TACC Technical Report TR-09-04

A Practical Guide to Large Tiled Displays

Paul A. Navrátil, Brandt Westing, Gregory P. Johnson
Ashwini Athalye, Jose Carreno, Freddy Rojas

Texas Advanced Computing Center
The University of Texas at Austin

`[ pnav | bwesting | gregj | ashwini | jcarreno | rfreddy ] @ tacc.utexas.edu`

July 15, 2009

This work was funded in part by generous donations from Dell, Microsoft, and the Office of the Vice-President for Research of the University of Texas at Austin.

Figure 1: Stallion, the 307 Mpixel display in the TACC ACES Visualization Laboratory at the University of Texas at Austin. The display consists of 75 display tiles driven by 23 render nodes, each with two GPUs. The rectangles overlaying the image show our hybrid mapping of display tiles to render nodes. Tiles in the display center (green) are mapped two tiles per node, with each tile driven by a separate GPU. This central "hotspot" offers improved rendering performance for GPU-intensive applications. Tiles on the display edges (orange) are mapped four tiles per node, with two tiles sharing a GPU. One node drives three tiles (blue) to accommodate the odd number of tiles. This hybrid tile-to-node mapping allowed us to expand the display by 30 tiles, increasing the resolution 67%, without adding render nodes or network capacity.

Abstract

The drive for greater detail in scientific computing and digital photography is creating demand for ultra-resolution images and visualizations. Such images are best viewed on large displays with enough resolution to show "big picture" relationships concurrently with fine-grained details. Historically, large scale displays have been rare due to the high costs of equipment, space, and maintenance. However, modern tiled displays of commodity LCD monitors offer large aggregate image resolution and can be constructed and maintained at low cost. We present a discussion of the factors to consider in constructing tiled LCD displays and an evaluation of current approaches used to drive them based on our experience constructing displays ranging from 36 Mpixels to 307 Mpixels. We wish to capture current practices to inform the design and construction of future displays at current and larger scales.

1 Introduction

Scientists, like photographers, seek the greatest possible detail in their images. Yet, our ability to generate large images and large image sets has outstripped our ability to view them at full resolution. Gigapixel scientific images are becoming commonplace, from the sub-kilometer resolution satellite images of NASA's Earth Observatory [30] to nanometer-resolution electron micrographs [41] used in three-dimensional cell reconstruction. Also, analysis of ultrascale supercomputing datasets increasingly requires high-resolution imagery to capture fine detail. Further, scientists using image-intensive processes, such as image alignment in biological microscopy, may track features across tens or hundreds of related images, the combined sizes of which can be much larger than conventional displays.

Historically, projection-based systems have been used for large, high-resolution displays, because of both their seamless image and a lack of viable alternatives. However, projection systems are expensive, both to purchase and to maintain. Recent technology advances have reduced the both purchase cost and the cost-per-pixel, but maintenance costs remain high, both for upkeep (projector bulbs, display alignment), and for the lab space needed to accommodate the screens, the projectors, and the necessary throw distance between them.

Recently, tiles of commodity LCD monitors have been used to construct displays of over two hundred megapixels [12]. Tiled LCD displays offer low purchase and maintenance costs, but often the software used to drive these displays requires a custom API [27, 7, 14, 18], a constraint that complicates application implementation and prohibits running third-party applications for which the source code is unavailable.

In this paper, we describe our experience constructing tiled LCD displays of various resolutions using only freely-available, open-source software. We show that while custom display software can provide high render rates for special applications, they are not required to drive these displays with good performance, which dramatically reduces development and maintenance costs. Further, we provide hardware and software recommendations to guide the construction of future displays both at current and larger resolutions.

2 The Case for Large High-Resolution Displays

Not everyone sees value in large high resolution displays. Such displays are sometimes labeled as "only good for demos" and, less charitably, as "fleecing rooms" for big donors. However, recent studies in human-computer interaction demonstrate that these large displays offer improved usability and performance for analyzing high-resolution imagery.

2.1 Improved Human Interaction

The human-computer interaction community has documented the benefits of high-resolution displays, and several studies have targeted tiled LCD displays in particular for increasing user perception [43], productivity [2, 6, 8, 9, 35], and satisfaction [36]. These benefits appear to scale with increased display size. Further, a large high-resolution display permits physical navigation of the image, where viewers walk about the display to view portions of the image. On geospatial visualization tasks, physical navigation was shown to provide superior task performance than virtual navigation, scrolling and zooming the image through a software interface on a single screen [5]. Of the components of Dourish's "embodied interaction" concept of interface design [16, 17], there is evidence that physical navigation of the image is a primary component of user productivity and satisfaction in large-scale visualization tasks [3, 4].

2.2 High Resolution Imagery

Scientific equipment contains increasingly high-resolution sensors that produce high-resolution images. Multiple images are often analyzed together, either by composition into a single enormous image or by comparison of a related image set. In astronomy, composite images from space probes range from 100 Mpixel panoramics of Mars [28] to over three gigapixels for full-Earth coverage at sub-kilometer resolution [30] to five gigapixel infrared scans of the inner Milky Way [29]. In biology, electron micrographs at nanometer resolution can be over one gigapixel [41]. Further, three-dimensional reconstruction of electron tomographs rely on proper alignment of the individual images [32], and the alignment process often requires manual identification of features across images. For both example domains, a large display would aid detecting features and relationships, either within a composite image or across a large image set.

2.3 Scientific Computing

Scientific computing exists in a feedback loop: the increasing capacity and capability of supercomputers drive increased resolution and precision in scientific simulations, which in turn require larger and more capable systems to effectively display and analyze the simulation results. Science ranging from universal dark matter N-body simulations [38, 34, 23] to high-resolution hurricane storm surge modeling [11] to turbulent fluid flow models [15] create ultra-resolution results. To match the result resolution, researchers should create ultra-resolution images of their data, and such images are better-analyzed on a large-format, high resolution display [4].

2.4 Projection or Tiled-LCD?

Though the largest projection display is less than one-fourth the resolution of the largest tiled-LCD displays [24], projection systems remain popular because of their seamless images. With seamlessness, however, comes higher purchase costs, larger space requirements, and maintenance costs for bulbs and projector alignment. The highest-resolution projectors today use an 8.8 Mpixel LCD [22, 37], a resolution slightly higher than two 30" LCD tiles. Assuming retail costs of $1000 per LCD tile and $100,000 per 4K projector, a projection display of the same resolution as the largest tiled LCD displays would cost nearly 47 times as much, without considering space and maintenance costs. A single 4K projector bulb costs thousands of dollars, the cost of several LCD display tiles. In addition, projectors must be kept in alignment, either physically or with an automated calibration system [20]. Even with the advent of thirty-five megapixel projectors [10], the price-per-pixel cost still favors tiled LCD technology.

3 Tiled-LCD Display Hardware

The highest-resolution displays currently use either tiled-LCDs or projectors, and all known displays over 100 Mpixels use tiled-LCDs [24]. Other high-resolution display technologies exist [31], but are not currently used for large displays. The remainder of the paper will concern tiled-LCD displays.

Tiled-LCD displays have been built entirely from commodity parts, at resolutions of ten to over two hundred megapixels [12, 21, 25]. Below, we highlight key aspects of the hardware used in Stallion, the 307 Mpixel display at the Texas Advanced Computing Center (TACC), that extend the commodity hardware trend. Specific hardware details can be found at the TACC website [40].

Stallion consists of seventy-five LCD monitors mapped to twenty-three rendering nodes, each with two GPUs, and a head node that acts as the user console. The machine contains a total of 100 processing cores, 108 GB aggregate RAM and 46 render-node GPUs with 36 GB aggregate graphics RAM. The render nodes of Stallion are Dell XPS "gaming boxes" marketed to home enthusiasts, rather than workstations or rack-mounted machines, and each node contains two NVIDIA GeForce gaming cards, rather than industrial-class Quadro cards. Home-user hardware can provide adequate performance for lower total cost, depending on the intended use of the display. We describe these considerations in Section 5.

Of the seventy-five display tiles, fifty-eight share a GPU with another tile and seventeen have a dedicated GPU. These seventeen tiles are centrally located in the display, creating a “hot-spot” with increased rendering performance. Table 1 quantifies the rendering performance of the hot-spot compared to other regions of the display.

4 Display Environment Evaluation

We identify three categories of display environments: windowing environments; OpenGL substitutes that reimplement the OpenGL API; and custom parallel libraries that implement a new rendering library interface. We will discuss the qualities of each category below through the performance of representative software on Stallion. The features of each category are summarized in Table 3.

4.1 Windowing Environments

Windowing environments for tiled displays include Distributed Multihead X (DMX) [13] and the Scalable Adaptive Graphics Environment (SAGE) [33]. DMX acts as an X Windows proxy to multiple X servers running on a tiled display, whereas SAGE hosts a separate windowing environment running within an X server on the cluster’s head node. While DMX can support most X-enabled software, the heavy communication load of the X protocol limits DMX’s scalability beyond sixteen nodes. In addition, to account for the display mullions, each tile must have a separate X display, since Xinerama with DMX does not permit compensation for the mullion gap. Thus, DMX is also effectively limited to sixteen displays.

SAGE scales beyond both the node and tile limits of DMX by implementing its own windowing and communication protocols, and it compensates for mullions. Though SAGE does not support a full X environment, it provides native image and video support and an API for “plug-ins” for third-party applications. In addition, it uses dynamic pixel routing to allow runtime movement and scaling of imagery and video across the tiled display. This pixel routing is bandwidth intensive: the image source node must stream pixels over the network to the

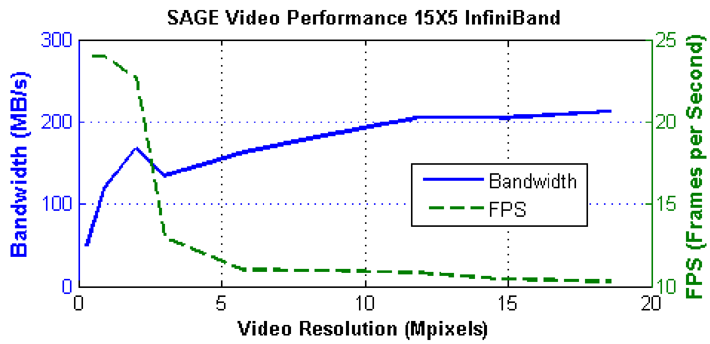

Figure 2: This plot shows that video streaming in SAGE is dependent on network bandwidth. Playback for resolutions higher than 1080p (1920x1080) is no longer real-time. The dip in bandwidth consumed just past 1080p resolution is due to the sudden drop in frame rate, causing less total data to be streamed.

render nodes where the image will be displayed. Thus, available network bandwidth from the source node is the bottleneck for SAGE performance. As Figure 2 shows, SAGE experiences a sharp performance drop past high-definition (1080p) resolution. Test videos were natively encoded at 24 fps, and each video test was placed over the entire 15×5 display to eliminate effects from display location and node communication. For a single source node on our SDR InfiniBand fabric, we reached bandwidth saturation at ~ 230MB/s. To increase SAGE performance for a single video stream above 1080p (1920x1080) resolution, we must use faster interconnect hardware. However, to increase SAGE performance for several video streams with an aggregate resolution above 1080p, the bandwidth load can be distributed across the cluster by sourcing videos from separate render nodes.

4.2 OpenGL Substitutes

Chromium [19] is a widely known OpenGL implementation for parallel and cluster rendering, though other implementations have been made [26, 42]. Chromium intercepts application-level OpenGL calls and distributes them across a rendering cluster. By sending rendering information (geometry, textures, transformations) rather than raw pixels, Chromium often consumes less bandwidth at high image resolution than a pixel streaming environment like SAGE. Because it uses the OpenGL API, Chromium allows unmodified OpenGL applications to be run directly on a tiled display.

Chromium is ill-suited for some applications. Because Chromium streams OpenGL calls, geometry- and texture-intensive applications can saturate network bandwidth. In addition, Chromium implements only up to the OpenGL 1.5 standard, though any missing OpenGL functionality can be implemented by the user. Finally, Chromium is subject to any resolution limitations built into the render nodes' native OpenGL stack, so maximum image resolution may be smaller than the resolution of the display.

4.3 Custom Parallel Libraries

Researchers have built custom parallel rendering libraries to overcome Chromium's limitations and to support specific application functionality. These include IceT [27], VR Juggler [7], CGLX [14], and Equalizer [18]. While custom library implementations can yield significantly better rendering performance over Chromium [39], their API calls must be implemented in source, thereby limiting their use with third-party applications. Since CGLX is in use on many of the largest tiled-LCD displays [12], we chose to explore its performance on Stallion.

In CGLX, an instance of the application is opened on each of the rendering nodes, and the head node communicates with the render nodes to synchronize the display, thereby reducing bandwidth requirements compared to pixel or OpenGL streaming. CGLX reimplements certain OpenGL methods, such as glFrustum, to perform correctly and efficiently in a distributed parallel context. Our CGLX evaluation used its native OpenSceneGraph viewer. Our tests used 20K vertex and 840K vertex geometry files from 3Drender.com's Lighting Challenge[1].

The benchmark results in Table 1 show that CGLX performance is determined by the slowest render node, which is due to the synchronization enforced by the head node. Further, the framerate doubles when the rendering window is displayed only on tiles with a dedicated GPU, demonstrating the increased performance of the "hot-spot" described in Section 3. The bandwidth used was approximately the same for all cases and did not exceed 160kB/s.

Table 1: This table shows the increase in rendering performance by having fewer displays per node. In addition, it shows that CGLX scales extremely well due to its distributed architecture. All render nodes have two GPUs and most nodes drive four screens. There is a centered 5×3 tile "hot-spot" where each GPU drives only one tile. (*) The 15×5 configuration includes both two tile per GPU nodes and the one tile per GPU hot-spot nodes. Performance is governed by the two tile per GPU nodes, with a slight performance penalty from the increased display area.

Tile Layout	**Render Nodes**	**Tiles per GPU**	**FPS @ 20K Verts**	**FPS @ 840K Verts**
5×3	8	1	585	115
5×4	5	2	250	68
15×5	23	2*	248	64

5 Recommendations

5.1 Hardware Selection

5.1.1 Framing and Display Layout

Stallion, along with other tiled-LCD displays [12, 21], uses modular metal framing from 8020. The cost of this framing comes to approximately $100 – $150 per tile, with decreasing marginal cost as total tiles increase. The frame specification can be designed in any 3D modelling tool that can make real-world distance measurements, such as Google SketchUp, and the frame can be reconfigured or expanded easily.

The relative quantities of render nodes, GPUs and display tiles dictate the layout by which tiles should be connected to rendering nodes. There are four cases, which we present in order of increasing complexity:

- A render node contains a single GPU connected to one tile. Applications displayed on the tile receive all CPU and GPU resources.
- A node contains multiple GPUs, each connected to one display tile. Applications displayed on any connected tile must share CPU resources and system memory.
- A node contains one GPU that drives multiple displays. Applications displayed on any connected tile must share GPU resources and graphics memory.

Table 2: This table shows the power usage of Stallion's Dell 3007WFP-HC LCD displays and Dell XPS 720 render nodes. Brightness governs the operating draw for the LCD displays; CPU and GPU load governs the operating draw for the render nodes.

	Draw with Power Off	**Observed Operating Draw**	**Rated Maximum Draw**
Dell 3007WFP-HC	0.05 A	0.5 — 1.15 A	1.6 A
Dell XPS 720	0.36 A	1.36 — 2.2 A	8.33 A

- A node contains multiple GPUs and each GPU drives multiple display tiles. Applications displayed on any connected tile must share both CPU and GPU resources.

When multiple tiles are assigned to a single render node (cases 2–4, above), these tiles should be both contiguous and regularly shaped, either in lines or rectangles, to minimize the number of applications or images sent to each render node. We have found that though "L"- and "S"-shaped layouts can be specified, they are not well-supported by graphics drivers.

5.1.2 Power Efficiency

When determining the power budget for a tiled display, prudent design suggests using max amperage draw for all components, plus a percentage for overage (which can also accommodate future expansion). We recommend this for displays in newly-constructed facilities where power requirements can be specified in advance. In our experience, actual amperage draw is significantly below the manufacturer-quoted maximum, which may allow existing circuitry to be used for a new display in repurposed space or to expand an existing display. Table 2 presents the measured power draw for Stallion hardware.

5.1.3 Interconnect

We have found that the system interconnect, especially connectivity from the cluster head node to the render nodes, plays a crucial role in overall system performance. At a minimum, the cluster should be interconnected with 1 Gb Ethernet (GbE), and we recommend maintaining a redundant 1GbE network as a fallback for a higher-bandwidth interconnect.

We recommend investing in a high-bandwidth interconnect, such as InfiniBand, especially if the display will be used for video streaming. We caution that, in contrast to Ethernet switches, multiple InfiniBand switches cannot easily be linked together while maintaining peak bandwidth rates. If the render node cluster may be expanded during the system lifetime, we recommend using a blade-based InfiniBand switch so that expansion ports can be added without impacting overall fabric bandwidth.

5.1.4 Render Nodes

We use Dell XPS 720 "gaming boxes" in Stallion and Dell Precision 690 workstation nodes in Colt, both with NVIDIA GeForce 8800 GTX GPUs. We use workstation form factors because rack-mounted nodes with GPUs were not yet available when the each machine was designed. Rack-mounted render nodes may better fit space, aesthetic and HVAC constraints, depending on the location of the display.

We have found the GeForce-class GPUs sufficient to drive both Stallion and Colt, though video lag among tiles can be seen at very high frame rates. Quadro-class GPUs are capable of hardware enforced frame-locking, though additional daughter cards are needed for each render node to enable it. Lag may seriously affect high framerate immersive applications, but we have found the actual impact of lag on display usefulness to be negligible, in part because the display mullions reduce the noticeable effects of lag between tiles.

5.1.5 Display Tiles

In addition to the per-pixel cost savings of using commodity LCD displays, the display mullions help reduce assembly and maintenance costs by masking small misalignments between displays that would otherwise be visually objectionable. Informally, we have found that users interpret the mullions as "window panes," and with this idea, they see "past" them as if looking out a window. We posit that this phenomenon exists only for mullions of a certain size. If the mullions are sufficiently thin, or removed entirely, a viewer may ignore the tile divisions and interpret the display as a single solid image. If this occurs, any tile misalignment would be visually objectionable. Further, the display would need periodic realignment due to natural shifting of the frame and building.

We advise purchasing extra displays at the time of the original order to ensure a supply of replacement tiles of the same form factor and manufacture lot. Small variances in the color temperature of LCD back-lights from different

Table 3: The table distinguishes the capabilities of display environments. Windowing Environments include SAGE and DMX. OpenGL Substitute refers to parallel rendering libraries like Chromium that implement the OpenGL API directly. Custom Parallel Library refers to parallel rendering libraries that use their own API, such as CGLX, VR Juggler and IceT. (1) DMX supports most X-enabled applications, while SAGE supports a limited range of applications via SAGE plug-in.

	Windowing Environment	**OpenGL Substitute**	**Custom Parallel Lib**
Application Location	head / cluster	head node	cluster nodes
Distributed Apps	✓		✓
Distributed Rendering		✓	✓
Distributed Display	✓	✓	✓
Must Modify App Code	*1		✓
Example Use Case	Image & Video Streaming	Parallel Render 3rd Party Apps	Parallel Render Custom Apps

manufacture lots can cause objectionable variance among tiles, though GPU driver settings can provide corrective adjustments. Having replacement displays on hand simplifies maintenance, and these displays can be used on other systems until needed.

5.2 Software Selection

Many tiled-LCD displays, including Stallion, use a Unix-based operating system such as Ubuntu, Red Hat, or Mac OS X [12, 21, 25]. We use a Long Term Service (LTS) release of the Ubuntu Linux distribution on Stallion, since LTS distribution support is guaranteed for two years and updated packages are provided every six months.

Display environments should be chosen according to anticipated uses of the tiled display. We summarize the available display environment options in Table 3. For distributed parallel applications that require an MPI stack, we have found that OpenMPI provides a simple and stable MPI environment, especially over gigabit Ethernet. For an InfiniBand-connected cluster, we recommend either OpenMPI or MVAPICH MPI stacks.

6 Future Work and Conclusion

In this paper, we catalog our experience constructing large tiled-LCD displays at resolutions ranging from 36 Mpixels to 307 Mpixels. We demonstrate that large tiled-LCD displays can be built using commodity parts and run using open-source software, which help make them the lowest price-per-pixel technology for high-resolution displays. While custom-built libraries provide the best rendering performance on these displays, windowing environments and parallel OpenGL implementations can provide adequate performance for video and third-party applications. Yet, these options could be improved: an efficient distributed parallel implementation for image and video streaming would mitigate the need for a high cost, high-bandwidth interconnect; and a parallel implementation of the current OpenGL standard would increase the types of software immediately usable on these displays. Also, progress on rack-mountable rendering nodes opens the possibility for mobile high-resolution tiled displays that could be deployed with remote research teams to analyze high-resolution data at the point of generation. We hope that the high-resolution display community continues to embrace open-source, freely available software so that access to these displays may continue to grow.

References

[1] 3DRender.com. http://www.3drender.com/challenges/, 2009.

[2] Robert Ball and Chris North. An Analysis of User Behavior on High-Resolution Tiled Displays. In *Tenth IFIP International Conference on Human-Computer Interaction*, pages 350–364, September 2005.

[3] Robert Ball and Chris North. Realizing Embodied Interaction for Visual Analytics through Large Displays. *Computers & Graphics*, 31(3):380–400, June 2007.

[4] Robert Ball and Chris North. The Effects of Peripheral Vision and Physical Navigation in Large Scale Visualization. In *Graphics Interface*, pages 9–16, June 2008.

[5] Robert Ball, Chris North, and Doug A. Bowman. Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays. In *ACM Conference on Human Factors in Computer Systems*, pages 191–200, April 2007.

[6] Robert Ball, Michael Varghese, Bill Carstensen, E. Dana Cox, Chris Fierer, Matthew Peterson, and Chris North. Evaluating the Benefits of Tiled Displays for Navigating Maps. In *International Conference on Human-Computer Interaction*, pages 66–71, November 2005.

[7] Aron Bierbaum, Patrick Hartling, Pedro Morillo, and Carolina Cruz-Neira. *Computational Science and Its Applications*, chapter Implementing Immersive Clustering with VR Juggler, pages 1119–1128. Springer Berlin / Heidelberg, 2005.

[8] Mary Czerwinski, Greg Smith, Tim Regan, Brian Meyers, George Robertson, and Gary Starkweather. Toward Characterizing the Productivity Benefits of Very Large Displays. In *Eighth IFIP International Conference on Human-Computer Interaction*, 2003.

[9] Mary Czerwinski, Desney S. Tan, and George G. Robertson. Women Take a Wider View. In *ACM Conference on Human Factors in Computing Systems*, pages 195–201, 2002.

[10] Chris Davies. JVC create 35-megapixel 8k x 4k projector LCD. *SlashGear.com*, May 2 2008.

[11] C. Dawson, J. Westerink, E. Kubatko, J. Proft, and C. Mirabito. Hurricane Storm Surge Simulation on Petascale Computers. TeraGrid 2008 presentation, June 2008.

[12] Thomas A. DeFanti, Jason Leigh, Luc Renambot, Byungil Jeong, Alan Verlo, Lance Long, Maxine Brown, Daniel J. Sandin, Venkatram Vishwanath, Qian Liu, Mason J. Katz, Philip Papadopoulos, Joseph P. Keefe, Gregory R. Hidley, Gregory L. Dawe, Ian Kaufman, Bryan Glogowski, Kai-Uwe Doerr, Rajvikram Singh, Javier Girado, Jurgen P. Schulze, Fako Kuester, and Larry Smarr. The OptiPortal, a Scalable Visualization, Storage, and Computing Interface Device for the OptiPuter. *Future Generation Computer Systems*, 25(2):114–123, February 2009.

[13] Distributed Multihead X Project. http://dmx.sourceforge.net/, 2004.

[14] Kai-Uwe Doerr and Falko Kuester. http://vis.ucsd.edu/ cglx/, 2009.

[15] D.A. Donzis, P.K. Yeung, and K.R. Sreenivasan. Energy Dissipation Rate and Enstrophy in Isotropic Turbulence: Resolution Effects and Scaling in Direct Numerical Simulations. *Physics of Fluids*, 20, 2008.

[16] Paul Dourish. Seeking a Foundation for Context-Aware Computing. *Human-Computer Interaction*, 16(2):229–241, December 2001.

[17] Paul Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. Cambridge, MA: MIT Press, 2001.

[18] Stefan Eilemann, Maxim Makhinya, and Renato Pajarola. Equalizer: A Scalable Parallel Rendering Framework. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):436–452, May 2009.

[19] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D. Kirchner, and James T. Klosowski. Chromium: a Stream-Processing Framework for Interactive Rendering on Clusters. In *ACM SIGGRAPH*, pages 693–702, 2002.

[20] C. Jaynes, W.B. Seales, K. Calvert, Z. Fei, and J. Griffioen. The Metaverse: a Networked Colleciton of Inexpensive, Self-Configuring, Immersive Environments. In *Proceedings of the EuroGraphics Workshop on Virtual Environments (EGVE)*, pages 115–124, 2003.

[21] Gregory P. Johnson, Paul A. Navrátil, David Gignac, Karl W. Schulz, and Tommy Minyard. The Colt Visualization Cluster. Technical Report TR-07-03, Texas Advanced Computing Center, July 26 2007.

[22] JVC. http://pro.jvc.com/prof/attributes/features.jsp?model_id=mdl101793, 2009.

[23] Juhan Kim, Changbom Park, J. Richard Gott III, and John Dubinski. The Horizon Run N-Body Simulation: Baryon Acoustic Oscillations and Topology of Large Scale Structure of the Universe. *The Astrophysical Journal (submitted)*, 2009.

[24] KVM sans V. http://kvmsansv.com/multi-megapixel_displays.html, 2009.

[25] Laboratory for Information Visualization and Evaluation, Virginia Tech. http://infovis.cs.vt.edu/gigapixel/index.html, 2009.

[26] T. Mitra and T.-C. Chiueh. Implementation and Evaluation of the Parallel Mesa Library. In *International Conference on Parallel and Distributed Systems*, pages 84–91, 1998.

[27] Kenneth Moreland, Brian Wylie, and Constantine Pavlakos. Sort-Last Parallel Rendering for Viewing Extremely Large Data Sets on Tile Displays. In *IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, pages 85–92, 2001.

[28] NASA. http://photojournal.jpl.nasa.gov/catalog/pia04182, 2005.

[29] NASA. http://www.spitzer.caltech.edu/media/releases/ssc2008-11/ssc2008-11a.shtml, 2008.

[30] NASA. http://earthobservatory.nasa.gov/features/bluemarble, 2009.

[31] Tao Ni, Greg S. Schmidt, Oliver G. Staadt, Mark A. Livingston, Robert Ball, and Richard May. A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. In *IEEE Virtual Reality 2006*, pages 223–234, March 2006.

[32] Sébastien Phan and Albert Lawrence. Tomography of Large Format Electron Microscope Tilt Series: Image Alginment and Volume Reconstruction. In *2008 Congress on Image and Signal Processing*, pages 176–182, 2008.

[33] L. Renambot, A. Rao, R. Singh, B. Jeong, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, A. Spale, C. Zhang, G. Goldman, J. Leigh, and A. Johnson. SAGE: the scalable graphics architecture for high resolution displays. In *Proceedings of WACE 2004*, 2004.

[34] Paul R. Shapiro, Ilian T. Iliev, Garrelt Mellema, Ue-Li Pen, and Hugh Merz. The Theory and Simulation of the 21-cm Background from the Epoch of Reionization. In *The Evolution of Galaxies through the Neutral Hydrogen Window (AIP Conf. Proc.)*, 2008.

[35] Lauren Shupp, Christopher Andrews, Margaret Kurdziolek, Beth Yost, and Chris North. Shaping the Display of the Future: The Effects of Display Size and Curvature on User Performance and Insights. *Human-Computer Interaction (to appear)*, 2008.

[36] Lauren Shupp, Robert Ball, Beth Yost, John Booker, and Chris North. Evaluation of Viewport Size and Curvature of Large, High-Resolution Displays. In *Graphics Interface (GI) 2006*, pages 123–130, June 2006.

[37] Sony Electronics Inc. http://pro.sony.com/bbsc/ssr/cat-projectors/cat-ultrahires/, 2009.

[38] Volker Springel, Simon D.M. White, Adrian Jenkins, Corlos S. Frenk, Naoki Yoshida, Liang Gao, Julio Navarro, Robert Thacker, Darren Croton, John Helly, John A. Peacock, Shaun Cole, Peter Thomas, Hugh Couchman, August Evrard, Joerg Colberg, and Frazer Pearce. Simulations of the Formation, Evolution and Clustering of Galaxies and Quasars. *Nature*, 435:629–636, 2005.

[39] Oliver G. Staadt, Justin Walker, Christof Nuber, and Bernd Hamann. A Survey and Performance Analysis of Software Platforms for Interactive

Cluster-Based Multi-Screen Rendering. In *IPT/EGVE*, pages 261–270, 2003.

[40] Texas Advanced Computing Center. http://www.tacc.utexas.edu/resources/vislab/, 2009.

[41] The Electron Microscopy Outreach Program. http://em-outreach.ucsd.edu/, 2008.

[42] Brian Wylie, Constantine Pavlakos, Vasily Lewis, and Ken Moreland. Scalable Rendering on PC Clusters. *IEEE Computer Graphics and Applications*, 21(4):62–70, July/August 2001.

[43] Beth Yost, Yonca Haciahmetoglu, and Chris North. Beyond Visual Acuity: The Perceptual Scalability of Information Visualizations for Large Displays. In *ACM Conference on Human Factors in Computer Systems (CHI)*, pages 101–110, April 2007.