DOE/ER/25387

Princeton Report of Corridor One Project

Project Title: Corridor One: An Integrated Distance Visualization Environment for SSI and ASCI Applications

(July 15, 1999 – July 14, 2000)

PI: Rick Stevens at Argonne National Lab

Princeton Co-PIs: Kai Li, Adam Finkelstein, and Thomas Funkhouser

Department of Computer Science
Princeton University
Princeton, NJ 08544

Email: {li, af, funk}@cs.princeton.edu

1 Summary

The Corridor One project is a three-year integrated research project that combines the forces of six leading-edge laboratory and university groups working in the area of visualization, distributed computing and high-performance networking (**Argonne National Lab, Lawrence Berkeley National Lab, Los Alamos National Lab, University of Illinois, University of Utah, and Princeton University**) to develop and to deploy the most advanced integrated distance visualization environment. Because of the budget reduction at DOE, this project was funded for only one year. This document reports the accomplishments of the research group at Princeton University during the funded one year period from June 1999 to June 2000.

The tasks of Princeton team in this project are: work with other groups to design, implement and evaluate comminication protocols to support remote visualization and collaboration on the next generation internet, collaborate with other groups to develop specialized windowing protocols for large-format scalable displays, and integrate both protocol implementations with parallel rendering system and remote visualization software. At the end of the project, we plan to demonstrate these systems over the networking infrastructure among the groups in this proposal.

During the funding period (June 1999 and June 2000), Princeton team successfully accomplished the proposed tasks. This report reviews the tasks planned, presents our accomplishments and experimental results, and lists related publications.

2 Proposed Activities in Year One

The proposed tasks of Princeton University team are:

- Developing a large-format display environment,
- Developing a prototype parallel rendering system for the large-format display environment, and
- Studying the tradeoffs of several remote visualization alternatives.

The proposal proposed to continue such efforts in year two and year three.

DOE Patent Clearance Granted
MPDvorscak Apr 29, 2004
Mark P Dvorscak Date
(630) 252-2393
E-mail mark.dvorscak@ch.doe.gov
Office of Intellectual Property Law
DOE Chicago Operations Office

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

3 Accomplishments

3.1 Large Format Visualization Testbeds

We successfully built both prototype systems with a set of software tools. We have built the first scalable display wall system driven by a PC cluster. This prototyping effort has motivated the community to see the potential of using a PC cluster and commodity projector array to construct an inexpensive, large-format visualization environment.

The display wall was built with 8 Proxima LCD polysilicon projectors in a 2×4 layout on an 8'×18' rear projection screen (black screen made by Jenmar Visual Systems). The maximum resolution of the display system is 4,096 × 1,536 (without any overlapping), or about 6 million pixels. The display wall was driven by a PC cluster (donated by Intel). Figure 1 shows such a display wall system.

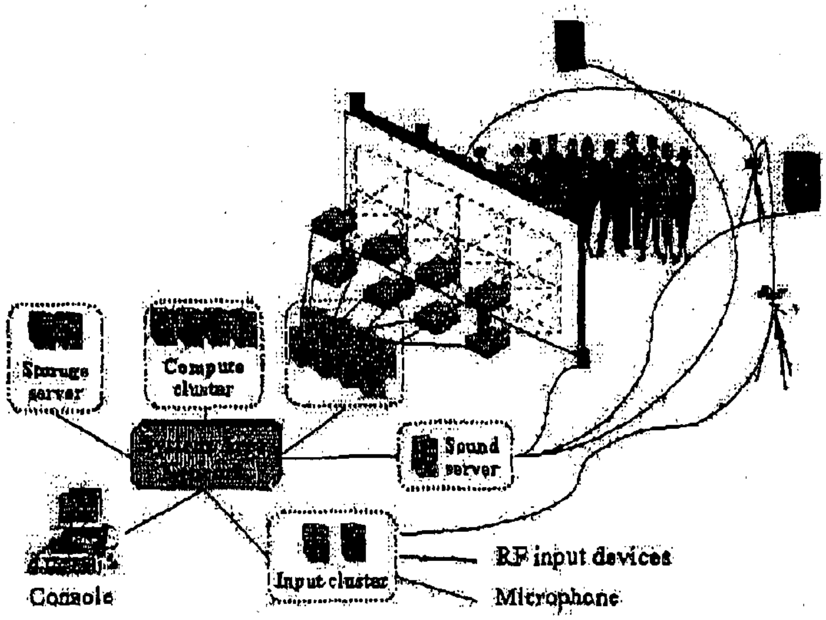

Figure 1: First Princeton Scalable Display Wall Prototype System.

The second scalable display wall prototype was built at the end of year 2 (October 2000), at the end of the funding period. It is a 24-projector array driven by a 24-PC cluster, as shown in Figure 2. The maximum resolution of the display wall is 6,144 × 3,072 (without overlapping), or about 20 million pixels.

The display wall prototypes include several subsystems in addition to the projector array and the PC cluster. The subsystems include a console to control the display wall system, a compute cluster to provide compute cycles for certain tasks, a storage server to store visualization datasets, a sound server to implements 8-channel spatialized sound, and an input cluster to input from video source, a Gyro-mouse, and a microphone. The second prototype system also includes an

HDTV receiver and a router to connect the display wall system to the several sites via either the Internet or dedicated links for remote data visualization.

All the PCs are connected by an 100 Base-T Ethernet network. In addition, the PCs of the display wall, compute cluster, and storage server are connected by a Myrinet system area network.

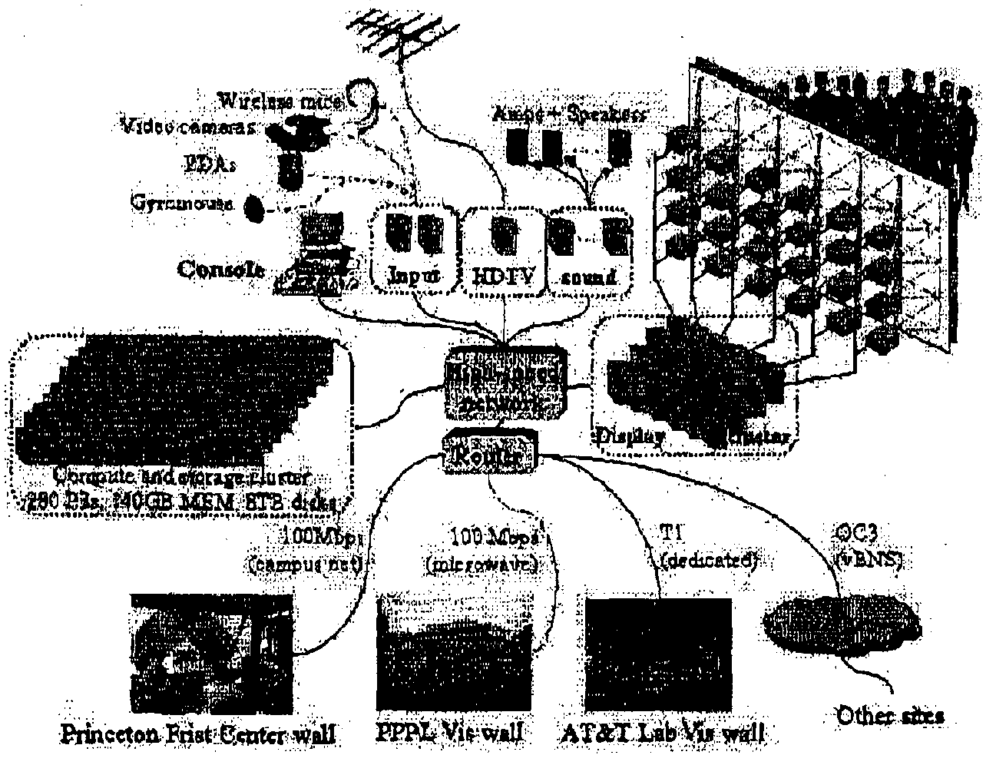

Figure 2: Princeton's second Scalable Display Wall prototype system.

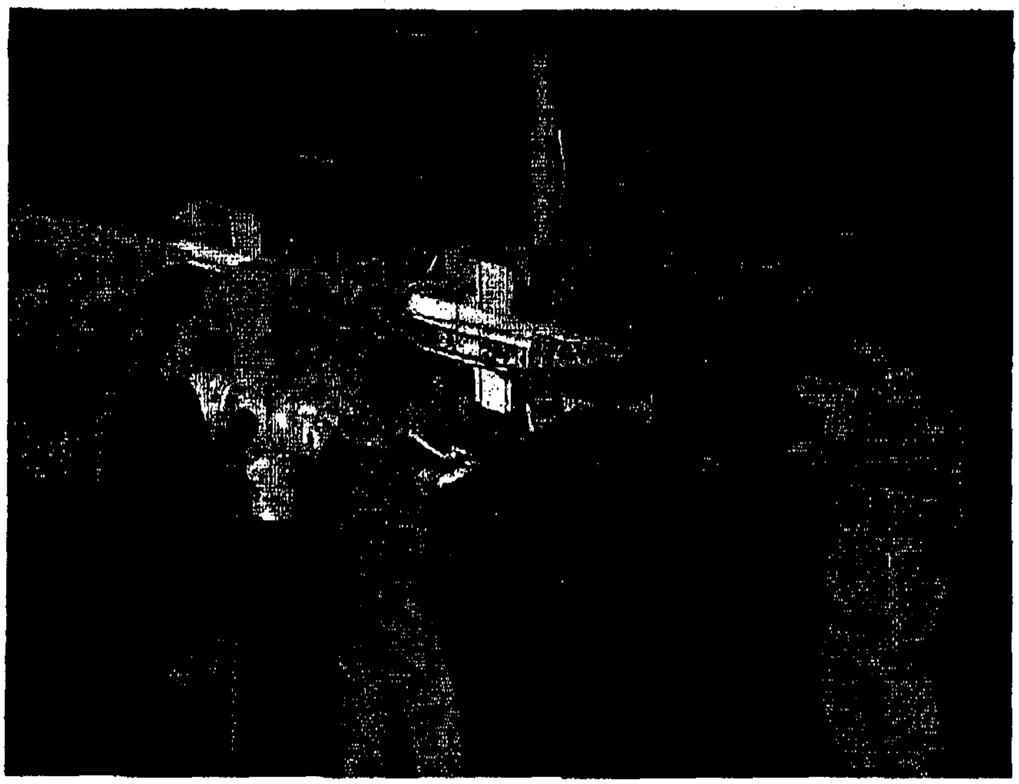

Figure 3: Looking at NASA Space Station on Princeton Display Wall System

We have developed several software tools in order to use the scalable display wall systems. First, we have developed a multimedia viewer that is capable of showing images in various formats efficiently. The viewer can also perform fading in and fading out. The viewer allows users to program with a simple script to interact contents with a Gyro-mouse input. Users can use the viewer to prepare a high-resolution image slide show with multi-channel audio. Figure 3 shows a simple usage of this tool on the display wall system.

The second software tool is a display wall manager. One important function of the display manager is to control the projectors in the projector array individually or together such as power on and off, zoom in and out, contrast and brightness adjustments, and so on. Another function is to manage the PC cluster that drives the display wall such as starting and killing an application program.

The third tool being developed is a virtual display driver (VDD). It is a mechanism to intercept the 2-D primitives to the display driver and distributes the 2-D primitives to the display cluster nodes. This tool allows users to run any Windows' 2-D applications such as Browsers, Photoshop, and Powerpoint, on the display wall.

3.2 Seamless Tiling

The focus of our work has been to provide automatic calibration using an uncalibrated video camera. This is different from the work by other groups who assume a calibrated camera and do not consider scalability issues in their studies.

We have successfully developed two approaches. The first approach is to use an inexpensive, uncalibrated camera to measure the relative mismatches between neighbor projectors. Since the camera only has to make "binary" decisions regarding these measurements, it is free to zoom and pan arbitrarily close to the target spot to obtain highly accurate observations. The alignment algorithm employs the simulated annealing technique to put the local observations together into a self-consistent global picture, and find a set of projection mappings that are consistent with the observations. This approach can complete the calibration of an 8-projector display wall system in tens of minutes to achieve sub pixel accuracy. But, the algorithm runs slowly for a large scale system.

The second approach is a vision-based calibration system for large format multi-projector displays. A spanning tree of homographies, automatically constructed from several camera images, accurately registers arbitrarily-mounted projectors to a global reference frame. Experiments on the 24-projector array demonstrate that the algorithm achieves sub-pixel accuracy even on large display surfaces. A direct comparison with the previous best algorithm shows that our technique is significantly more accurate, requires far fewer camera images, and runs faster by an order of magnitude. It takes less than 10 minutes to calibrate a 24-projector array using this system, whereas it takes over 100 minutes using the previous approach.

In addition, we have invented an optical blending method which uses blades to modulate the aperture of projections to deal with the "non-zero-black" problem when projected images overlap.

3.3 Parallel Rendering on A PC Cluster

The main goal of this research is to develop a fast algorithm and implementation that can take advantage of 3-D graphics accelerators with minimal communication requirements.

Our first study is to build a sort-first based distributed OpenGL system to run existing sequential OpenGL application on a scalable display system. The first is to use a Dynamic Linked Library (DLL) to intercept OpenGL primitives and distribute them via a remote procedure call mechanism to PC nodes that drive the projector array of the scalable display wall system. This approach requires minimum modification to OpenGL applications. The second approach is a synchronized programming mechanism for running multiple instances of a desktop application on the display wall PCs, to make the application appear as a high-resolution application on the display wall. The approach is to implement a virtual machine that intercepts primitives at system call level. The related system calls synchronize with each other. This approach requires no source code access to OpenGL applications.

We have demonstrated several applications with this approach. Figure 3 shows a walkthrough application developed by Co-PI Thomas Funkhouser and his student using this approach.

Figure 4: A walkthrough program using a simple sort-first algorithm.

Our second research is to develop a fast communication mechanism that work well for the parallel rendering systems. We have leveraged the Virtual Memory-Mapped Communication (VMMC) mechanism developed in the SHRIMP (Scalable High-performance Really Inexpensive MultiProcessor) project. VMMC implements a protected, reliable, user-level communication protocol and its end-to-end latency at the user level is about 13 microseconds and its peak user-level bandwidth is about 100Mbytes/sec on the Myrinet.

In the area of parallel rendering, we have focused on load balancing for parallel rendering to maximize performance and minimize communication requirements. Our first investigation is to understand how to do load balancing on a client machine connecting to a scalable display wall's display cluster at object level. This approach assumes a fully replicated scene database on each node in the cluster. By replicating the database, the client node can simply tell the server nodes which object to render. We have developed a sort-first method on the client node to partition the screen space with a K-D tree algorithm to perform load balancing. We have implemented this algorithm performs better than traditional bucket based algorithms. Our implementation of this approach is the first sort-first algorithm implemented for a scalable display wall driven by a PC cluster.

We have investigated a hybrid of sort-first and sort-last approach for parallel polygon rendering on a cluster of PCs. Unlike previous methods that statically partition the 3D model and/or the 2D image, our approach performs dynamic, view dependent and coordinated partitioning of both the 3D model and the 2D image. Using a specific algorithm that follows this approach, we show that it performs better than previous approaches and scales better with both processor count and screen resolution. Overall, our algorithm is able to achieve interactive frame rates with efficiencies of 55.0% to 70.5% during simulations of a system with 64 PCs.

Another research effort is to investigate how to perform parallel rendering well within the memory constraints and communication limitations of a networked cluster. Previous systems have required the entire 3D scene to be replicated in memory on every PC. While this approach can take advantage of view-dependent load balancing algorithms and thus largely avoid the problems of inter-process communication, it limits the scalability of the system to the memory capacity of a single PC. We proposed a k-way replication approach in which each 3D primitive of a large scene is replicated on k out of n PCs (k << n). The key idea is to support 3D models larger than the memory capacity of any single PC, while retaining the reduced communication overheads of dynamic view-dependent partitioning. We have investigated several algorithms for distributing copies of primitives among PCs and for dynamic load balancing under the constraints of partial replication. Our main result is that the parallel rendering efficiencies achieved with small replication factors are similar to the ones measured with full replication. By storing one-fourth of Michelangelo's David model (800MB) on each of 24 PCs (each with 256MB of memory), our system is able to render 40 million polygons/second (65% efficiency).

3.4 Applications

Our goal is to develop a visualization system infrastructure to visualize large datasets and our focus is to understand how to extract isosurface that supports parallelization on a PC cluster.

We have developed a new isosurface extraction algorithm with several desired features for parallelization and remote data visualization and on a network attached display wall, including multi-resolution, view dependent, parallelizable and easy to incorporate a remote visualization protocol. The algorithm is a hybrid of ray casting and propagation. The algorithm uses ray casting to find a cell that has isosurface within the threshold and then propagate the isosurface through its neighboring cells. We have shown with the visible human datasets that this algorithm works quite well. This algorithm is quite easy to be parallelized and extended for remote data visualization.

Our second effort is to study how to extend the algorithm for remote data visualization. We have investigated two mechanisms. The first is to perform isosurface extraction in a hierarchical way by using the hybrid method above using a hierarchical index data structure. This allows users to see the shape of the isosurface quickly and the refined isosurface progressively. The second mechanism is to let the extraction machine keep track of what primitives the viewing machine has at all times. The mechanism uses the propagation group as a unit to inexpensively identify which group of primitives the remote display already has. Our implementation and experiments show that this algorithm performs much better than other known methods.

In addition to the proposed tasks, we have worked with Ben Shedd, an IMAX film producer and director, on how to present information on a scalable display wall system. Several courses have been taught to explore information visualization. Our experience shows that a scalable display wall can make a dramatic difference for visualization. Proper utilization and information presentation is important.

4 Impacts

The most important impact this project has made is to propose and to advocate the concept of building a scalable display wall system using a cluster of PCs and an array of commodity presentation projectors. By building the first scalable display wall prototype system, this project demonstrates to the community the viability and astounding result of this approach. At the beginning of the project, the community uses high-end SGI machines to drive high-end (CRT) projectors. At the end of this project, building a scalable display wall with this approach has become a common methods in almost every research group in the data visualization community.

We have served as the editors of a special issue of *IEEE Computer Graphics & Applications* on large display systems, to raise the awareness of the scalable display wall approach. Several DOE groups (ANL, LANL, LLNL, and Sandia) and other research groups have published their approaches to display wall in this forum.

We have been working with the AccessGrid team to connect our large-format visualization environment to collaborate among multiple institutions. We have invited these researchers several times to see our system on site. We worked with Rick Steven's group at Argonne National Lab to help them build their scalable display wall. Princeton has given ANL the design and prototype of a sophisticated project mount system co-designed by Princeton and Intel. Princeton has also provided the optical blending technology for seamless tiling. Their demonstration at Supercomputing conferences includes both technologies.

We have been working closely with Bill Tang's visualization group at Princeton Plasma Physics Lab to help them construct a small scalable display wall system, similar to our prototype system.

We have shown the display wall to Senator Bill Bradley and Senator Bill Frist to encourage the support for high-performance computing. Senator Frist was so excited to put a display wall in the campus student center he donated to Princeton.

5 Publications

Some of the publications were after the funding period, but some of the work was funded in part by this grant.

1. Yuqun Chen, Stefanos N. Damianakis, Sanjeev Kumar, Xiang Yu, and Kai Li. Porting a User-level Communication Architecture to NT: Experience and Performance (1999). In *Proceedings of 3rd Usenix Windows NT Symposium*, Seattle, Washington, July 12-15, 1999.

2. Rudrajit Samanta, Jiannan Zheng, Thomas Funkhouser, Kai Li, and Jaswinder Pal Singh. Load Balancing for Multi-Projector Rendering Systems. *SIGGRAPH/Eurographics Workshop on Graphics Hardware*, Los Angelos, CA, August, 1999.

3. Kai Li and Yuqun Chen, "Optical Blending for Multi-Projector Display Wall System." In *Proceedings of the 12th Lasers and Electro-Optics Society 1999 Annual Meeting*. November 1999.

4. Thomas Funkhouser and Kai Li, Large Format Displays. *IEEE Computer Graphics and Applications*, 20(4): 20-21, July 2000.

5. Kai Li, Han Chen, Yuqun Chen, Douglas Clark, Perry Cook, Stefanos Damianakis, Georg Essl, Adam Finkelstein, Thomas Funkhouser, Timothy Housel, Allison Klein, Zhiyan Liu, Emil Praum, Rudrajit Samanta, Ben Shedd, Jaswinder Pal Singh, George Tzanetakis, Jiannan Zheng, Building and Using A Scalable Display Wall System. *IEEE Computer Graphics & Applications*, Volume 20, Number 3, July/August Issue, 2000.

6. Rudrajit Samanta, Thomas Funkhouser, Kai Li, and Jaswinder Pal Singh, Sort-First Parallel Rendering with a Cluster of PCs, *Technical Sketch in SIGGRAPH 2000*, July, 2000.

7. Rudrajit Samanta, Thomas Funkhouser, Kai Li, and Jaswinder Pal Singh, Hybrid Sort-First and Sort-Last Parallel Rendering with a Cluster of PCs. In *Proceedings of SIGGRAPH/Eurographics Workshop on Graphics Hardware*, August, 2000.

8. Yuqun Chen, Douglas W. Clark, Adam Finkelstein, Timothy Housel, and Kai Li, Automatic Alignment Of High-Resolution Multi-Projector Displays Using An Un-Calibrated Camera, To Appear in IEEE Visualization 2000, Salt Lake City, Utah, October 8-13, 2000.

9. Zhiyan Liu, Adam Finkelstein, and Kai Li. Progressive View-Dependent Isosurface Propagation. IEEE TCVG Symposium on Visualization (VisSym 2001). Ascona, Switzerland. May 28 - May 30, 2001.

10. Yuqun Chen, Han Chen, Douglas W. Clark, Zhiyan Liu, Grant Wallace, and Kai Li. Software Environments for Cluster-based Display Systems (2001). The First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), Brisbane, Australia, 15-18 May 2001.

11. Sanjeev Kumar, Yitzhak Mandelbaum, Xiang Yu, Kai Li. ESP: A language for programmable devices. *Proceedings of ACM SIGPLAN Programming Language Design and Implementation (PLDI).* June 2001.

12. Han Chen, Yuqun Chen, Adam Finkelstein, Thomas Funkhouser, Kai Li, Zhiyan Liu, Rudrajit Samanta, and Grant Wallace. Data Distribution Strategies for High-Resolution Displays. *Computers & Graphics*, Special Issue on Mixed Realities - Beyond Conventions, 25(5):811-818, October 2001.

13. Rudrajit Samanta, Thomas Funkhouser, and Kai Li. Parallel Rendering with K-Way Replication. *IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, October, 2001.

14. Han Chen, Rahul Sukthankar, Grant Wallace, T.-J. Cham. Calibrating Scalable Multi-Projector Displays Using Camera Homography Trees. In Computer Vision and Pattern Recognition 2001 (technical sketch), Kauai, Hawaii, December 2001.

15. Han Chen, Grant Wallace, Anoop Gupta, and Kai Li, Tom Funkhouser, Perry Cook, Experiences with Scalability of Display Walls, In 7th Annual Immersive Projection Technology Symposium (IPT), March 2002.

16. Han Chen, Kai Li, and Bin Wei. A Parallel Ultra-High Resolution MPEG-2 Video Decoder for PC Cluster Based Tiled Display System. In *Proceedings of International Parallel and Distributed Processing Symposium* (IPDPS2002), April 2002.

17. Zhiyan Liu, Adam Finkelstein, and Kai Li. Improving Progressive View-Dependent Isosurface Propagation. *Computers & Graphics*. 26(2): 209-218. 2002.