# High-resolution Data Visualization*

**Jessica David**
University of Evansville

*University of Maine at Orono*
*Advisors: Dr. Bruce Segee, Jason Withee, Nathan Bourgoin*

**Abstract:** High-resolution data visualization is useful when wanting to look at data on a larger scale while maintaining finer details. While using a projector helps show information on a larger scale, the pixels are only enlarged and detail is lost due to the inflated pixel size. By using a grid of monitors, however, one gains a higher resolution and still maintain the details, as well as being able to display more information than a single monitor. The research in this paper focuses on using VNC (Virtual Network Computing) to create high-resolution displays and for the monitor wall in the Innovation Center at the University of Maine in Orono and dynamic walls using students' and teachers' laptops for the MLTI (Maine Learning Technology Initiative) program. Edits were made to allow the VNC client to access a specific region of the server's desktop, and tests were performed to discover the performance differences between RealVNC and TightVNC on both of the walls.

**Keywords:** *high-resolution visualization, VNC, virtual network computing, RealVNC, TightVNC, MLTI, Maine Learning Technology Initiative*

* Research is co-funded by the ASSURE program of Department of Defense in partnership with the National Science Foundation REU Site program under Grant No. 0754951.

## Introduction

High-resolution displays created from many smaller monitors or projectors provide an opportunity to see a large amount of information at a greater detail than any commercial product today. Tiling monitors can increase the amount of screen space while maintaining details that would be lost if one was using a projector or other method that would only enlarge pixels. Research is also being performed to examine the effects larger displays have on performance and increased user productivity [1]. The proposed research will look into more efficient ways of create visualization walls, including dynamic walls created by laptops.

## I. Existing High-resolution Walls

Various other display walls have been constructed along with the wall being used at the University of Maine. Some examples include HIPerWall, or Highly Interactive Parallelized Display Wall, at the California Institute for Telecommunications and Information Technology [2] and SAGE (Scalable Adaptive Graphics Environment) at University of Illinois at Chicago [3]. These walls are all constructed of many monitors to create a high-resolution display and range from 100 to 200 Megapixel displays.

While these examples are certainly innovative and useful to consider for researching purposes, they (talk about the difference between their walls and our wall) (talk about the difference between their walls and our wall) (talk about the difference between their walls and our wall) (talk about the difference between their walls and our wall) (talk about the difference between their walls and our wall) (talk about the difference between their walls and our wall) (talk about the difference between their walls and ours)

## II. VNC

VNC, or virtual network computing, is a “remote control software which allows you to view and fully interact with one computer desktop (the "VNC server") using a simple program (the "VNC viewer") on another computer desktop anywhere on the Internet” [4]. Software does not even need to be installed, as there is a Java viewer that enables the desktop to be controlled via a browser. In the sense of a large display wall, one computer serves up a large display to be shown on all of the monitors. Keyboard and mouse events are sent from the server to the client, and communication between the two is handled by the RFB protocol.

(talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here) (talk about types of servers and clients here)

## III. The RFB Protocol

Part of the researching involved reading through the RFB Protocol [5]. RFB, or read frame buffer, is “a simple protocol for remote access to graphical user interfaces” and the protocol used in VNC. It is cross compatible, therefore usable in Windows and *nix operating systems because it works at the framebuffer level, and it is designed to make very few requirements of the client machine.

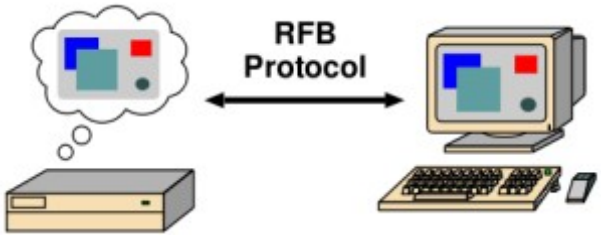


*Figure 1: Graphical interpretation of commucation using the RFB Protocol. The server is shown on the left and the client on the right. From [rfb]*

The server serves up a display that can be accessed by the client machine and the state of its interface, upon disconnecting from the server, is maintained when the client machine reconnects, and the state is preserved between multiple locations.

The communication begins with a “handshaking” process between the server and the client in which information about pixel format and encoding is exchanged. The client will request a specific format and encoding from the server, and while the server must be able to display the requested pixel data, the client may choose a combination that is easier for the server to produce. Typical pixel formats are either 24- or 16-bit color; current encodings are Raw, CopyRect, RRE, Hextile and ZRLE, and Tight is a new encoding offered in TightVNC. The version of the RFB protocol to be used and security information is also exchanged in the initial handshaking, and both the client and server send messages back and forth to determine the session settings.

## IV. Innovation Wall

The Innovation Wall is located at the Foster Student Innovation Center at the University of Maine in Orono [6]. The Wall is constructed of sixteen seventeen-inch Dell LCD monitors in a four-by-four grid. Each monitor has a resolution of 1280 by 1024 pixels, giving an overall resolution of 5120 by

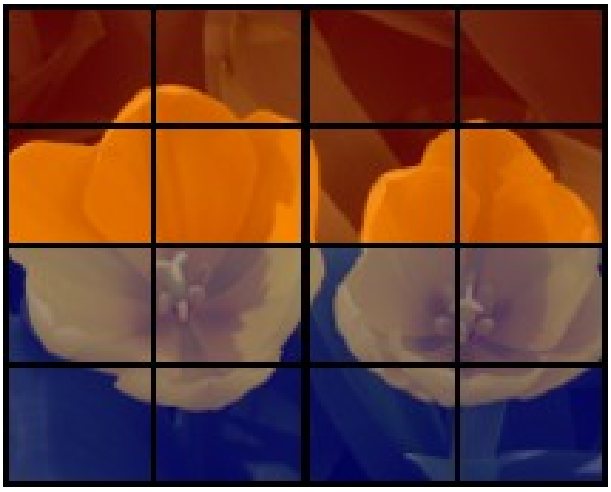

*Figure 2: Design of the Innovation Wall. The red-shaded area represents node1 and the blue-shaded area node2.*

4096 pixels for a nearly 21 Megapixel display. The top eight montiors are controlled by one computer, named node1, and the bottom are controlled by another named node2. These display nodes currently run Xubuntu 8.04 [7] and have four video cards with dual DVI output, with each card controlling a vertical pair of displays.

The large display is created by another computer, called appnode. This head node also runs Xubuntu 8.04 and contains programs such as Google Earth and other VirtualGL programs for the wall to display. The head node starts up a session of a VNC server at the correct geometry and color depth, and the two display nodes then start their own session of a VNC client to display the desktop from the server.

## V. Dynamic Wall

Dynamic Walls [8] are created when using laptops from the MLTI program. A dynamic wall server is set up to create custom sessions for teachers to use in the classroom. When a session is created, the teachers are asked to give the following:

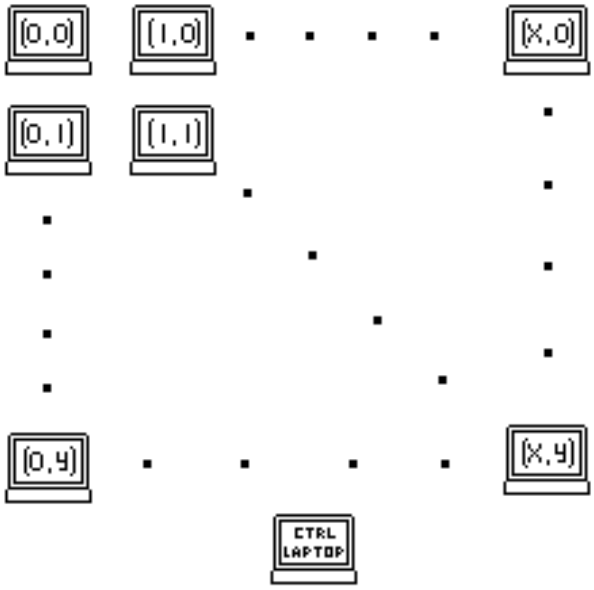


*Figure 3: Dynamic wall of laptops. The laptops are arranged in a grid and the control laptop powers the display across all of them.*

1. A session name
2. A session password
3. The length of the session in minutes
4. The laptop grid size
5. The monitor resolution of the laptops

Once the session is created, a desktop is created by the server at a size of (number of laptops wide) x (resolution width) by (number of laptops tall) x (resolution height). The user is then prompted to log in to the session and redirected to the session page. The page displays a grid of laptops, a control computer image, and a list of images available for slideshows. Each user should then click on the laptop in the grid of images associated with the location of their laptop or the control laptop if they are to have control of the whole screen. A VNC Java applet is then started showing their section of the whole display, and the control displays a scaled down version of the entire desktop.

## Methods

Initially, the nodes were running various versions of Linux and were inconsistent in the programs and distributions they possessed. Clean installations of Xubuntu 8.04 were performed on all of the machines to make a clean start. For the version of VNC to use, TightVNC was initially chosen be the distribution of choice for the display nodes. RealVNC was then edited based off of those modifications, and both the C code and Java code was edited for each to see results both on the Innovation Wall and Dynamic Wall.

### I. -region edit in C code

To make the modifications needed to TightVNC's client, software was discovered that had been developed and utilized by NCSA at the University of Illinois at Urbana-Champaign. Called Display Wall-in-a-Box, or DWiB [9], there were several materials available for display wall development, including a version of VNC that, while outdated, contained a “region” argument that would allow access only certain areas of big display. By analyzing the implementation of region in Tile Viewer, their version of VNC Viewer, modifications were made to TightVNC to accommodate for region access.


*Figure 4: Example of the implementation of region. The server display is the whole image, and the requested region is what the user would see on the client side.*

First, Tile Viewer was run on the display nodes to understand the behavior of the region argument. The argument took four options, two for the x- and y-coordinates and two for width and height, and displayed the desired section of the server's display starting at the given point (x, y) at the given width and height. Upon further inspection of Tile Viewer's code, it was clear that it definitely used to be developed alongside TightVNC Viewer. Therefore, by finding where edits were made in Tile Viewer from the original code, it would be easier to find where adjustments would need to be made in the current version of TightVNC. By comparing the older files with the most recent version, the differences were found and work began on editing TightVNC.

### II. -region edit in Java code

Text text text.

### III. Testing on Innovation Wall

Text text text.

### IV. Testing on Dynamic Wall

Text text text.

## Results

### I. Performance on Innovation Wall

Text text text.

### II. Performance on Dynamic Wall

Text text text.

## Discussion

Need list of discussion topics, including difficulties with 64-bit, further investigation into the server code, etc.

## Literature Cited

[x] “Foster Student Innovation Center,” July 2008. http://www2.umaine.edu/innovation/.

[x] “Ubuntu Home Page,” July 2008. http://www.ubuntu.com/.

[x] R. Blanchette, “Hardware and Software Solutions for Low Cost Distributed Visualization ,” Master’s thesis, University of Maine, 2008.

[x] “Display Wall-in-a-Box,” June 2008. http://www.ncsa.uiuc.edu/Projects/DWiB/.

[x] http://dynamicwall.eece.maine.edu/.

[x] T. Richardson, “The RFB Protocol,” June 2008. http://www.realvnc.com/docs/rfbproto.pdf.

[x] “Scalable adaptive graphics environment (sage),” June 2008. http://www.evl.uic.edu/cavern/sage/index.php.

[x] “Hiperwall,” June 2008. http://hiperwall.calit2.uci.edu/