A Real-time Seamless Tiled Display System for 3D Graphics

Zhiyun Li   Amitabh Varshney
Department of Computer Science and UMIACS
University of Maryland at College Park
{zli, varshney}@cs.umd.edu

Abstract

We outline our seamless tiled display system for interactive 3D graphics applications that is low-cost, easy to calibrate, scalable, and portable. Our system achieves geometric alignment in software by pre-warping the 3D space in contrast with the current systems that usually achieve this by 2D image pre-warping. Our system accomplishes this through real-time image capture from a digital camcorder, image segmentation, and derivation of the 3D warping matrices for each 3D graphics pipeline that feeds a projector. Our prototype system demonstrates our results on a 2 × 2 tiled array of projectors.

1. Introduction

Virtual environments (VEs) are finding increasing use in design, simulation, and training activities related to medicine, science, and engineering. This recent growth of interest in VEs has been accompanied by a corresponding increase in the types of devices for sensory feedback, especially visual feedback. At present, the multifarious display devices can be broadly classified into three main categories [15]:

(a) Head-coupled displays, including head-mounted displays,
(b) Spatially-immersive displays, including CAVE [8] and tiled displays [16] [18] [9], and
(c) Virtual model displays, including immersive and responsive workbenches [14][20].

Our target application is conformational steering for protein datasets that multiple users can collaboratively use. The head-coupled displays are ill suited for such applications, or for tasks that take several hours (due to eye and neck fatigue). Similarly, the virtual model displays are not suited for applications that are rich in visual information content and require a high-resolution display or require a wide field of view and large area. Spatially immersive displays, such as wall-sized tiled displays, allow long periods of work, offer a very high field of view and resolution, and afford a strong self-presence. Several groups have reported that wall-sized tiled displays are more supportive for collaboration and learning than regular monitors [30]. Users stay longer in such displays, move and discuss the datasets more, and treat such displays as "murals" that they repeatedly touch, inspect, walk around and see from different viewpoints.

In this paper we outline our tiled display system that achieves geometric alignment for 3D graphics applications by pre-warping 3D objects on the fly.

2. Previous and Related Work

In the past decade, a lot of research has been done to create large display wall systems. These various systems can be roughly classified into the following types according to their configuration:

2.1. Systems with fixed projectors

In this type of systems, the projectors are fixed at pre-specified positions and mechanically precisely aligned. Cruz-Neira *et al.* [8] have created an immersive display system called CAVE (CAVE Automatic Virtual Environment). This system uses one rear-projection screen for each of the three walls and one down-projection screen for the floor to generate the immersive display. The system uses electromagnetic sensors for tracking and LCD stereo shutter glasses to create an effective and convincing virtual environment.

Hereld *et al.* [10] have introduced the basic issues in building a projection-based tiled display system. Several mechanical alignment methods are described. These include Intel-Princeton projector positioner with aperture modulation frame, the improved version at Argonne National Laboratory, and the commercial projector mounting system used in Lawrence Livermore National Laboratory. The NCSA display wall system at the University of Illinois [2] is also using such positioners.

Schikore *et al.* have built the InfinityWall [29] visualization system using two SGI Onyx2 visualization servers. They use single-lens LCD-based projectors so that the convergence does not drift over time. The geometric alignment is achieved by finely adjusting the lens position.

The display walls with fixed projector configurations have greatly reduced overhead for geometric alignment. However, such systems are also difficult to set up, calibrate and not easily portable.

2.2. Geometric calibration by image warping

The projectors in this type of display wall systems do not have fixed positions and do not require precise manual geometric calibration. The projectors only need to be roughly aligned. The geometric calibration is then achieved by 2D image warping.

The Princeton Display Wall project by Li *et al.* [16] has built a large-format display using a cluster of workstations with low-cost commodity components. In this system, the geometric alignment is done by first calculating a 3×3 projection matrix for each projector, and then using it to resample the images by an image warping process. Further details of this method are available in the paper by Chen *et al.* [4]. Chen *et al.* [6] have described another alignment method based on the mapping function for a projector. Chen *et al.* discuss the details of their software implementation in [5][7].

Majumder *et al.* [17] have presented an algorithm designed to align and blend images to create a single seamless display. This algorithm produces a texture map from images from a camera cluster and then uses the texture map to resample the original images. Raskar *et al.* [23] describe a robust calibration algorithm for arbitrary display surfaces. This algorithm first computes warping functions for each projector to achieve geometric alignment and then computes intensity-weighting basis functions to achieve blending across the seams. This algorithm is based on 2D texture mapping. Yang *et al.* [33] have presented a reconfigurable multi-projector display system that generates accurate warping and blending functions. However, the calibration is still based on 2D texture mapping.

The above techniques at North Carolina and Princeton have pioneered the use of software warping to accomplish geometric alignment. This is an excellent solution for display of 2D environments. However, using these techniques with 3D graphics scenes requires first rendering them into a texture memory followed by texture mapping – a relatively expensive proposition since the frame-buffer read back is usually slow.

Raskar *et al.* [26] have described a multi-projector system to display panoramic imagery. The geometric registration is achieved by using cameras to recover the 3D display environment including the configuration of the projectors, the geometry of the display surface, and the location of the viewer. The overlap is removed by surface mesh unification.

Raskar [27] has presented methods to correct the keystoned images created by oblique projectors by using a collineation matrix and a single-pass rendering. First, the collineation matrix between projected image and the image in the frame buffer is computed, then a new 4×4 matrix is derived from the 3×3 matrix to apply to 3D points and render perspectively-correct images. This single-pass approach is very efficient in rendering perspectively-correct images in oblique projection because the rendering and the warping are achieved by traditional graphics hardware. However, since the projector is calibrated by only one matrix, the four corners of the projection area cannot be controlled completely. But for the tiled display wall, we need precise geometric calibration for each projector to achieve seamlessness. So, if we directly apply this approach for our tiled display wall, the seams between projection areas are unavoidable. In this paper, we extend this approach by creating triangular-base frustums. Our approach works for both oblique and orthogonal projectors.

2.3. Other Related Work

Humphreys *et al.* have described WireGL [12][13], which is a scalable graphics software system for combining the power of a cluster of workstations. WireGL can be used in both standalone displays and tiled display walls with a very high rendering performance. Samanta *et al.* [28] have developed several coarse-grained partitioning algorithms for a parallel rendering system.

Raskar *et al.*[25][24] have presented environments in which the physical models are surrounded by projectors to display seamless images on the surface of models. This is very effective to add color or texture on the surface of models. Their system uses real models and projected images to create compelling 3D virtual objects.

2.4. Applications of the display wall

Raskar *et al.* [22] have constructed a spatially immersive display system to simulate an office environment. The InforWall [32] is a PC-cluster-based system built to manage and display network data. Buxton *et al.* [3] use their display system in automotive design. Bishop *et al.* [1] create an office display environment by large-screen projection. Mynatt *et al.* [19] use the display wall to create a virtual whiteboard. PowerWall [21] is a high-performance, high-resolution visualization system for scientific visualization applications.

3. System Organization

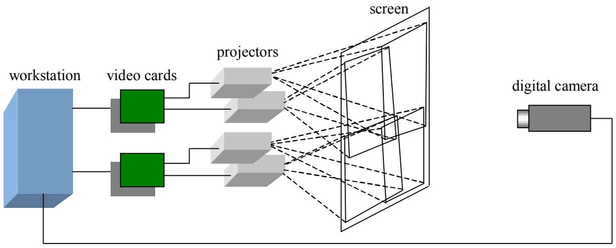


Figure 1: High-level system block diagram

Our system consists of low-cost off-the-shelf components. As shown in Figure 1, it includes one Sony DCR VX2000 digital video camera, one Da-lite portable wall-sized (6'×8') display screen, four Proxima Ultralight X350 DLP projectors arranged in a 2×2 array behind the display wall, and one dual-processor 933MHz Dell Precision 620 workstation installed with two Appian Jeronimo 2000 video cards. Each video card has two graphics adapters with 3D hardware acceleration.

Sukthankar *et al*. [31] have used dynamic shadow elimination for front-projection systems. However, we decided to use a rear-projection system to avoid shadows of viewers when they get close to the screen. Our projectors are roughly orthogonal to the display wall and the areas of projection only need to be roughly aligned and overlapped, making the system portable. The digital camera is used to capture and provide feedback to the geometric calibration information over the four projectors. By exploiting the multi-monitor 3D hardware acceleration of the Appian Jeronimo 2000 video cards, the system can display seamlessly aligned 3D graphics in real time with our 3D geometric alignment algorithm. Figure 2 is a snapshot of the working system behind the display wall.

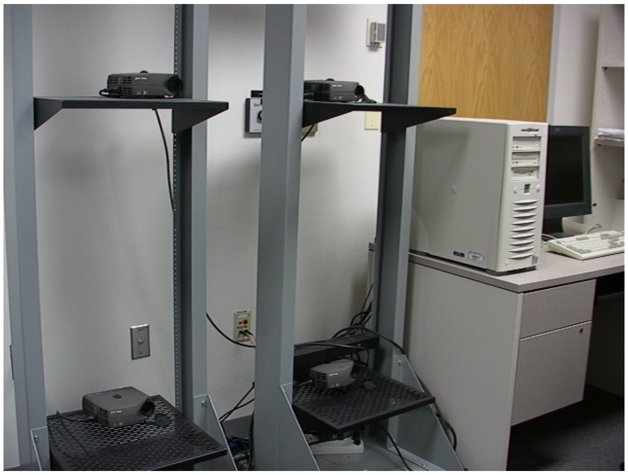


Figure 2: A system snapshot behind the display wall.

4. Geometric Calibration

Our system achieves 3D geometric calibration using the following three steps:

(1) Use a digital camera to align 2D images on the display wall.

(2) Achieve pixel-level accuracy of calibration by software.

(3) Derive 3D transformation matrices and achieve 3D geometric calibration.

The following subsections give detailed description for each step.

4.1. Initial geometric calibration by camera

Before we start the geometric calibration, we assume that the projectors are roughly orthogonal to the display surface and the projection areas partially overlap each other. Such an example is shown below in Figure 3. This configuration is easy to get manually.

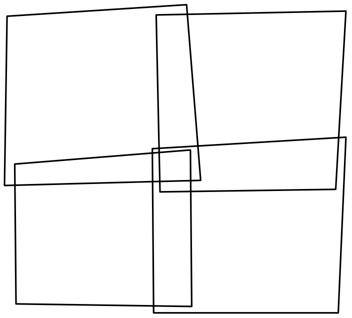


Figure 3: Screen layout

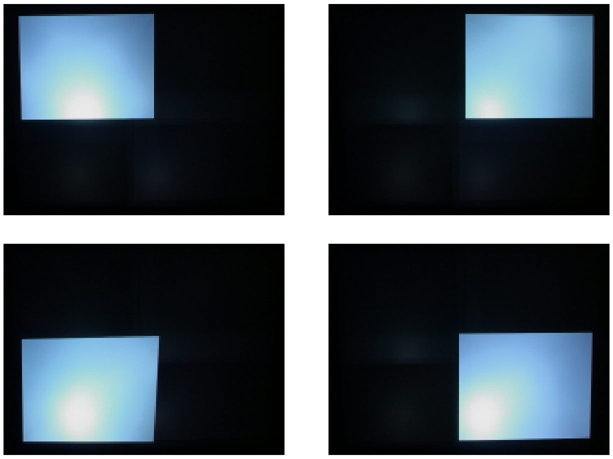


Figure 4: Pictures for each projection area

We use a digital camera to take a picture for each projection area as shown in Figure 4. From these pictures we calculate the *effective display area* on the display wall.

The effective display area is the maximum rectangular region that can fit within the four projection areas and that covers exactly one corner of each projection area. The effective display area of Figure 3 is shown in Figure 5(a). We segment the effective display area into four roughly equal-sized *normalized display rectangles*. Each normalized display rectangle is completely covered by a single projector's projection area. The segmentation of Figure 5(a) into four normalized display rectangles is shown in Figure 5(b) in thick lines.

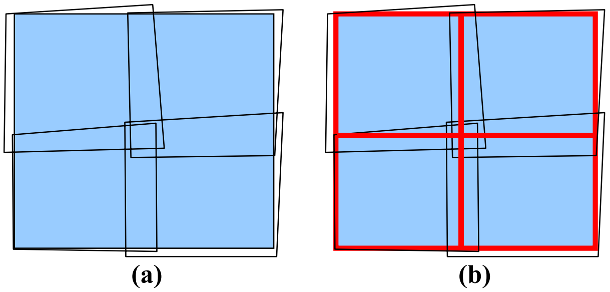


Figure 5: (a) Effective display area (b) segmentation of the area into four normalized display rectangles

After segmentation, we can derive the 2D warping matrices for each projector. Let the outer quadrilateral **P** = $P_1P_2P_3P_4$ shown in Figure 6(a) represent the projection area of a single projector and the inner rectangle **Q** = $Q_1Q_2Q_3Q_4$ represent the normalized display rectangle (shown in thick lines in Figure 5(b)) for that projector. Our transformation proceeds in two steps shown in Figures 6(a) and 6(b). In the first step the inner and outer quadrilaterals **P** and **Q** are transformed from the display-wall space to quadrilaterals **P'** and **Q'** in the frame-buffer space. As can be seen in Figure 6(a), this causes the quadrilateral **P** to transform to a rectangle **P'** and the normalized display rectangle **Q** to transform to a quadrilateral **Q'**. In the second transformation step the contents of the frame-buffer **P'** are squeezed into the inner quadrilateral **Q'**. This is achieved by transforming **P'** to **P"** = **Q'** as shown in Figure 6(b). Each of the two transformation steps involves mapping one convex quadrilateral into another. To do this accurately we model this problem as mapping of two triangles into two other triangles as shown in Figure 6(c), thereby using two matrices for each of the two transformation steps shown in Figures 6(a) and 6(b). Each affine transformation matrix **M** shown in Figure 6(d) is computed as follows:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 \\ t_1 & t_2 & t_3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 \\ t_1 & t_2 & t_3 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

Here (x_i, y_i) are the vertices of a triangle before transformation and (s_i, t_i) are the transformed vertices. The steps shown in Figures 6(a) and (b) can be combined into a single step by compositing their transformation matrices appropriately.

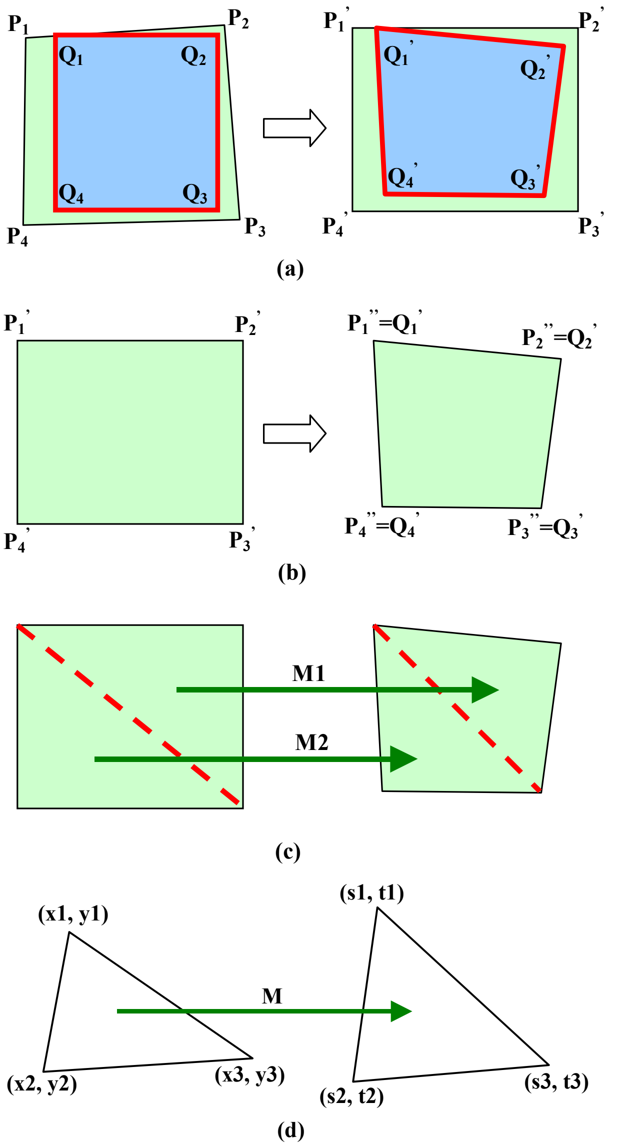


Figure 6: (a) Transformation from display-wall space to frame-buffer space, (b) Transformation to maximally cover the frame-buffer space, (c) Representing the transformation from one quadrilateral to another as two transformations on triangles, (d) Transforming one triangle to another

4.2. Pixel-level Accuracy Calibration

We first attempted geometric calibration using digital imaging. However, since the resolution of our digital camera was less than that of the display wall and we were exploring solutions that would proceed with a single image of the entire display wall, the results we got from camera-based geometric calibration were not very accurate. Using a camera with higher resolution is a possible solution to this problem. However, this approach does not scale well for systems with more projectors. Another solution might have been to use multiple screen shots of the display wall or to use precise positioners or other mechanical devices, but this increases the system set-up time.

In our current display wall we have built a software system to get precise geometric calibration. The program moves the corners of each segment area to achieve pixel-level accuracy in calibration. The control interface is shown in Figure 7. The selected segment area is highlighted and the corner under control is indicated by a "×" mark. By running this software, the user can move the specified corner around its position with different granularities of span length. This process is repeated for all the corners until we achieve the perfect geometric calibration.

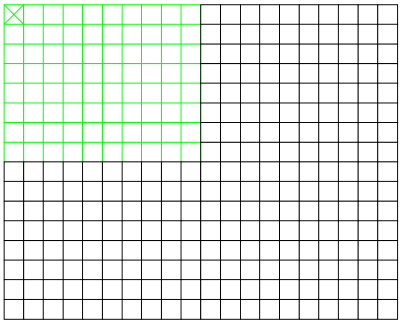


Figure 7: Interface sketch of positioner simulator.

4.3. 3D geometric calibration

We have derived the eight transformation matrices for the four projectors in Section 4.1. This texture-mapping-based approach works well on 2D images. However, this approach does not extend to display of real-time 3D objects because of the relatively slow speeds of reading back the frame-buffer. To overcome this limitation, we next generalize this approach to a 3D case.

Let us assume that the Z-axis is perpendicular to the display wall surface. For this case we can directly create the 3D geometric calibration matrices from its 2D counterpart:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} m_{11} & m_{12} & 0 & m_{13} \\ m_{21} & m_{22} & 0 & m_{23} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now we need to apply the eight 4×4 matrices to 3D space. We create two view frustums with triangular bases and make use of the 3D hardware acceleration of the graphics processors. Figure 8 shows a frustum with a rectangular base that is split diagonally across the display plane into two frustums with triangular bases. Then, for each triangular-base frustum, we can apply the 4×4 transformation matrix. In the available graphics APIs (OpenGL and DirectX) there is no explicit function that can create such frustums. Therefore, in addition to the regular front and back clipping planes we define three lateral clipping planes on the sides of each triangular-base frustum to set the boundary of the transformation space as shown in Figure 9.

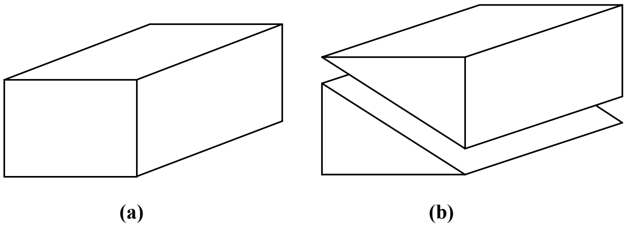


Figure 8: (a) Frustum with a rectangular base, (b) Two frustums with triangular bases

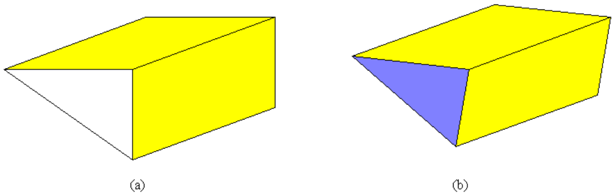


Figure 9: (a) Triangular-base frustum with clipping planes (yellow) attached on sides. (b) Transformation of frustum together with clipping planes.

5. Implementation Details

We use Microsoft DirectX 8.0 as our graphics API instead of OpenGL because the latter does not support multi-monitor hardware acceleration. Using DirectX, we create a "display device" for each projector so that one graphics processor controls the display of one projector.

To make our system scalable, we divide the 3D coordinate space into the following four subspaces, one subspace for each processor:

$$\{(x, y, z): x \geq 0, y \geq 0\},$$
$$\{(x, y, z): x \geq 0, y < 0\},$$
$$\{(x, y, z): x < 0, y \geq 0\},$$
$$\{(x, y, z): x < 0, y < 0\}.$$

Thus, instead of sending the complete 3D data to each processor, we only send the data that is necessary for rendering on each processor. For each processor we further divide the 3D data into two triangular-base frustums. In the next section, we will see the scalability we achieve by this approach. The pseudo-code for our system that we just discussed is shown in Figure 10.

```
//initialization.
Load calibration data;
Compute transformation matrices;
Compute clipping planes;

Enumerate monitors; //in our system,
                    //"monitor" means "projector"
For each monitor:
      Create a device for each monitor
      with hardware support;

//display data on one or more monitors.
Set the clipping planes and transformation matrices
      for the chosen device;
Pre-multiply the matrix for each triangular-base frustum;
Display;
```

Figure 10: The pseudo-code of our system.

After the geometric calibration, we can still see the crisp seams between the projected areas. Several research groups have used different methods to achieve seamlessness in tiled displays. Li *et al.* [16] have used optical edge-blending to generate seamless rendering. This method uses placement of physical cardboard occluders around a projector's light path. The partial diffraction of light about these cardboard occluders helps in softening the seams between individual projected areas. To remove the seams and blend across the projected images we use a method similar to the one proposed by Humphreys *et al.* [11] for the "Stanford Interactive Mural". This system uses *feathering* polygons to remove the seams between overlapping regions. The cost of feathering is acceptable compared with hardware-blending solutions. We have found this approach to be very effective in removing seams.

We can use more than two matrices for one projector but our results show that this is not necessary because four control points are enough to achieve exact alignment assuming there are only linear distortions. We have not included color calibration in this work; we are currently working on it.

6. Results

Our prototype system achieves real-time 3D geometric alignment with hardware acceleration for each graphics pipeline. We have created a software library so that other 3D programs can be ported to the display wall easily. Figure 12 shows the application in protein steering. Figures 13—15 show further results of our system.

We have tested our system with several 3D models. Table 1 compares the performance in two cases: (i) rendering seamlessly on a tiled display wall using four projectors and four graphics processors with distributed data and (ii) traditional rendering using a single projector and a single processor. As can be seen, the performance in these two cases is very close. The results show that our system is highly scalable.

Table 1: Performance of some 3D models.

Model	Num of Tris	Four projectors	One projector
		Framerate (per sec)	Framerate (per sec)
Teapot	4k	32.26	43.48
Cow	6k	25.64	32.26
Bunny	69k	2.76	2.56
Dragon	202k	1.02	0.89

Our system achieves pixel-level accuracy. Figure 11 shows a close-up view of the Stanford Bunny's ear rendered with and without alpha blending.

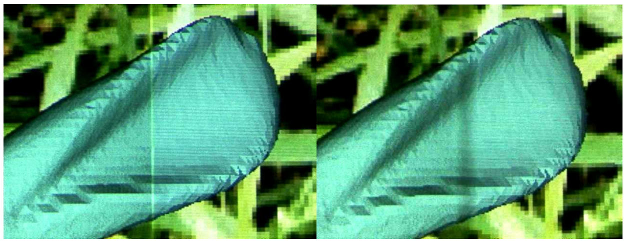

(a) (b)

Figure 11: Bunny's ear (a) Without alpha blending. (b) With alpha blending

7. Conclusions

In summary, our system achieves real-time 3D geometric alignment with hardware acceleration for each graphics pipeline. It accomplishes this by pre-warping 3D objects at every frame followed by conventional 3D graphics rendering.

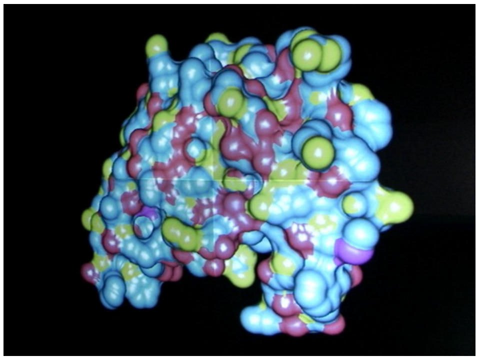


Figure 12: HIV protease on display wall

8. Acknowledgments

We would like to acknowledge several constructive comments from the anonymous reviewers. We thank the Maryland's Graphics and Visual Informatics Lab members for several beneficial discussions, especially Chang Ha Lee for his help on the visualization of the HIV protease dataset. We would also like to thank Fritz McCall, Brad Erdman, and other UMIACS technical staff members for their help in setting up the display wall system. We will like to acknowledge the Stanford Graphics Lab for providing the models of the Bunny and the Dragon and the Protein Data Bank for the model of the HIV protease. This work has been supported in part by the NSF grants ACR-98-12572 and IIS-00-81847.

References

[1] G. Bishop and G. Welch, "Working in the Office of 'Real Soon Now'", *IEEE Computer Graphics and Applications,* vol 20(4), pp 76-78, 2000.

[2] Building Tiled Displays, University of Illinois at Urbana-Champaign. http://www.ncsa.uiuc.edu/Divisions/DMV/Vis/Projects/TiledWall/

[3] W. Buxton, G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach, "Large Displays in Automotive Design", *IEEE Computer Graphics and Applications,* vol 20(4), pp 68-75, 2000.

[4] H. Chen, R. Sukthankar, G. Wallace, and T. Cham, "Calibrating Scalable Multi-Projector Displays Using Camera Homography Trees", *Computer Vision and Pattern Recognition 2001 (technical sketch),* Kauai, Hawaii, December 9-14, 2001.

[5] H. Chen, Y. Chen, A. Finkelstein, T. Funkhouser, K. Li, Z. Liu, R. Samanta, and G. Wallace, "Data Distribution Strategies for High-Resolution Displays", *Computers & Graphics Journal, Special Issue on Mixed Realities - Beyond Conventions*, 2001.

[6] Y. Chen, D. Clark, A. Finkelstein, T. Housel, and K. Li, "Automatic Alignment of High-Resolution Multi-Projector Displays Using An Un-Calibrated Camera", *IEEE Visualization 2000,* pp 125-130, Salt Lake City, Utah, October, 2000.

[7] Y. Chen, H. Chen, D. Clark, Z. Liu, G. Wallace, and K. Li, "Software Environments for Cluster-based Display Systems", *IEEE International Symposium on Cluster Computing and the Grid,* pp 202-210, Brisbane, Australia, May 2001.

[8] C. Cruz-Neira, D. Sandin, and T. DeFanti, "Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE", *SIGGRAPH 1993 Conference Proceedings*, *Annual Conference Series*, Vol. 27, pp. 135-142, Anaheim, CA, August 1993.

[9] A. Fox, B. Johanson, P. Hanrahan, and T. Winograd, "Integrating Information Appliances into an Interactive Workspace", *IEEE Computer Graphics and Applications,* vol 20(3), pp 54-65, 2000.

[10] M. Hereld, I. R. Judson, and R. L. Stevens, "Introduction to Building Projection-based Tiled Display Systems", *IEEE Computer Graphics and Applications,* vol 20(4), pp 22-28, 2000.

[11] G. Humphreys, P. Hanrahan, "A Distributed Graphics System for Large Tiled Displays", *IEEE Visualization 1999*, pp. 215-224, San Francisco, CA, October 1999.

[12] G. Humphreys, M. Eldridge, I. Buck, and P. Hanrahan "Distributed Rendering for Scalable Displays", *IEEE Supercomputing 2000*, pp. 60-60, Dallas, TX, November 2000.

[13] G. Humphreys, M. Eldridge, I. Buck, G. Stoll, M. Everett, and P. Hanrahan, "WireGL: A Scalable System for Clusters", *SIGGRAPH 2001 Conference Proceedings, Annual Conference Series,* pp. 129-140, Los Angles, CA, August 2001.

[14] W. Krueger, M. Berndtgen, C. Bohn, B. Froehlich, H. Schueth, T. Sikora, J. Speier, W. Strauss, G. Wesche, and J. Ziehm, "The Responsive Workbench: A Virtual Working Environment for Architects, Designers, Physicians, and Scientists", *SIGGRAPH 1994 Visual Proceedings*, *Annual Conference Series*, pp. 200-201, Orlando, FL, August 1994.

[15] E. Lantz, S. Bryson, D. Zeltzer, M. Bolas, B. Chapelle and D. Bennett, “The Future of Virtual Reality: Head Mounted Displays Versus Spatially Immersive Displays”, *SIGGRAPH 1996 Conference Proceedings*, *Annual Conference Series*, pp. 485-486, New Orleans, LA, August 1996.

[16] K. Li, H. Chen, Y. Chen, D. Clark, P. Cook, S. Damianakis, G. Essl, A. Finkelstein, T. Funkhouser, A. Klein, Z. Liu, E. Praun, R. Samanta, B. Shedd, J. Singh, G. Tzanetakis, and J, Zheng. “Building and Using A Scalable Display Wall System”, *IEEE Computer Graphics and Applications,* vol 20(4), pp 29-37, 2000.

[17] A. Majumder, G. Meenakshisundaram, W. B. Seales, and H. Fuchs, “Immersive Teleconferencing: A New Algorithm to Generate Seamless Panoramic Video Imagery”, *Proceedings of the 7th ACM International Conference on Multimedia*, pp 169-178, Orlando, FL, October, 1999.

[18] A. Majumder, Z. He, H. Towles, and G. Welch "Achieving Color Uniformity Across Multi-Projector Displays", *IEEE Visualization 2000,* pp 117-124, Salt Lake City, Utah, October, 2000.

[19] E. Mynatt, T. Igarashi, W. Edwards, and A. LaMarca, “Designing an Augmented Writing Surface”, *IEEE Computer Graphics and Applications,* vol 20(4), pp 55-61, 2000.

[20] U. Obeysekare, C. Williams, J. Durbin, L. Rosenblum, R. Rosenberg, F. Grinstein, R. Ramamurthi, A. Landsberg, and W. Sandberg, “Virtual Workbench - A Non-Immersive Virtual Environment for Visualizing and Interacting with 3D Objects for Scientific Visualization", *IEEE Visualization 1996,* pp 345-349, San Franscisco, CA, October, 1996.

[21] PowerWall, University of Minnesota. http://www.lcse.umn.edu/research/powerwall/powerwall.html.

[22] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, “The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays”, *SIGGRAPH 1998 Conference Proceedings, Annual Conference Series*, pp 179-188, Orlando, FL, July, 1998.

[23] R. Raskar, G. Welch, and H. Fuchs, “Seamless Projection Overlaps Using Image Warping and Intensity Blending”, *Proceedings of 4th International Conference on Virtual Systems and Multimedia,* Gifu, Japan, November, 1998.

[24] R. Raskar, G. Welch, and H. Fuchs, “Spatially Augmented Reality”, *Proceedings of 1st International Workshop on Augmented Reality*, San Francisco, CA., November*,* 1998.

[25] R. Raskar, G. Welch, and W. Chen, “Table-Top Spatially-Augmented Reality: Bringing Physical Models to Life with Projected Imagery”, *Proceedings of 2nd International Workshop on Augmented Reality*, pp 64-71, San Francisco, CA., October, 1999.

[26] R. Raskar, M. S. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs, “Multi-Projector Displays Using Camera-Based Registration”, *IEEE Visualization 1999,* pp 161-168, San Francisco, CA, October, 1999.

[27] R. Raskar, “Immersive Planar Display using Roughly Aligned Projectors”, *IEEE Virtual Reality 2000,* pp 109-116, New Brunswick, NJ, March, 2000.

[28] R. Samanta, J. Zheng, T. Funkhouser, K. Li, and J. Singh, “Load Balancing for Multi-Projector Rendering Systems”, *SIGGRAPH/Eurographics Workshop on Graphics Hardware,* pp 107-116, Los Angeles, CA, August, 1999.

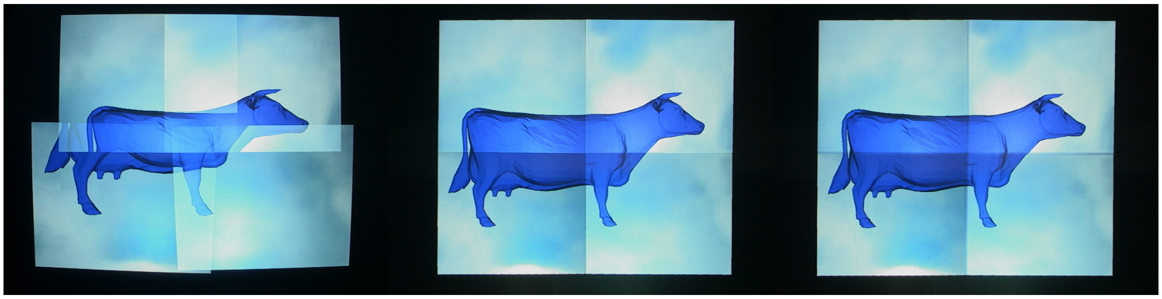
[29] D. Schikore, R. Fischer, R. Frank, R. Gaunt, J. Hobson, and B. Whitlock, “High-Resolution Multiprojector Display Walls”, *IEEE Computer Graphics and Applications,* vol 20(4), pp 38-44, 2000.

[30] R. Stevens, M. Papka, J. Ahrens, K. Li, and D. Reed, “Next-Generation Visualization Displays: The Research Challenges of building Tiled Displays”, *IEEE Visualization 2000 Panel,* pp 527-528, Salt Lake City, Utah, October, 2000.

[31] R. Sukthankar, T. Cham, and G. Sukthankar, “Dynamic Shadow Elimination for Multi-Projector Displays”, *Computer Vision and Pattern Recognition 2001,* Kauai, Hawaii, December, 2001.

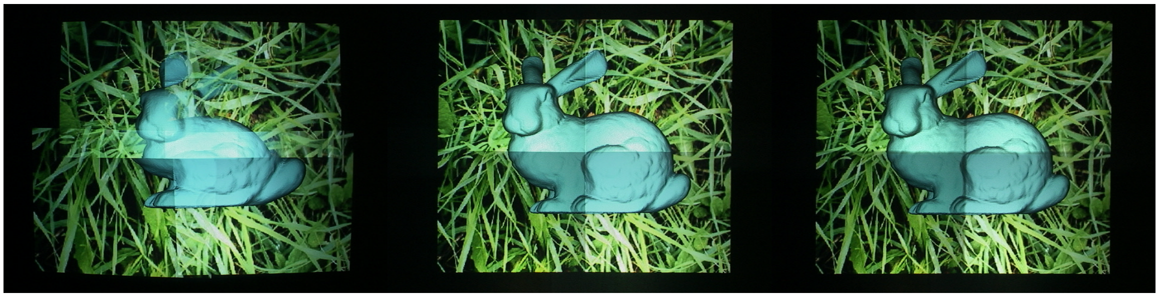
[32] B. Wei, C. Silva, E. Koutsofios, S. Krishnan, and S. North, “Visualization Research with Large Displays”, *IEEE Computer Graphics and Applications,* vol 20(4), pp 50-54, 2000.

[33] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. Brown, “PixelFlex: A Reconfigurable Multi-Projector Display System”, *IEEE Visualization 2001,* San Diego, California, October, 2001.



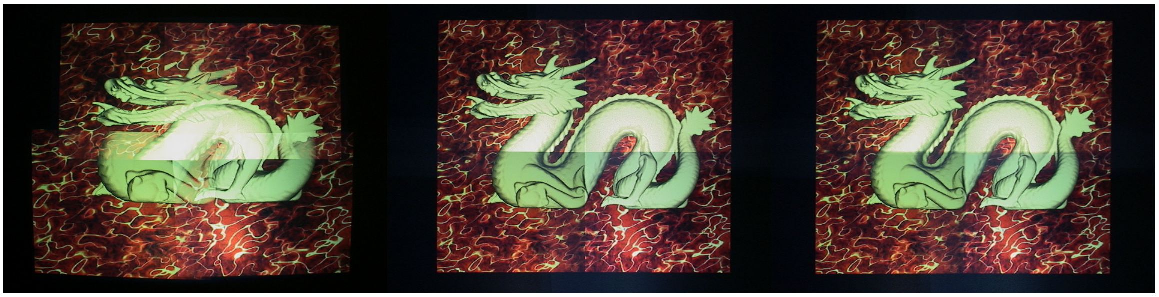
(a) No calibration, no blending. (b) After calibration, no blending. (c) After calibration and blending.

Figure 13: Cow on display wall



(a) No calibration, no blending. (b) After calibration, no blending. (c) After calibration and blending.

Figure 14: Bunny on display wall



(a) No calibration, no blending. (b) After calibration, no blending. (c) After calibration and blending.

Figure 15: Dragon on display wall.