Chapter 4

VideoBlaster: A Distributed, Low-Network Bandwidth Method for Multimedia Playback on Tiled Display Systems

4.1 Introduction

Video and multimedia content is a well understood method for communication of ideas. Large tiled display walls present a powerful interface for presenting audio/visual information, but the general techniques used for these systems require substantial network resources and provide challenges for scalability. Often, videos are decoded by a single machine and then streamed in sections to the tiled display system. In general, a gigabit Ethernet interface is the minimum requirement for these systems to play high-definition content.

The approach presented in this chapter removes these high bandwidth constraints by distributing the decoding of the video over the entire system. The compressed form of the content can be pulled from network file systems, multicast network streams, or local harddrives. Each of these approaches requires substantially less network overhead than the existing solutions.

Figure 4.1: The system displaying BBC's *HD In Full Bloom* video of time lapse photography [bbc]. Each of the 70 monitors shows a temporal offset of a single video frame.

This distributed approach also allows for video tiles to change their display characteristics on the fly. Users have the ability move the video data between the displays, rescale and skew movie data interactively. Users can also toggle displays to show the same information on each screen as well as temporal offsets as shown in figure 4.1.

Because data is not sent over the network, the systems is frame size agnostic, and is limited by the speed of decoding on the nodes themselves. This allows for multimedia content to be played on tiled display environments using very low bandwidth networks, allowing for cluster-display systems to be connected via wireless networks or separated by vast distances.

4.2 Background

For tiled display walls, there a few different techniques generally used in order to present a coherent visual across all display tiles. One way of controlling tiled-display walls is to create a virtual unified display as shown by DMX, Chromium [HEB$^+$01], OpenSG [VBRR02] as described in Chapter 1 . This method is advantegous as it works with most applications and may not require code to be recompiled. This approach can generally not work with textures, shaders, has poor synchronization mechanisms and is not useful approach for video playback.

Another approach is to send pixel content to display nodes as outlined in Scalable Adaptive Graphics Environment (SAGE) [JJR+05] [RJJ+06] [JRJ+06] as described in Chapter 1. In this approach one system renders content into a buffer which is mapped to the tiled display environment. This buffer is segmented so that each node in the tiled display system only recieves information which is will currently need to display. The pixel information for each display is then streamed out via network.

The advantage of this approach is that data and video playback applications need to exist only on the head node to fill the pixel buffer which needs to be streamed. The drawback of this system is that it requires a very low latency, high bandwidth network. Larger video sizes or frame rates require increased network costs. Furthermore, the readback and splitting operation also have performance costs associated with them. Finally, because a single node is tasked with reading, decoding, and rendering into the buffer, the maximum rate at which the video can be played back is limited by the processing power of this computer.

Most compression schemes do not work for partial frame decoding due to motion vectors in progressive frame decoding. This is because motion vectors translate information already decoded to different parts of screen space. Consequently, motion vectors may bring information in from outside a region of interest, which would need to already be decoded.

In the MPEG2 standard, motion vectors are confined to macro blocks. This allows for the possibly of partial frame decoding as shown in [Che02] and [Che03]. Unfortunately the MPEG2 standard only allows for video sizes up to 1,920x1,152 [STA], meaning that encoding videos of greater resolution can not be done using common encoders.

Furthermore, this approach requires a second level of nodes in-between the head node and render nodes in order to negotiate macro-block forwarding. These routing nodes must receive and resend information, including header data which incurs an additional 20% bandwidth cost. While this method is useful for ultra-large resolution video data, it requires additional hardware, is limited in its playback ability, and still requires a decent network in-order to operate.

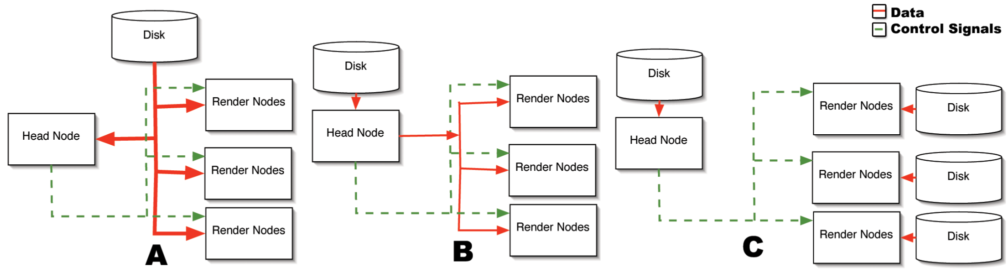

Figure 4.2: Data flow for three different distribution paradigms. Figure A shows the data being fetched for all nodes via a network file system. Figure B shows a multi-cast distribution through the head node resulting in less network overhead. Figure C shows the resulting substantial network reduction from pre-distributing the video.

The distributed application approach as shown in VRJuggler [BJH$^+$08], and CGLX [DK10] allows for a tiled display environment to also act as a distributed computer as mentioned in Chapter 1. This distributed approach was selected in order to allow for greater performance, a higher level of the data control and the ability to playback data with minimal network overhead.

4.3 Data Distribution

Using this distributed system, there are three different ways in which video content can be delivered to the tiled display environment as shown in Figure 4.2. The simplest way is through a network mounted file system. This mount point could come from a remote file server or simply through the head nodes hard drive. Since each node will be pulling the same information, the is no worry of disk thrashing. The one downside of this approach is that each node will pull this information through the network.

A second approach is to stream the data via UDP multi-cast. In this way, the data is sent only once to a multicast stream, and each node in the tiled display system simply joins this stream. The advantage of this approach is that the data is sent only once through the network, so adding additional nodes does not result in extra network bandwidth. The problem with this approach is that UDP is not reliable transport mechanism, so data may be lost in transit. This problem is even more apparent when streaming from remote locations.

The third approach is to put the data locally on each system. In this setup, all of the needed information is already at each of the nodes so the only data which needs to be relayed is synchronization information. While this method requires pre-distribution of the data, it allows for a minimal amount of network resources.

4.4 Decoding Paradigms

Moving the data to the correct place is just the first step in displaying video information on tiled display systems. After the data is transfered, it then must be decoded in order to extract the video frames to display. Different decoding paradigms present different benefits and challenges when working on a tiled display system.

The simplest method may be to decode the information in one location, and then stream the raw data to tiled display system as shown in the streaming playback paradigms above. In this scenario, render nodes receive information and immediately display it on the screen. Because there is no buffering involved, as long as the transport of the information can be guaranteed, this method does not require extra synchronization controls.

The disadvantage of this approach is that all screen information must be sent through the network. This, in turn, means that as frame sizes and frame rates increase, so must the network bandwidth requirements. This is also true for streaming multiple videos to a single display environment. As shown in the results section below these requirements are quite high even for a 1080p video.

Another method would be to synchronize the decoding of each frame across the entire system. In this scenario, each node reads the same frame from disk and decodes a single frame at the same time. The system waits until all systems have read the same frame before moving to the next one. This paradigm is advantageous for maintaining consistency on the display environment. In this way the head node and render nodes all are at the same state for buffering and playback. If the decoding falls behind on any given node, it will slow the entire system. As a result, this set-up is only as fast as the slowest node.

For improved performance, one can also have each node decode asynchronously. In this paradigm, each node decodes at the fastest rate it can. In this way, the decoding pipeline is not stalled and each node has the ability to decode as fast as possible. The only synchronization mechanism sent is the either a frame swap token or a time token from the head node. The challenge of this method is that there is no consistency between render nodes and the head node or between any of the render nodes themselves.

In order to ensure that all nodes have frames ready to play, extra signals must be passed from the display nodes back to the head node in order to stall the video playback if buffering is necessary. In this system, message passing is accomplished by encoding these status messages inside of the handshaking mechanism in the acknowledgment system. Also, as opposed to the head node simply sending a signal to switch to the next frame, it can also send the playback time to the nodes. In this way, each node can negotiate which frames to display based on the contents of its buffer. The advantage of this method is nodes which are less powerful can simply skip decoding frames but maintain synchronization with the rest of the display environment.

4.5 Display System

While the decoding of video and audio frames can be assigned to working threads, the data transfer onto the graphics card can only be accomplished with a valid OpenGL context. A flexible and performance optimized visualization framework is needed that provides a distributed and scalable OpenGL context on a tiled display and allows for asynchroneous data upload and processing on the GPU while maintaining a synchronized and coherent visualization grid. As the underlying visualization system, the CGLX (Cross-Platform Cluster Graphic Library)[DK10] framework was selected, which provides these characteristic as build-in features.

The minimal network footprint of CGLX was used to manage large-scale visualization systems and use its extended API which provides flexibility for application specific performance tuning.

In order to determine the appropriate position to display video frames on the tiled display environment, CGLX is queried for information about the arrangement and position of each monitor in the visualization grid. CGLX determines the correct projection and transformation matrices in order to create a seamless display.

CGLX also provide mechanisms in order to synchronize user events, such as mouse and keyboard I/O. On top of this, CGLX provides the ability to create user events which are propagated reliably and efficiently throughout the tiled display environment. Through these events, users can interactively change the layout of the video on the tiled display system. Videos can be moved, rescaled and distorted instantaneously without need for reconfiguration as seen in streaming setups.

To minimize the amount of syncronization needed, two different types of CGLX events were registered, local events and global events. Local events are seen in the event queue for a single node, but are not propogated to the tiled display system and are not syncrhonized. These are generated when the frame has been decoded and is ready to be uploaded to the graphics card.

Global events can only be generated by the head node and are propogated to the render nodes. The head node monitors the time of video playback through either sound card if audio was being played back or though the system clock. When enough time has passed for a new frame to be shown, a global event, which passed the playback time, is sent to the nodes. In testing, the CGLX monitoring system showed delays of less than two milliseconds for these messages to be propagated.

Another optimization is to upload the YUV textures as opposed to RGB textures. By uploading each of the YUV image planes as three separate textures, the amount of data which needs to be sent to the graphics card is reduced by 50%. The YUV to RGB color conversation is done in a fragment program inside of the drawing loop. This resulted in a savings of approximately 0.5 ms for 1080p HD videos and more than 2 ms for 4k videos per frame.

4.6 Applications

Because the entire video file is fully available to the render nodes, other non-standard playback tecniques can be utilized. These techniques are invaluable in the field of digital cinematography, as directors, editors and cinematographers can see a myriad of design decisions without loss of resolution.

One very interesting playback technique is to show the same video on each monitor, but with a different temporal offset between displays as shown in Figure 4.1. This effect is amplified when viewed through a large number of displays. In this way the viewer has a three-dimensional understanding of the video content using time as the third dimension. Features such as camera movements and cuts produce visual patterns which the human mind can easily catch.

Figure 4.3: The system displaying HD video with color variations between each display.

Another interesting technique is to show the same video frame on all displays, but to use fragment shaders with different parameters between nodes. This technique allows for side-by-side comparison of potential design decisions, such as color tinting, contrasting, saturating, as shown in Figure 4.3. Because these operations are performed on the GPU, these comparisons can been seen in realtime during playback in solidarity without need for re-rendering video content.

Because the requirements of video playback described in this Chapter are so low, other network setups could be ultilized. For instance tiled display environments to display video content could easily be created on top of wireless communication systems. Furthermore video can be displayed and controlled between remote sites using standard internet thoroughfares.

4.7 Performance Measurements

Measuring performance of video playback is a difficult endeavor. In many ways, video playback represents a binary conclusion, either the video is able to play effectively or not. While framesize and frequency are two components of this performance measurement, they are both dependent on many other factors.

Because most video codecs use lossy compression, the amount of data needed to decode any particular frame is dependent on not only the content of the frame itself, but also the content of the preceding frames. The codec parameters are tunable and the results of the encoding process can result in videos which require vastly different amount of resources in order to play correctly.

For testing, the Pixar *Cars* movie trailer [car] was selected which can easily be found online. The video is 1,920x800 pixels in resolution encoded with the h264 video codec. The video contained 5.1 surround sound with a framerate of 23.98 fps. The trailer lasted two minutes and six seconds with filesize of 158 MB and an average bitrate of 9,983 kb/s.

The method described was tested using tiled display system which consisted of 70 30" monitors. The system was driven by 18 Dell XPS710 computers each using a pair of nVIDIA Quadro FX5600s to drive 4 displays per node, with one node driving only two displays. The system, including a head node of identical hardware, was connected by a gigabit ethernet network.

4.8 Results

As shown in Table 4.1, the pixel streaming approach as used by SAGE required a gigabit network interface in-order to operate. As nodes were not available to recreate the macro-block decoding as described in [Che02], an estimate was derived for the bandwidth using a 1-4-(14x5) configuration with the network data shown in the Chapter. As demonstrated by the green bars in Figure 4.4, the approaches undertaken in this Chapter required substantially less network resources. When the data was pre-distributed, network traffic was decreased by almost 100,000 times compared to the pixel streaming method.

Table 4.1: Average Bandwidth for video playback of the *Cars* Trailer.

Method	SAGE	Macro Block	CIFS/NFS	Multicast	Pre Distributed
Bandwidth	950 Mbps	726 Mbps	180 Mbps	10 Mbps	0.010 Mbps

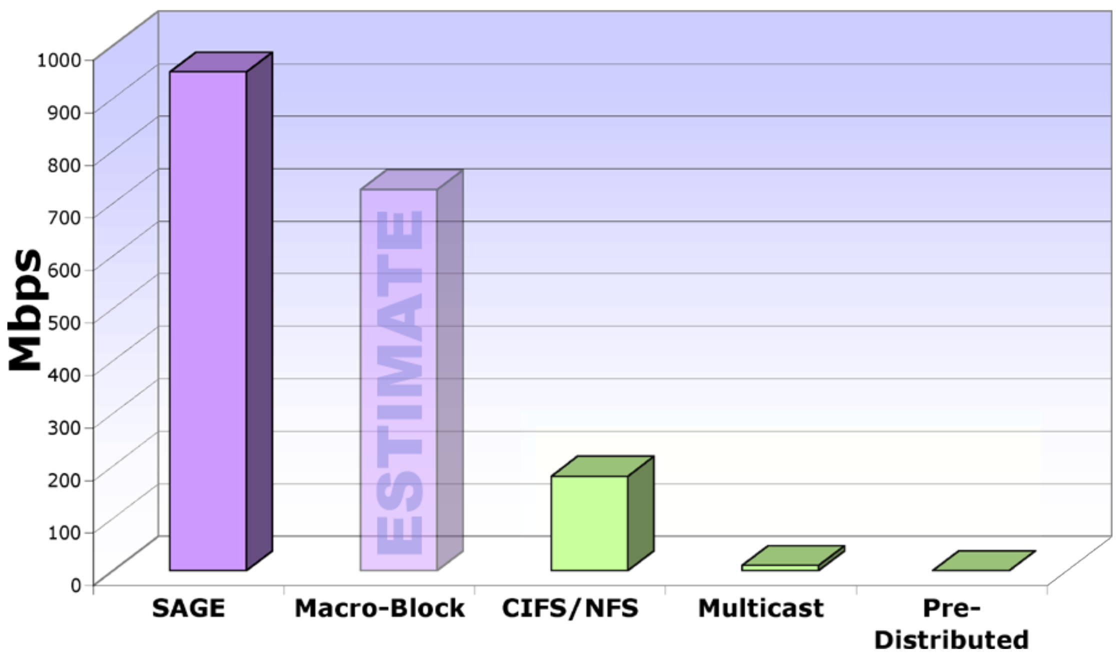

Figure 4.4: Network bandwidth in Mbps for different methods playing the *Cars* trailer as shown in Table 4.1

4.9 Scalability

Since the method described in this Chapter uses different paradigms, it is important to notice how the system scales.

When using a streaming approach (SAGE), the network bandwidth is proportionally equal to

$$NetworkBandwidth \approx Framesize \times FPS \tag{4.1}$$

Conversely, in the method described in this Chapter,

$$NetworkBandwidth \approx NumberOfNodes \times FPS \tag{4.2}$$

for when video content is pre-distributed. In general, the number of pixels in each video frame will be orders of magnitude greater than the number of nodes in the tiled display system.

For SAGE the latency is seen in two parts.

$$L_{streamingnode} \approx L_{read} + L_{decode} + L_{split} + L_{send} \tag{4.3}$$

$$L_{rendernode} \approx L_{receive} + L_{upload} \tag{4.4}$$

The inherrient latency involved in decoding video is equal to

$$L_{videoblaster} \approx L_{read} + L_{decode} + L_{upload} \tag{4.5}$$

removing the time needed to split the frame, send and receive this information. This gives the described method a slight advantage for playing back real time content. Furthermore, on a multi-core or multi-processor system, with this approach the read, decode, and upload opperations can be pipelined to improve throughput without signficantly effecting latency. With a streaming approach like SAGE, pinelining of the decode and upload is impossible since the opperations occur on seperate machines.

4.10 Conclusion

This chapter demonstrates a method for playing multimedia content on a tiled display environment with minimal network overhead. This system is interactively configurable, as videos can be moved, resized and scaled on the fly. Each display tile has the ability to show the same video frame as well as temporal offsets between frames. The low network bandwidth requirements allows for a greater level of scalability.

4.11 Acknowledgments

©2009 IEEE. This chapter is a reprint, with permission, from "VideoBlaster: a distributed, low-network bandwidth method for multimedia playback on tiled display systems" as it appears in the proceeding of IEEE International Symposium on Multimedia 2009. Ponto, K., Wypych, T., Doerr, K., Yamaoka, S., Kimball, J., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Bibliography

[Ake93] Kurt Akeley. Reality engine graphics. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 109–116, New York, NY, USA, 1993. ACM.

[bbc] http://www.apple.com/quicktime/guide/hd/bbc-cfb.html.

[BJH+08] Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, and Carolina Cruz-Neira. Vr juggler: a virtual platform for virtual reality application development. In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses*, pages 1–8, New York, NY, USA, 2008. ACM.

[BN05] Robert Ball and Chris North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05 extended abstracts on Human factors in computing systems*, pages 1196–1199, New York, NY, USA, 2005. ACM.

[BR98] Uwe Behrens and Ralf Ratering. Adding shadows to a texture-based volume renderer. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 39–46, New York, NY, USA, 1998. ACM.

[BVG05] Stefan Bruckner, Ivan Viola, and M. Eduard Gröller. Volumeshop: interactive direct volume illustration. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 60, New York, NY, USA, 2005. ACM.

[BXH+09] Leonardo Bonanni, Xiao Xiao, Matthew Hockenberry, Praveen Subramani, Hiroshi Ishii, Maurizio Seracini, and Jurgen Schulze. Wetpaint: scraping through multi-layered images. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 571–574, New York, NY, USA, 2009. ACM.

[car] http://www.apple.com/trailers/disney/cars/.

[CB04] Xiang Cao and Ravin Balakrishnan. Visionwand: interaction techniques for large displays using a passive wand tracked in 3d. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 729–729, New York, NY, USA, 2004. ACM.

[CCF94] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *VVS '94: Proceedings of the 1994 symposium on Volume visualization*, pages 91–98, New York, NY, USA, 1994. ACM.

[CEM01] F. Capani, M.H. Ellisman, and M.E. Martone. Filamentous actin is concentrated in specific subpopulations of neuronal and glial structures in rat central nervous system. *Brain Research*, 923(1-2):1–11, 2001.

[Che02] Han Chen. A parallel ultra-high resolution mpeg-2 video decoder for pc cluster based tiled display system. to appear. In *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS), IEEE CS*, page 30. Press, 2002.

[Che03] Han Chen. *Scalable and Ultra-High Resolution MPEG Video Delivery on Tiled Displays*. PhD thesis, Princeton University, 2003.

[CHS04] Ian Creighton and Chris Ho-Stuart. A sense of touch in online sculpting. In *GRAPHITE '04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 118–122, New York, NY, USA, 2004. ACM.

[CI05] Alvaro Cassinelli and Masatoshi Ishikawa. Khronos projector. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Emerging technologies*, page 10, New York, NY, USA, 2005. ACM.

[Cor09] Carlos D. Correa. Visualizing what lies inside. *SIGGRAPH Comput. Graph.*, 43(2):1–6, 2009.

[CS02] Hui Chen and Hanqiu Sun. Real-time haptic sculpting in virtual volume space. In *VRST '02: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 81–88, New York, NY, USA, 2002. ACM.

[CSC06] Carlos Correa, Deborah Silver, and Min Chen. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1069–1076, 2006.

[CSM02] E.F. Churchill, D.N. Snowdon, and A.J. Munro. Collaborative virtual environments: digital places and spaces for interaction. *Educational Technology & Society*, 5(4), 2002.

[CT09] Andrew A. Chien and Nut Taesombut. Integrated resource management for lambda-grids: The distributed virtual computer (dvc). *Future Generation Computer Systems*, 25(2):147 – 152, 2009.

[DC02] James Davis and Xing Chen. Lumipoint: multi-user laser-based interaction on large tiled displays. *Displays*, 23(5):205 – 211, 2002.

[DDS+09] Thomas A. DeFanti, Gregory Dawe, Daniel J. Sandin, Jurgen P. Schulze, Peter Otto, Javier Girado, Falko Kuester, Larry Smarr, and Ramesh Rao. The starcave, a third-generation cave and virtual reality optiportal. *Future Generation Computer Systems*, 25(2):169 – 178, 2009.

[DK10] Kai-Uwe Doerr and Falko Kuester. CGLX: A Scalable, High-performance Visualization Framework for Networked Display Environments. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints), 2010.

[DL01] P. Dietz and D. Leigh. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226. ACM New York, NY, USA, 2001.

[DLR+09] 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, Falko 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, 2009.

[EKCB03] Jr. Easton, R.L., K.T. Knox, and W.A. Christens-Barry. Multispectral imaging of the archimedes palimpsest. *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd*, pages 111–116, Oct. 2003.

[EKE01] Klaus Engel, Martin Kraus, and Thomas Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *HWWS '01: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 9–16, New York, NY, USA, 2001. ACM.

[Elv92] T. Todd Elvins. A survey of algorithms for volume visualization. *SIGGRAPH Comput. Graph.*, 26(3):194–201, 1992.

[FAJ07] G. Flint, C. Aves, and MT Jones. The gigapxl project. http://www.gigapxl.org, 2007.

[GH91] Tinsley A. Galyean and John F. Hughes. Sculpting: an interactive volumetric modeling technique. *SIGGRAPH Comput. Graph.*, 25(4):267–274, 1991.

[Gra72] R. L. Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132 – 133, 1972.

[GRC+07] J.F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz. The expanding digital universe: A forecast of worldwide information growth through 2010. *IDC white paper*, 2007.

[GSW01] François Guimbretière, Maureen Stone, and Terry Winograd. Fluid interaction with high-resolution wall-size displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 21–30, New York, NY, USA, 2001. ACM.

[HA08] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7(1):49–62, 2008.

[Han05] J.Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM New York, NY, USA, 2005.

[Har90] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335 – 346, 1990.

[HEB+01] Greg Humphreys, Matthew Eldridge, Ian Buck, Gordan Stoll, Matthew Everett, and Pat Hanrahan. Wiregl: a scalable graphics system for clusters. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 129–140, New York, NY, USA, 2001. ACM.

[Her08] L. Herr. Creation and Distribution of 4 K Content. *Television Goes Digital*, page 99, 2008.

[HKSB06] M. Hadwiger, A. Kratz, C. Sigg, and K. Bühler. Gpu-accelerated deep shadow maps for direct volume rendering. In *Graphics Hardware 2006: Eurographics Symposium Proceedings, Vienna, Austria, September 3-4, 2006*, pages 49–52. Eurographics Association, 2006.

[HLSR08] Markus Hadwiger, Patric Ljung, Christof Rezk Salama, and Timo Ropinski. Advanced illumination techniques for gpu volume raycasting. In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses*, pages 1–166, New York, NY, USA, 2008. ACM.

[HYB02] T. Hansen, P. Yalamanchili, and H.W. Braun. Wireless measurement and analysis on HPWREN. In *Proceedings of Passive and Active Measurement Workshop, Fort Collins, Co*, pages 222–229, 2002.

[Ini06] Digital Cinema Initiatives. Standard evaluation material (stem), 2006.

[JC06] G. Johansson and H. Carr. Accelerating marching cubes with graphics hardware. In *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research*, page 39. ACM New York, NY, USA, 2006.

[JJR+05] B. Jeong, R. Jagodic, L. Renambot, R. Singh, A. Johnson, and J. Leigh. Scalable graphics architecture for high-resolution displays. In *IEEE Information Visualization Workshop*, 2005.

[JRJ+06] Byungil Jeong, Luc Renambot, Ratko Jagodic, Rajvikram Singh, Julieta Aguilera, Andrew Johnson, and Jason Leigh. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In *SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, page 108, New York, NY, USA, 2006. ACM.

[KCC08] D. Kim, K. Cha, and S.I. Chae. A high-performance openvg accelerator with dual-scanline filling rendering. *IEEE Transactions on Consumer Electronics*, 54(3):1303–1311, 2008.

[KKH01] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of the conference on Visualization'01*, pages 255–262. IEEE Computer Society Washington, DC, USA, 2001.

[KSR$^+$06] Matthias Koenig, Wolf Spindler, Jan Rexilius, Julien Jomier, Florian Link, and Heinz-Otto Peitgen. Embedding vtk and itk into a visual programming and rapid prototyping platform. *Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display*, 6141(1):61412O, 2006.

[KUDC07] Johannes Kopf, Matt Uyttendaele, Oliver Deussen, and Michael F. Cohen. Capturing and viewing gigapixel images. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 93, New York, NY, USA, 2007. ACM.

[KVV$^+$04] NK Krishnaprasad, V. Vishwanath, S. Venkataraman, AG Rao, L. Renambot, J. Leigh, AE Johnson, and B. Davis. JuxtaView-a tool for interactive visualization of large imagery on scalable tiled displays. In *Cluster Computing, 2004 IEEE International Conference on*, pages 411–420, 2004.

[LBS85] SK Lee, W. Buxton, and KC Smith. A multi-touch three dimensional touch-sensitive tablet. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 21–25. ACM New York, NY, USA, 1985.

[LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169. ACM New York, NY, USA, 1987.

[Lee84] S. Lee. A fast multiple-touch-sensitive input device. *Master's thesis, University of Toronto*, 1984.

[Leh97] Roy S. Lehrle. Forensics, fakes, and failures: Pyrolysis is one part in the overall armoury. *Journal of Analytical and Applied Pyrolysis*, 40-41:3 – 19, 1997. PYROLYSIS '96.

[LM04] Eric B. Lum and Kwan-Liu Ma. Lighting transfer functions using gradient aligned sampling. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 289–296, Washington, DC, USA, 2004. IEEE Computer Society.

[Mar91] K. Martinez. High resolution digital imaging of paintings: The vasari project. *Microcomputers for Information Management*, 8(4):277–83, 1991.

[MC05] Kirk Martinez and John Cupitt. Vips - a highly tuned image processing software architecture. In *ICIP (2)*, pages 574–577, 2005.

[MCSP02] K. Martinez, J. Cupitt, D. Saunders, and R. Pillay. Ten years of art imaging research. *Proceedings of the IEEE*, 90(1):28–41, 2002.

[MDH+03] A. MacEachren, X. Dai, F. Hardisty, D. Guo, and G. Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *IEEE Symposium on Information Visualization, 2003 (INFOVIS 2003); 19–21 Oct. 2003; Seattle, Washington*, pages 31–38. Citeseer, 2003.

[Mit97] J.L. Mitchell. *MPEG video compression standard*. Kluwer Academic Publishers, 1997.

[Mor98] H. Moravec. When will computer hardware match the human brain. *Journal of Evolution and Technology*, 1:1–14, 1998.

[MRB05] Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 43–52, New York, NY, USA, 2005. ACM.

[MTB03] Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 53, Washington, DC, USA, 2003. IEEE Computer Society.

[NR02] S. Navrud and R.C. Ready. *Valuing cultural heritage*. Elgar, 2002.

[PDMDRP08] A. Pelagotti, A. Del Mastio, A. De Rosa, and A. Piva. Multispectral imaging of paintings. *Signal Processing Magazine, IEEE*, 25(4):27–36, July 2008.

[PKS+08] Peter Peltonen, Esko Kurvinen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, John Evans, Antti Oulasvirta, and Petri Saarikko. It's mine, don't touch!: interactions at a large multi-touch display in a city centre. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1285–1294, New York, NY, USA, 2008. ACM.

[Ple08] L. Plesea. The design, implementation and operation of the JPL OnEarth WMS server. In *Geospatial Services and Applications for the Internet*, pages 93–109. Springer US, 2008.

[PSH97] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. 1997.

[PWFO01] KL PERNG, WT WANG, M. FLANAGAN, and M. OUHYOUNG. A Real-time 3D Virtual Sculpting Tool Based on Modified Marching Cubes. In *Int Conf Artif Real Telexistence*, volume 11, pages 64–72, 2001.

[RBJW01] Meredith Ringel, Henry Berg, Yuhui Jin, and Terry Winograd. Barehands: implement-free interaction with a wall-mounted display. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 367–368, New York, NY, USA, 2001. ACM.

[Rek98] Jun Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 344–351, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[RJJ+06] L. Renambot, B. Jeong, R. Jagodic, A. Johnson, J. Leigh, and J. Aguilera. Collaborative visualization using high-resolution tiled displays. In *ACM CHI Workshop on Information Visualization Interaction Techniques for Collaboration Across Multiple Displays*, 2006.

[RJL05] L. Renambot, A. Johnson, and J. Leigh. Lambdavision: Building a 100 megapixel display. In *NSF CISE/CNS Infrastructure Experience Workshop, Champaign, IL*, 2005.

[RP00] M. Riesenhuber and T. Poggio. Models of object recognition. *Nature Neuroscience*, 3:1199–1204, 2000.

[Ryd] Thomas Rydell. Virtual autopsy table. https://www.tii.se/projects/autopsy.

[SBD+] J. Schöning, P. Brandl, F. Daiber, F. Echtler, O. Hilliges, J. Hook, M. Löchtefeld, N. Motamedi, L. Muller, P. Olivier, et al. Multitouch surfaces: A technical guide.

[SBdL09] Larry Smarr, Maxine Brown, and Cees de Laat. Special section: Optiplanet – the optiputer global collaboratory. *Future Generation Computer Systems*, 25(2):109 – 113, 2009.

[SC93] D. Saunders and J. Cupitt. Image processing at the national gallery: The vasari project. 1993.

[SGHB07] J.D. Smith, TC Graham, D. Holman, and J. Borchers. Low-cost malleable surfaces with multi-touch pressure sensitivity. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP'07.*

Second Annual IEEE International Workshop on, pages 205–208, 2007.

[SGM03] Stacey D. Scott, Karen D. Grant, and Regan L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *ECSCW'03: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 159–178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.

[SHP+96] Rajeev Sharma, Thomas S. Huang, Vladimir I. Pavlovi'c, Yunxin Zhao, Zion Lo, Stephen Chu, Klaus Schulten, Andrew Dalke, Jim Phillips, Michael Zeller, and William Humphrey. Speech/gesture interface to a visual computing environment for molecular biologists. In *IEEE Computer Graphics and Applications*, pages 30–35, 1996.

[SLJM08] D. Svistula, J. Leigh, A. Johnson, and P. Morin. MagicCarpet: a high-resolution image viewer for tiled displays, 2008.

[SPS48] C. Shannon, N. Petigara, and S. Seshasai. The Mathematical Theory of Communication . *Communication, Bell System Technical Journal*, 1948.

[Sre08] M. Sreenivasan. Microsoft silverlight. 2008.

[STA] C. STANDARD. THE MPEG VIDEO COMPRESSION STANDARD.

[SVFR04] Chia Shen, Frédéric D. Vernier, Clifton Forlines, and Meredith Ringel. Diamondspin: an extensible toolkit for around-the-table interaction. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 167–174, New York, NY, USA, 2004. ACM.

[SVS+05] R. Stockli, E. Vermote, N. Saleous, R. Simmon, and D. Herring. he blue marble next generation – a true color earth dataset including seasonal dynamics from modis. *Published by the NASA Earth Observatory*, 2005.

[SW06] Bram Stolk and Paul Wielinga. Building a 100 mpixel graphics device for the optiputer. *Future Generation Computer Systems*, 22(8):972 – 975, 2006.

[SYK+05] H. Shimamoto, T. Yamashita, N. Koga, K. Mitani, M. Sugawara, F. Okano, M. Matsuoka, J. Shimura, I. Yamamoto, T. Tsukamoto, et al. An Ultrahigh-Definition Color Video Camera With 1.25-inch

Optics and 8k x 4k Pixels. *SMPTE Motion Imaging Journal*, pages 3–11, 2005.

[SYS+06] D. Shirai, T. Yamaguchi, T. Shimizu, T. Murooka, and T. Fujii. 4k shd real-time video streaming system with jpeg 2000 parallel codec. In *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pages 1855–1858, Dec. 2006.

[TC05] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* National Visualization and Analytics Ctr, 2005.

[TFM96] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381(6582):520–522, 1996.

[Tuf91] E.R. Tufte. Envisioning information. *Optometry and Vision Science*, 68(4):322, 1991.

[TWC+06] Nut Taesombut, Xinran (Ryan) Wu, Andrew A. Chien, Atul Nayak, Bridget Smith, Debi Kilb, Thomas Im, Dane Samilo, Graham Kent, and John Orcutt. Collaborative data visualization for earth sciences with the optiputer. *Future Generation Computer Systems*, 22(8):955 – 963, 2006.

[VBRR02] G. Voß, J. Behr, D. Reiners, and M. Roth. A multi-thread safe foundation for scene graphs and its extension to clusters. In *EGPGV '02: Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization*, pages 33–37, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[VL03] H.R. Varian and P. Lyman. How much information. *University of California at Berkeley, School of Information Management & Systems (SIMS)*, 2003.

[VOT] http://www.nationalgeographic.com/field/projects/valley-khans-project.html.

[WAB+05] G. Wallace, O.J. Anshus, P. Bi, H. Chen, Y. Chen, D. Clark, P. Cook, A. Finkelstein, T. Funkhouser, Anoop Gupta, M. Hibbs, K. Li, Z. Liu, Rudrajit Samanta, Rahul Sukthankar, and O. Troyanskaya. Tools and applications for large-scale display walls. *Computer Graphics and Applications, IEEE*, 25(4):24–33, 2005.

[WE98] R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. In *Proceedings of SIGGRAPH*, volume 98, pages 169–178, 1998.

[WEH01] W. Westerman, J. Elias, and A. Hedge. Multi-touch: A new tactile 2-d gesture interface for human-computer interaction. In *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting*, volume 1, pages 632–636, Minneapolis/St. Paul, MN, 2001.

[Wil83a] Lance Williams. Pyramidal parametrics. In *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, New York, NY, USA, 1983. ACM.

[Wil83b] Lance Williams. Pyramidal parametrics. In *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, New York, NY, USA, 1983. ACM.

[Wil04] A.D. Wilson. Touchlight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76. ACM New York, NY, USA, 2004.

[WK95] Sidney W. Wang and Arie E. Kaufman. Volume sculpting. In *I3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 151–ff., New York, NY, USA, 1995. ACM.

[wms] http://www.opengeospatial.org/standards/wms.

[Zha09] Jian-Feng Zhang. Gpu-based direct volume rendering with advanced illumination and deep attenuation shadows. *Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics '09. 11th IEEE International Conference on*, pages 536 –539, 2009.