runs on the laptop (*i*). What spacedesk accomplishes is that the client's display becomes visible as a second monitor on the Windows server. To be able to run spacedesk, there are restrictions on the server side: The spacedesk server requires Windows 10 as its operating system (which the Windows server has). However, the client (here, the additional laptop) can run an older version of Windows, an Apple device, or an android device. The second display could therefore even be a portal mobile device, such a tablet or a smartphone (*k*).

---

[3]The default streaming protocol of this module is "rtmp", which is associated with flash and not universally supported anymore, but it should be possible to switch to different protocols like "hls" and "dash."

[4]The PC server can be driven remotely as well, using a VNC server for *Microsoft Windows*, such as *UltraVNC* ("VNC server3" in Figure 2).
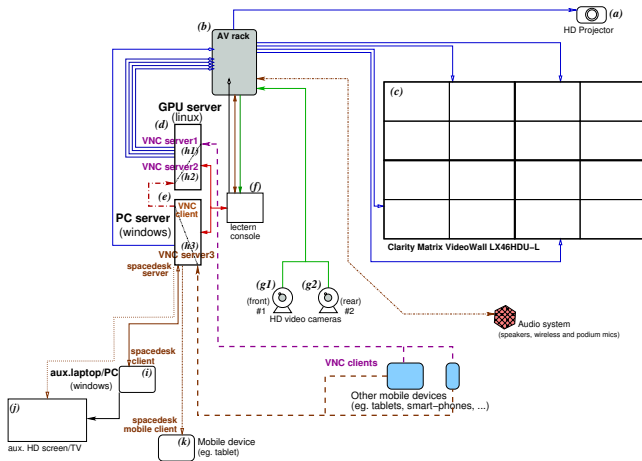
Figure 2: Diagram of our video wall system, including computer servers, audiovisual media, as well as software connection using VNC, denoted by dashed lines in the figure. Notice that the AV rack, in addition to provide the hardware infrastructure for connecting all possible medias, also serves as large multiplexer allowing to share some of the different resources among the GPU and PC servers.

The result of combining, VNC, OBS, nginx-rtmp, and other software, is that all six capabilities from the previous section are realized in the teaching/visualization room.

There are other software approaches/systems to drive and utilize large video walls. All of them present substantially different scopes, goals and higher complexity to the solution presented here. For instance, *SAGE2* –Scalable Amplified Group Environment– [8] from the Electronic Visualization Laboratory at the University of Illinois at Chicago (EVL/UIC), is an interesting tool for sharing files and images through a web-browser interface. Its utilization is quite simple from the user perspective, however its main and default features are mostly limited to upload and manipulate certain predefined file types (images, videos, documents) into the video wall. It is also possible to write specific applications for taking advantage of *SAGE2*, but complex software and complicated software interfaces/interactions, such as the ones described in the following section would not be possible to achieve in this framework. *Display Cluster* [6] from Texas Advanced Computing Center (TACC), offers a more complex and elaborated approach to driving a tiled display, fulfilling these and other features. However it requires a hardware setup much more sophisticated than the one presented here, basically it is driven by a cluster-type server. Moreover, we should emphasize that the type of hardware, ie. video wall we have, has to be handled as just one input/output connection due to a design requirement to be able to connect many different input devices to the system.

5 CONCLUSION

In this paper, we have presented a versatile and modular setup to drive, manage and enhance the utilization of a video wall system in

its native UHD resolution for visualization, lecturing, broadcasting, recording, video conferences, collaborations, and demonstrations.

The novelty we would like to emphasize here is the relative simplicity, flexibility and cost efficient approach we proposed. One limitation with the current setup is related to the precise GPU model our server uses, which does not have the capability of rendering 8K videos in hardware. But given the modularity and flexibility of the current approach, such a restriction can easily be overcome by upgrading the GPU model that does have this capability. Flexibility of the setup is also important to be able to incorporate new features and modern technology-enhanced teaching techniques.

This software-enhanced solution will be applicable to other systems like this, if not in its entirety then at least in parts. The approach augmenting the existing capability of a video wall using software, is more efficient in terms of cost and flexibility than redesigning and upgrading the hardware system.

REFERENCES

[1] ActivePresenter, ver. 6.0.5. https://atomisystems.com/activepresenter. Accessed: 2017-06-09.
[2] Open Broadcaster Software, ver. 17.0.0. https://obsproject.com. Accessed: 2017-06-09.
[3] spaceDesk: Windows Network Display Monitor Software. http://spacedesk.ph/. Accessed: 2017-10-18.
[4] Virtual Box, ver. 5.1.22. https://www.virtualbox.org. Accessed: 2017-06-13.
[5] Roman Arutyunyan. Nginx-based media streaming server. https://github.com/arut/nginx-rtmp-module. Accessed: 2017-06-12.
[6] Gregory P. Johnson, Greg Abram, Brandt M. Westing, Paul A. Navrátil, and Kelly P. Gaither. DisplayCluster: An Interactive Visualization Environment for Tiled Displays. *2012 IEEE International Conference on Cluster Computing*, pages 239–247, 2012.
[7] Aditi Majumder and Behzad Sajadi. Large area displays: The changing face of visualization. *Computer*, 46(5):26–33, 2013.
[8] Thomas Marrinan, Jillian Aurisano, Arthur Nishimoto, Krishna Bharadwaj, Victor Mateevitsi, Luc Renambot, Lance Long, Andrew Johnson, and Jason Leigh. Sage2: A new approach for data intensive collaboration using scalable resolution shared displays. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*, pages 177–186. IEEE, 2014.
[9] B. F. Meade, C. J. Fluke, S. Manos, and R. O. Sinnott. Are Tiled Display Walls Needed for Astronomy? *Publications of the Astronomical Society of Australia*, 31:e033, August 2014.
[10] Planar. Clarity Matrix LCD Video Wall System. http://www.planar.com/media/436303/clarity-matrix-lcd-video-wall-system-brochure-2-17.pdf. Accessed: 2017-06-09.
[11] Planar. Digital displays and signage solutions (company website). http://www.planar.com. Accessed: 2017-06-09.
[12] Planar. Clarity matrix LX46HDU LCD video wall system. Technical report, Planar, June 2017. Accessed: 2017-06-09.
[13] T. Richardson, Q. Stafford-Fraser, K.R. Wood, and A. Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1):33–38, 1998.
[14] Karl J. Runge. x11vnc: a VNC server for real X displays, ver. 0.9.13. http://www.karlrunge.com/x11vnc/. Accessed: 2017-06-12.
[15] TechSmith. Camtasia Video Editing Software for Windows and Mac, ver. 9.0.1. https://www.techsmith.com/camtasia.html. Accessed: 2017-06-09.