Scalable Graphics Architecture for High-Resolution Displays

Byungil Jeong+, Ratko Jagodic, Luc Renambot, Rajvikram Singh, Andrew Johnson, Jason Leigh
Electronic Visualization Laboratory, University of Illinois at Chicago

1 INTRODUCTION

We envision situation-rooms and research laboratories in which all the walls are made from seamless ultra-high-resolution displays fed by data streamed over ultra-high-speed networks from distantly located visualization, storage servers, and high definition video cameras [1,6]. It will allow local and distributed groups of researchers to work together on large amounts of distributed heterogeneous datasets. We are taking the next steps toward this vision by building LambdaVision - an 11x5 tiled display with a total resolution of 100 megapixels and developing SAGE, the Scalable Adaptive Graphics Environment (see Figure 1). SAGE allows the seamless display of various networked applications over the high-resolution displays. Each visualization application (such as 3D rendering, remote desktop, video streams, very large 2D maps) streams its rendered pixels (or graphics primitives) to SAGE, allowing for any given layout onto the displays (e.g. the output of arbitrary M by N pixel rendering cluster nodes can be streamed to X by Y pixel display screens).

The graphics streaming architecture of SAGE addresses two non-trivial problems in scientific visualization. One is heterogeneity: since most visualization applications are closely tied to their graphics environment, it is difficult to integrate various visualization applications into a unified graphics environment. The other is scalability: the ability of visualization software and systems to scale in terms of the amount of data they can visualize and the resolution of the desired visualization [5]. SAGE addresses the heterogeneity problem by decoupling graphics rendering from graphics display so that visualization applications developed on various environments can easily migrate into SAGE by streaming their pixels into the virtual frame buffer. Also, SAGE provides scalability by supporting any number of rendering and displaying nodes, number of tiles, and screen resolution, and the SAGE visualization applications have extremely fast access to huge databases at remote or local sites taking advantage of affordable ultra-high-bandwidth networks.

There are several other existing systems with parallel or remote rendering schemes related to SAGE. WireGL [7] for example uses sort-first parallel rendering scheme from a single source or parallel scene-graph rendering. This approach allows a single serial application to drive a tiled display by streaming graphics primitives that will be rendered in parallel on display nodes. However, it has poor data scalability due to its single source limitation. Flexible scalable graphics systems such as Chromium [8] and Aura [10] are designed for distributing visualizations to and from cluster driven tiled-displays. They have a similar parallel rendering scheme with SAGE on local area network; however, Chromium is not designed to execute multiple applications on a tiled display, and its applications have a static layout on the tiled display – it can divide the tiled display into several parts and execute multiple applications, but each tile can support only one application. Chromium's DMX extension allows execution of multiple applications and window moving and resizing but does not support parallel applications – it has a single source (serial application) and its design is not suitable for graphics streaming over wide-area networks. IBM's Scalable Graphics Engine [9] is a hardware-based approach that allows for the reception of pixels streamed over networks and can drive an array of displays synchronously. Currently, it is limited to 16 1GigE network inputs, 4 DVI outputs, and SGE specific network protocol which prevents users from exploiting advanced network protocols. Also as 10GigE (10 gigabit Ethernet) network becomes prevalent, 1GigE network interface becomes a drawback of SGE. Our previous work, TeraVision [3], is a scalable platform-independent solution that is capable of transmitting multiple synchronized high-resolution video streams between single workstations and/or clusters. TeraVision was also designed for high-speed graphics streaming over wide-area networks and SAGE directly exploits its network transport libraries. However, TeraVision has a static application layout on a tiled display it is not suitable for supporting parallel applications and multiple instances of applications since each TeraVision box can stream the pixels of only one application.

From our prior work on the Continuum [2], we have learned that it is crucial for collaborators to have both local control (e.g. on a tablet or laptop) and the casual ability to share their work and see what others are working on (e.g. on a large tile display). For that purpose SAGE's Free Space Manager (i.e. window manager) provides an intuitive interface for moving and resizing visualizations on the tiled display. The interface is also completely transparent giving the user the illusion that they are working on one continuous computer screen, even though the systems performing the visualizations may be several thousand miles apart. The Free Space Manager is akin to a traditional desktop manager in a windowing system, except that it can scale from a single tablet PC screen to a desktop spanning over 100 million pixel displays. Since SAGE was designed with collaboration in mind, it makes sense to give control to multiple users that are not necessarily collocated. To facilitate that requirement, Free Space Manager can accept multiple UI client connections and receive commands from any of them and broadcast the changes to all the other clients. Since commands are issued via simple message passing, various user interfaces can be used for control at the same time. For the basic user interaction a SAGE Graphical User Interface was developed which resembles contemporary desktop environments (see Figure 2). SAGE GUI allows multiple users to connect to multiple SAGE displays and run and shutdown applications and control their windows. As an alternative user interface, a system is currently being developed for controlling applications with a camera-tracked device on the horizontal 5x3 tiled display, the Lambda Table [12].

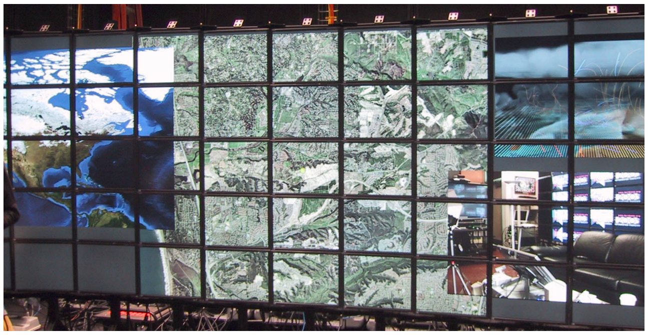


Figure 1. SAGE running on a 55-tile display in San Diego with applications streamed from Amsterdam and Chicago (iGrid 2005).

2 SAGE

The current implementation of SAGE can execute multiple instances of various visualization applications on local or remote rendering clusters and dynamically stream the pixels of the applications to tiled displays so that multiple application windows can be freely moved or resized on the tiled displays. Each application exploits SAGE Application Interface Library (SAIL) to stream the output of application. Most visualization applications can be easily integrated into SAGE framework by adding a few lines of API code for SAIL.

Figure 1 shows the current implementation of SAGE running two high-resolution aerial photography applications (left and center), an HD movie (top right) and a live HD camera feed (bottom right).

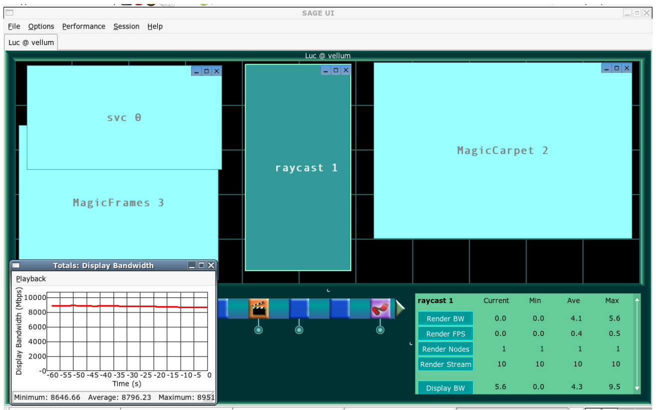
2.1 Motivation

Due to the enhancement of scientific data acquisition techniques, the resolution of scientific image datasets easily exceeds the resolution of single desktop displays. For example, geoscientists working with aerial and satellite imagery deal with 365Kx365K pixels maps and neurobiologists create images of the brain with montages consisting of thousands of pictures from a high-resolution microscope (4Kx4K pixels sensor). SAGE can support hundreds of megapixels of display resolution which is enough for the high-resolution visualizations of such datasets.

SAGE allows scientists to visualize multiple relevant datasets on a big tiled display by executing multiple instances of an application and comparing the datasets to get some meaningful information by freely moving, resizing, and overlapping the visualization windows of the datasets. This feature is very useful for biologists analyzing high-resolution microscope images or geoscientists analyzing multiple core images.

Using wide area network streaming capability of SAGE, scientists can visualize remotely stored datasets on several different distributed clusters and display them on the local tiled display. Large clusters with connectivity to high-speed networks can be used to share data and visualizations using SAGE. As the size of scientific datasets increases rapidly, sharing data or visualizations becomes increasingly important.

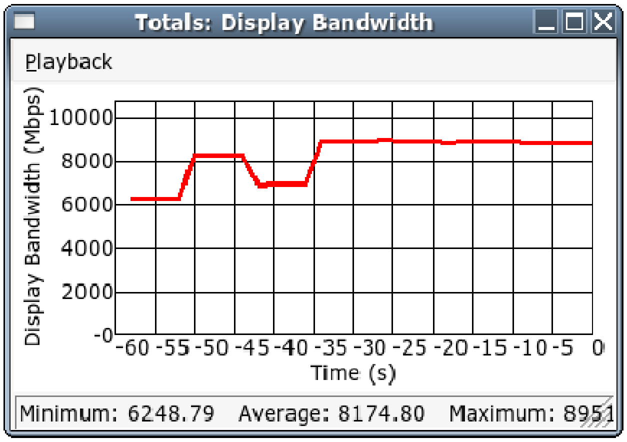
At iGrid 2005 workshop, we succeeded in streaming the output of visualization applications from three distinct remote clusters in KISTI (South Korea), EVL (Chicago), and SARA (Amsterdam) to a big tiled display (11x5 tiles, total 100Mpixel display resolution) in UCSD (San Diego) over 10 Gigabit networks. Total bandwidth used for this experiment reached 9 Gigabits/sec.



Minimum: 8646.66 Average: 8796.23 Maximum: 8951

raycast 1	Current	Min	Ave	Max
Render BW	0.0	0.0	4.1	5.6
Render FPS	0.0	0.0	0.4	0.5
Render Nodes	1	1	1	1
Render Stream	10	10	10	10
Display BW	5.6	0.0	4.3	9.5

Figure 2. SAGE UI screenshot from San Diego with applications streamed from Korea, Amsterdam and Chicago (iGrid 2005)



Minimum: 6248.79 Average: 8174.80 Maximum: 8951

Figure 3. SAGE performance data during streaming from Korea, Amsterdam and Chicago (iGrid 2005)

Since SAGE is well suited for use in collaborative environments, several tools have been incorporated into SAGE GUI to facilitate joint work. Users could, for example, have discussions and meetings in front of a tiled display where each of them is running an instance of the SAGE GUI. When a user wants to show something on the display, they just invoke the application from the UI that supports that type of data, choose a dataset and position it on the display. For basic communication, a chat capability and a list of users currently connected to the display are available. Every user could also be connected to multiple displays at the same time and control application windows on any of them. This could prove to be especially useful when multiple sites are working together. At the end of a meeting, users could save the session and the state of the tiled display so that they can quickly resume their work at a later time.

2.2 Future Work

We are working on extending SAGE to scalably support distance collaboration with multiple endpoints by streaming pixels to all the participating endpoints using either traditional router-based multicasting or photonic multicasting. When multicasting capability is added to SAGE, different endpoints in the collaboration may be seeing the same visualization on different devices with different display characteristics and effectively share the visualization without wasting network bandwidth.

Also, we are developing new streaming protocols and real-time compression to improve the pixel streaming capabilities of SAGE which in turn will make it more reliable and faster over wide area networks. Eventually, we will extend SAGE to stream other graphics data types such as polygons, voxels or progressive meshes so that it can support wider range of applications and utilize networks, rendering and display resources more effectively.

As for user interaction, our goal is to make collaboration natural and intuitive for any user and not just the experienced ones. SAGE allows just that by decoupling the control from the main control module of SAGE, the Free Space Manager. With that decoupling, the user interface can easily be extended to support more specific features, possibly in form of modules or plug-ins. As of now, each application's user interface has to be started separately but by packaging them with the SAGE UI as modules, the process of running applications in the SAGE environment will be even more transparent to the users. To ease the process of making user interfaces for different applications, a template UI that already handles frequently used commands will be provided. That will be possible since Free Space Manager accepts application UI commands and forwards them to the applications themselves. This in turn will allow multiple distributed application UIs to control the same application. Another planned tool useful in group work is the ability to annotate on the display with the possibility of saving the screenshots for later review.

3 SAGE Applications

We briefly describe several applications which have been integrated into the SAGE environment.

3.1 Ultra-high-resolution Image Viewer

JuxtaView [4] is a cluster-based application for viewing ultra-high-resolution images on scalable tiled displays. We ported JuxtaView to the SAGE environment, where each node of the cluster generates a portion of the total image. The pixels are then streamed using SAGE to a high-resolution tiled display. JuxtaView enables a user to interactively roam through potentially terabytes of distributed, spatially referenced image data sets such as those generated from electron microscopes, satellites and aerial photography. Through the use of a large amount of bandwidth, SAGE enables the domain scientist to bridge distributed resources including storage, rendering, and display clusters.

MagicCarpet is another cluster-based ultra-high-resolution image viewer for scalable tiled displays. MagicCarpet reads pre-processed multi-resolution images and shows the appropriate images at different zoom levels. High-resolution images with multiple frames can be read and animated by this software. It uses VTK to visualize image data much faster than JuxtaView.

Both, JuxtaView and MagicCarpet, were used for iGrid demonstrations to show aerial photography of Hurricane Katrina and San Diego, montages of microscope images and Blue Marble - true color global imagery at 1km resolution.

3.2 Volume Visualization Application

Vol-a-Tile [11] is an interactive tool for exploring large volumetric data on scalable tiled displays. Hardware texture mapping and level-of-detail techniques improve interactivity. Since Vol-a-Tile is an OpenGL-based application, we added a pixel 'readback' step at the end of the rendering phase. These pixels are streamed for display using SAGE. This is a minimal modification to the source code, and could be applied to any OpenGL application. This is an easy way to port various open-source applications to SAGE. Given the fill-limited performance characteristics of volume rendering applications (Vol-a-Tile uses OpenGL 3D textures and fragment shaders), the SAGE enabled version of this application runs faster when it is executed on a cluster of machines where each rendering node generates a sub-portion of the final image. SAGE then recombines all the streams and provides the user with a coherent picture, which can be moved and scaled on any portion of the tiled display.

3.3 HD Video Streaming

The National Center for Supercomputing Applications (NCSA) developed an HD Movie Player which reads raw HD images from the disk and plays them back. We ported this software to the SAGE environment and used it for iGrid demonstration of HD video streaming (1920x1080) from Amsterdam or Chicago to a large tiled display in San Diego. An HD stereo movie was also streamed from NCSA to San Diego using SAGE.

Scalable Visualization Consumer (SVC) receives MPEG2 data through IEEE1394, files on disk or network interface and decompresses it for streaming to a tiled display using SAGE. To be shown on a tiled display, MPEG2 data has to be decompressed, split as sub-images and streamed to the appropriate display nodes. The reason for decompression is that compressed data cannot be split up for streaming to separate nodes. We used this software for iGrid demonstration to stream HD video captured by an IEEE 1394 HD camera. It shows 30fps frame rate and uses 650Mbps of network bandwidth.

3.4 OpenGL Wrapper

Numerous scientific applications and visualization packages are using the OpenGL API (OpenDX, VTK, or Paraview for instance). The success of Chromium and WireGL shows a need to support native OpenGL applications in binary mode (without source code modification). We developed an OpenGL wrapper library along the scheme used by WireGL: using a shared library mechanism, we only capture the calls to the 'glSwapBuffer' function and add a pixel readback step. The captured pixels are then streamed to SAGE. This is an efficient and extremely easy mode to port native OpenGL applications to SAGE. The performance is sufficient to run a 1280x1024 resolution application at an interactive frame rate. The new generation of PCI-express graphics card will even increase the performance several times.

3.5 VNC Viewer

Finally, we developed a Virtual Network Computer (VNC) protocol client that enables users to bring desktop content to the SAGE environment. Our VNC application is a regular VNC viewer program modified to serve as a proxy between a VNC server (of any size and pixel depth) and SAGE. Once the pixels are retrieved from the VNC server, the same pixels are given to the SAGE API for immediate display. The functionality is critical in a collaborative environment where each scientist with his/her laptop needs to share information (web browser, presentations…). SAGE supports any number of simultaneous VNC applications, making use of the large real estate offered by high-resolution tiled displays.

4 Conclusion

SAGE can support collaborative scientific visualizations at extremely high display resolution. Decoupling of rendering and display addresses heterogeneity and scalability concerns. We have successfully demonstrated the functionality of SAGE at iGrid 2005 workshop where multiple remote sites were simultaneously streaming pixels of various applications to a single 55-tile display. The data streamed was also heterogeneous and included very high resolution 2D satellite images, rendered 3D volume data, live HD camera feed, HD movie frames and live electronic microscope feeds. By further extending the SAGE environment we are hoping to bridge the physical barriers between teams of collaborators and allow them to work together as if they were collocated. The challenges lie in the availability of reliable high speed networks, further simplifying ease of use and efficient data and visualization sharing.

5 PUBLICATIONS

Byungil Jeong, Luc Renambot, Rajvikram Singh, Andrew Johnson, Jason Leigh, *High-Performance Scalable Graphics Architecture for High-Resolution Display,* Technical publication

Renambot, L., Rao, A., Singh, R., Jeong, B., Krishnaprasad, Naveen, Vishwanath, V., Chandrasekhar, V., Schwarz, N., Spale, A., Zhang, C., Goldman, G., Leigh, J., Johnson, A., *SAGE: the Scalable Adaptive Graphics Environment,* Proceedings of WACE 2004, Nice, France, 09/23/2004 - 09/24/2004

REFERENCES

[1] Larry L. Smarr, Andrew A. Chien, Tom DeFanti, Jason Leigh, Philip M. Papadopoulos "The OptIPuter" Communications of the ACM, Volume 46, Issue 11, November 2003, pp. 58-67.

[2] Park, K., Renambot, L., Leigh, J. and Johnson, A., "The Impact of Display-rich Environments for Enhancing Task Parallelism and Group Awareness in Advanced Collaborative Environments", In Workshop on Advanced Collaboration Environments, June 22-24, 2003, Seattle, WA.

[3] Rajvikram Singh, Byungil Jeong, Luc Renambot, Andrew Johnson and Jason Leigh "TeraVision: a Distributed, Scalable, High Resolution Graphics Streaming System" , in the proceedings of IEEE Cluster 2004, San Diego, September 20-23, 2004.

[4] Naveen K. Krishnaprasad, Venkatram Vishwanath, Shalini Venkataraman, Arun G. Rao, Luc Renambot, Jason Leigh, Andrew E. Johnson, and Brian Davis "JuxtaView – a Tool for Interactive Visualization of Large Imagery on Scalable Tiled Displays" , in the proceedings of IEEE Cluster 2004, San Diego, September 20-23, 2004.

[5] W. Blanke, C. Bajaj, D. Fussell, and X. Zhang, "The Metabuffer: a Scalable Multiresolution Multidisplay 3-D Graphics System using Commodity Rendering Engines." Tr2000-16, University of Texas at Austin, February 2000.

[6] J. Leigh, L. Renambot, T.A. DeFanti, M.D. Brown, E. He, N.K. Krishnaprasad, J. Meerasa, A. Nayak, K. Park, R. Singh, S. Venkataraman, C. Zhang, D. Livingston, M. McLaughlin, "An Experimental OptIPuter Architecture for Data-Intensive Collaborative Visualization", 3rd Workshop on Advanced Collaborative Environments, Seattle, WA, June 2003

[7] G. Humphreys, I. Buck, M. Eldridge, and P. Hanrahan, "Distributed rendering for scalable displays.", IEEE Super-computing 2000.

[8] G. Humphreys, M. Houston, Y. Ng, R. Frank, S. Ahern, P. Kirchner, and J. T. Klosowski, "Chromium: A Stream-Processing Framework for Interactive Rendering on Clusters", Proceedings of SIGGRAPH 2002.

[9] Kenneth A. Perrine, Donald R. Jones, William R. Wiley "Parallel Graphics and Interactivity with the Scaleable Graphics Engine" Proceedings of the 2001 ACM/IEEE conference on Supercomputing.

[10] D. Germans, H.J.W. Spoelder, L. Renambot, H. E. Bal, "VIRPI: A High-Level Toolkit for Interactive Scientific Visualization in Virtual Reality", Proc. Immersive Projection Technology/Eurographics Virtual Environments Workshop, Stuttgart, May 2001.

[11] Nicholas Schwarz, Shalini Venkataraman, Luc Renambot, Naveen Krishnaprasad, Venkatram Vishwanath, Jason Leigh, Andrew Johnson, Graham Kent, Atul Nayak, "Vol-a-Tile - a Tool for Interactive Exploration of Large Volumetric Data on Scalable Tiled Displays", IEEE Visualization 2004, Poster session, Austin, TX, October 2004.

[12] Krumbholz, C., Leigh, J., Johnson, A., Renambot, L. and Kooima, R., Lambda Table: High Resolution Tiled Display Table for Interacting with Large Visualizations. in Proceedings of 5th Workshop on Advanced Collaborative Environments, Redmond, Washington, 2005.