A Survey on Projector-based PC Cluster Distributed Large Screen Displays and Shader Technologies

Munjae Song
Institute for Graphic Interfaces, 3rd Fl, Ewha-Sk Telecom Bldg. 11-1 Daehyun-dong, Seodaemun-gu, Seoul 120-750, Korea.

Seongwon Park
Yonsei University, 134 Sinchon-Dong, Seodaemun-Gu, Seoul, Korea.

YongBin Kang
Institute for Graphic Interfaces, 3rd Fl, Ewha-Sk Telecom Bldg. 11-1 Daehyun-dong, Seodaemun-gu, Seoul 120-750, Korea

Abstract - *Large screen display systems are common display systems. Especially projector-based PC cluster large screen display systems share most of large screen display system market and they are main research topics today. A lot of researchers research on cluster algorithms, computation power improvements, high performance graphic rendering technologies, high speed buses, networks, and HCIs. Remarkable research results are being published by technical leading groups. Whereas, following groups who want research on large screen display have difficulties even to build a test system. Hens there are not enough information to build large screen display systems. In this paper, we survey on projector-based PC cluster large screen display technologies that use distributed rendering. Also, this paper shows basic implementation technologies for building projector-based PC cluster large screen display systems.*

Keywords: *Large screen display, Tiled display, PC cluster, Projector –based, Distributed rendering*

1 Introduction

When the reality center facility was built in Korea in 2000, the large screen display system used Silicon Graphics Inc. Onyx2 8 CPU system and 6 BARCO LCD projectors for 3 channel passive stereo display system. In 2005, Institute for Graphics Interface (IGI) built projector-based PC cluster large screen display system with 26 projectors (Fig. 1 (a)), whereas IGI used a budget of only 10% of the budget spent for the reality center facility [26]. Even though they have different purposes and different size, the projector-based PC cluster large screen display technology has a dramatic improvement a past decade. It has three advantages, and its advantages make PC clustered systems are popular.

- PC performance is improving very fast. Current CPUs have multiple cores and acceleration functions for vector calculation, e.g. MMX, 3D NOW and SIMD. Almost every PC is equipped with high-performance graphic cards having multiple rendering pipelines, programmable Shaders, and sometimes multiple GPUs

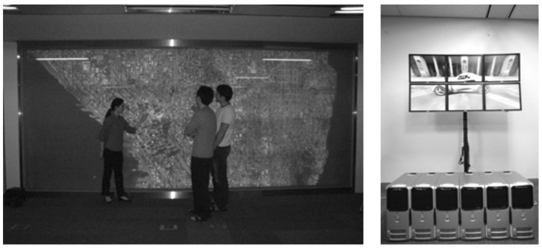
- Using COTS (Commercial Off-The-Shelf) systems lowers system cost by cutting down prices and increasing product stability. Projector-based PC cluster large screen display systems which use COTS devices are widely used because they are low-price and reliable system.

- There is a number of PC clustering open/free software which has abilities to customize and improve for building large screen display systems. Display devices have remarkable technical improvements. Early large screen display systems used CRT projectors, but CRT projectors have a problem that the refresh rate decreases when their resolution increases. Using CRT or flat panel display (FPD) to build large screen systems is called multi-monitor system. The multi-monitor systems have a serious shortcoming in each display's vessels creating discontinuity of the large screen. Because of this, users have troubles with interaction and a multi-monitor large screen does not look like one screen. Fig.1 (b) shows FPD multi-monitor large screen display system.

Nowadays, many researchers use seamless projector-based PC clustered distributed large screen display systems. Color correction technologies, edge blending technologies and NLDC technologies are important technical elements of projector-based large screen display.

We include our surveys, experiences of building large screen display systems, and implementation techniques with 3D graphics and Shader programming in this paper. We believe this paper will help research groups who want build a projector-based PC clustered distributed large screen display system. Especially, we focus on research groups that research with large screen display systems, but do not want research on large screen display system deeply.

Cluster software is surveyed in section 2. Hardware devices for large screen displays are reviewed in section 3, and software requirements for projector-based large screen displays are discussed in section 4. Section 5 shows basic implementation of major issues of large screen displays. Section 6 is summary and conclusion.



(a) (b)

Fig 1. (a) Projector-based PC cluster large screen display system in IGI. (b) Multi-monitor PC cluster large screen display system in IGI

2 Cluster software

In this section we survey cluster software, discriminating large screen display software. One is image-based streaming software, another is distributed rendering software. Table 1 shows advantages and shortcomings of two types of large screen display software. The image-based streaming software uses image data generated by applications. First, the applications generate the images which are application's result. Next, the applications send those images to cluster members. Finally, cluster members display image data with synch signal. This method does not need complex techniques like scene graph update, global timer and user inputs because it uses only image data. Whereas, image-based streaming software needs very wide bandwidth network and clients PCs use very small part of their resources. The distributed rendering software system launches applications on each slave. This method needs complex techniques like scene graph update, random number synchronizing, timer, and user inputs transmission, but the distributed rendering software uses each slave's resource. Also, the distributed rendering software divides a whole work among cluster members. Consequently, it can handle bigger and more complex jobs than the image-based streaming software. Finally, the distributed rendering software requires only ordinary networking devices because it uses small size of data to communicate.

Table 1. Comparison table of large screen display software

Type	Advantages	Shortcomings
Image-based streaming software	- Do not consider that update scene graph, timer, random number, and user input	- Dedicate network devices for wide bandwidth - Clients do not use their resource
Distributed rendering software	- Typical network device - Clients use their resource and master has small loads - Handle big and complex data with distributed system	- Difficulties of update scene graph, timer, random number, and user input

2.1 Image Based Streaming Software

There is a lot of image based streaming software, but in this section, we will only review remarkable software.

TeraVision: EVL-UIC's TeraVision has a server/client structure. Image data from TeraVision server to TeraVision clients via Giga bit network [19].

SAGE: EVL-UIC developed the Scalable Adaptive Graphics Environment (SAGE) [8] [23]. SAGE uses virtual frame buffer for large screen display and it supports 3D rendering, images, and video data from various applications. SAGE has display layout management function. Also, SAGE uses dynamic route for decreasing server load.

Juxta View: An application based on LamdaRAM[10]. Juxta View supports image panning, zooming, and pre-fetching. LamdaRAM is a network shared memory. Cluster PCs share data that in the LamdaRAM environment. LamdaRAM use 10 Giabits/sec photonic network device

2.2 Distributed Rendering Software

Even though distributed rendering has many difficulties, distributed rendering software are more popular than image based streaming software because those use ordinary network hardware. Also, distributed rendering systems use slaves' whole resource. In contrast with image based streaming software do not use clients' resources because clients show only image data from server.

Following, we review remarkable distributed rendering software, some of them are open source and others are commercial products. Open source software are flexible but commercial products do not need additional development and are stable.

Good classifications of distributed rendering software can be found in [27], [18].

CAVE: The CAVELib [15] is the most important distributed rendering software. It was made by EVL-UIC for CAVE systems. It uses shared memory via network for input data, view matrix, etc. First it supported SGI machine only, but nowadays it supports Microsoft Windows too.

VR Juggler: VR juggler [3] is an API for VR applications based on the MPI library. It was developed to make VR application free licensed and simple. VR Juggler supports IRIX, LINUX, and Windows.

Syzygy: Syzygy [24] was developed for VR applications on PC clusters. It has a lot of remarkable technical elements e.g. sound rendering, virtual devices, filters for encapsulation, Master/Slave frame work, etc. The most important feature is Phleet. Phleet is the distributed operating system of Syzygy, managing whole cluster configuration. Also, Syzygy is free software.

Jinx: Jinx [25] supports X3D format. This is very important to distributed rendering systems because most systems do not support 3D description file formats like VRML or X3D. 3D description file formats increase efficiency of contents production.

OpenSG: OpenSG [22] works with scene graphs. It distributes a scene graph to slaves for cluster rendering and uses multi-threading for high performance rendering. It supports its own file format OSB (OpenSG Binary) file and VRML. OpenSG is great scene graph architecture, but it does not provide animation nodes.

Chromium: Chromium [6] is influenced by WireGL. WireGL wraps OpenGL command and distributes commands to clients. Basic Chromium feature is intercepting OpenGL applications' command and distributing the OpenGL command to the rendering nodes via Stream Processing Units (SPU).

NAVER: Networked and Augmented Virtual Environment aRchitecture (NAVER)[17] was developed by IMRC, KIST for Virtual Reality Theater based on SGI OpenGL Performer. This API was framework for showing large screen VR contents to innumerous unprofessional audiences for entertainments.

Nova: Nova [32][29] is a commercial 3D scene graph API and supports multimedia contents on large screen display API for VisionMax. It supports VRML and X3D file format, software edge blending, and Nonlinear Distortion Correction (NLDC). It distributes scene graph for 3D rendering, images and movies. Nova has essential technologies for large screen displays, but customers cannot modify or improve Nova API.

3 Hardware devices

Projector-based PC cluster large screen display system needs special hardware devices. Projectors, graphic cards, projector stages, and screens are needed for building large screen display systems

3.1 Projector specification

Projector-based PC cluster large screen display systems require projections with same brightness and color characteristics. Perceptible difference in brightness and color can disturb audiences, since the projectors are set sided by side. Also projection regions are overlapped for edge blending. If projectors are not same color characteristic, audiences will find un-uniformity of color and they will feel the result is not good. Thus, projections have to be color corrected. To get a better result, the projectors should be 3-panel LCD projectors or 3-chip DLP projectors.

For an active stereo display, the image for the right and left eye will be made with one graphic device, and the display device must have at least 96Hz vertical refresh rate to avoid flickering. 120Hz vertical refresh rate provides comfortable viewing experience.

3.2 Graphic cards

Active stereo displays make left and right eye images alternatively. The images are swapped every vertical refresh. The user wears a LCD glasses that have a shuttering function that triggered by signal. The LCD glasses open left eye side when the left image is displayed, and open right eye side when the right image is displayed. Large screen display systems use a display devices for every projector, thus every display device must displays left images at the same time and vice versa. If the projectors cannot display same side images at the same time, users start feeling dizzy. Since active stereo use only one device to render two images per frame, the recommended frame rate has to be doubled.

High performance graphic cards are essential component for active stereo systems. Gen-lock is a mandatory function for active stereo large screen display systems. Gen-lock makes all projectors display same eye side images at the time. For example, nVidia Quadro 4000G [14] and 3D labs Wildcat Realizm [31] series support Gen-lock.

Passive stereo systems project left and right eye images through two different projectors. For passive stereo, graphic cards do not satisfy special requirements. To separates left and right eye images, passive stereo systems uses polarizing filters or color filters.

3.3 Screens

Screens as the projection surface have a high impact on the quality of the display. High gain screens supply a bright image. Passive stereo display systems have to meet specific screen properties. Typical passive stereo displays use polarizing filters for separating left and right eye images. Consequently, the screen should keep polarizing characteristics.

- Linear and circulation polarizing filters

Polarization is an important property of light. There are two types of polarization, one is linear polarization and the other is circular polarization. For example, user wears a vertical polarizing glass for left eye and the projector that displays left eye image wears vertical polarizing filter. A horizontal polarizing filter for right eye and the projector for right eye image. Linear polarize filter is cheaper than circular polarize filter, but circular polarize filters keep stereo quality when the user tilts his head.

Passive stereo usually dims the projection images, thus passive stereo needs very bright projectors or a high gain screen.

-Infitec [7]

Infitec separates color spectrum for separating left and right images. Infitec filter does not need special screen material and is not interfered by user head position either, whereas Infitec filters are expensive and clients who want use Infitec filter have to send projector to supplier for adjusting color

3.4 Projector stages

Projector states are essential hardware devices for projector-based large screen display system because projectors have to be aligned. There is a lot of software-based projector alignment researches [28] [20] [30] with image warping.

However, software-based projector alignment method wastes projection region. Thus projectors are aligned first mechanically, then fine adjustments are made with software-based methods.

4 Software requirements

Some projector-based large screen displays need several features. These features are not essential functions; however, these functions make large screen display system better.

4.1 Edge blending

Early large display systems used edge-matched display. That was easy to implement and the result was acceptable. Nowadays most large screen display systems use edge blending technology. Edge blending technology means to overlap images and control pixel brightness in the overlapped region to get a seamless display. Both hardware and software exists to realize edge belnding

- **Hardware method**

Some projectors have edge blending function or edge blending add-on E.g. BARCO [2], Christie [4], Panasonic [16], etc. Also, there is a dedicate hardware like CompactU [33]. However, hardware edge blending methods are very expensive.

- **Software method**

There are remarkable researches on software edge blending [12][11][5][13] going on. Software edge blending technology has benefits. It does not need extra budget and users can change the blending function anytime. Although software edge blending methods need extra CPU resources and graphic card resources, current PCs easily handling this relatively small computation load.

4.2 Nonlinear Distortion Correction (NLDC)

Nonlinear distortion correction function is an essential function for high-quality projector-based large screen display. When corrects lens distortion and projects onto non planar screen, projector-based large screen display systems require NLDC function [20] [21] [1].

Same as edge blending technologies, NLDC technologies are realized as hardware or software methods. Nowadays, many research approaches use software methods. Novel approach uses GPU [9].

4.3 Frame-lock

Frame-lock makes sure every cluster PCs draw images and swap those images at the same time. Between multi-projection images, time differences decrease quality of the result, even though time differences are very small. A decade before, large screen display systems used dedicated hard-wired frame-lock technology, whereas nowadays software frame-lock through network is used.

5 Basic implementation techniques

This section explains basic implementation techniques. A lot of researches on projector-based PC cluster large screen display have been published, whereas those researches are focused on just one part of large screen display systems. Hence, if following research groups want to build a large screen display system, they have to research on whole large screen display systems, even though large screen display systems are not in their major interests. E.g. high resolution video codec researchers and HCI researchers need large screen display systems but they do not want research on large screen display systems. Distributed rendering software, section 2.2, and implementations in this section will help for building projector-based large screen display systems.

5.1 System process

Projector-based large screen display systems have three stages of process. Fig. 2 shows the process flow. The first stage is generating images. The application generates images to a P-buffer or image buffer. The NLDC function uses those images as textures in 3D graphics. It function corrects distortions with measured screen shape data. The second stage is based on Shader technology. Color correction functions change color values according to color measurement data, edge blending functions are applied to images, and the edge blended images wait until swap buffer signal. The swap buffer signal triggered by frame-lock signals. Finally, the images are projected.

5.2 Nonlinear Distortion Correction

A Nonlinear distortion correction function is used for correction of screen distortions. There are many kinds of screen shapes. Toward audience, cylindrical and spherical screens are more immersive than planar screens. When images are projected onto non planar screens directly, the images are distorted. NLDC compensates such distortion on non-planar screen by applying an inverted distortion function on the projected images.

Conventional NLDC functions use a difficult warping calculation that uses a lot of CPU resources. We suggest a way to implement NLDC that exploits the power of graphics hardware found in current PCs. This method uses 3D graphics. Our method makes a plane in 3D space, and uses the image to project as texture on this plane. When the screen needs NLDC images, the plan is deformed according to the measured screen shape data. The graphics hardware is taking the job of calculating the texture mapping and thus the inverse distortion of the projected image, too. Fig. 3 shows the NLDC work process. Using 3D graphics is not an overly accurate NLDC method; however, this method does not use CPU resources, just uses simple 3D graphics functions that use graphics cards accelerating calculation.

5.3 Color correction

A Color correction function is a unique function of large screen display systems. The color correction function is used for color matching because multi-monitor systems and multi-projector systems use many display devices side by side. If each device has different color characteristics, audiences easily find non-uniformity of colors. The color correction is adjusting color values in the projected images to keep color uniformity of the display devices.

We suggest using Fragment (Pixel) Shader programs for color correction. A fragment Shader is useful tools for color correction. Typically, implementations of a color correction use a gamma value change or color value change with CPU. The Gamma value changing is not accurate. Also color value changing is heavy load to CPU. Whereas, the Shader program uses GPU.

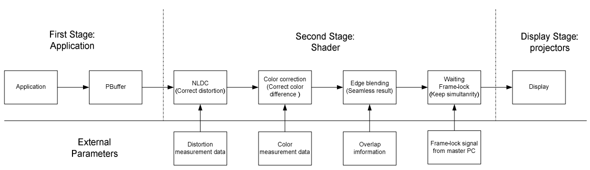


Fig 2. Flow chart of implementation process

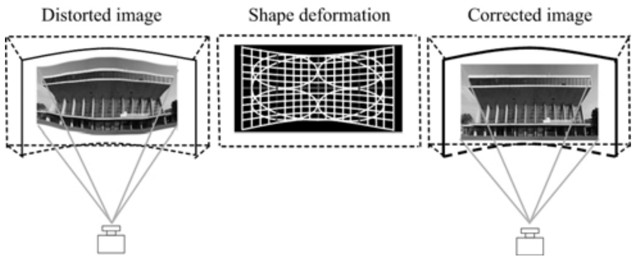


Fig 3. Flow chart of implementation process

5.4 Edge blending

Edge blending technologies are an important function to make seamless large screen display systems. For edge blending, the areas of each projection are overlapped. The overlapped regions are brighter than other areas. Edge blending techniques control brightness in the overlapped area. The overlapped regions are dimmed by the edge blending; consequently, large screen display systems get a seamless display.

We suggest using Fragment Shader program for edge blending. The Shader programs utilize the GPU, which to be found in almost every current PC. They are fast and use small CPU resources. First, make an edge blending map with CPU, Fig. 4 (a) shows an edge blending map. Use multi-texturing technique for edge blending, even though this method uses CPU at the first time for generate an edge blending map, but it does not need calculation every frame. It is helpful for increase performance. Fig .4 (b) shows the result of edge blending.

5.5 Frame-lock

Fame lock is used to ensure for time consistency between all displays devices in the system. Different complexities of scenes result in different rendering times amongst cluster PCs. It becomes a problem when large screen displays shows 3D animations or movies. Frame-lock is a method to synchronize the time of frame buffer swap. Typically, UDP network broadcast packets used for frame-lock. In closed networks, UDP is the best choice for frame-lock because UDP is faster than TCP/IP. Also, in closed networks is only very low possibility to loose UDP packets.

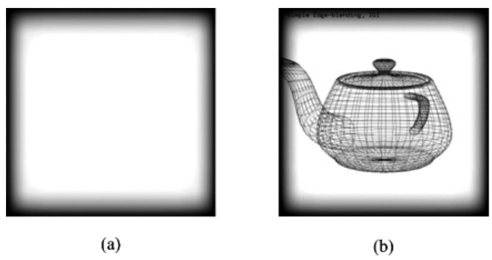


Fig 4. (a) Edge blending map, (b) edge blending result

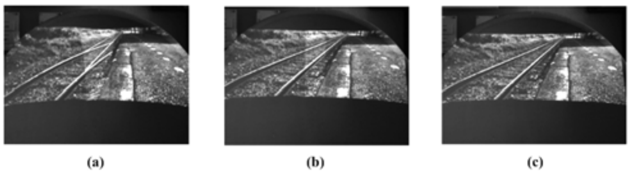


Fig 5. Project 2D image onto dome screen: (a) projector aligned only, (b) after NLDC, (c) edge blend result

6 Conclusions

Large screen display systems are common display systems today. Especially projector-based PC cluster large screen displays are most popular large screen display systems.

Projector-based systems are more useful for making seamless large screen display systems than multi-monitor systems. However, a seamless display system needs color correction and edge blending technologies. Many kinds of screen shapes are used for many purposes, thus projector-based large screen display systems have to supports nonlinear distortion correction function.

PC technology has developed fast past a decade. CPUs have multiple cores (the latest CPUs include 8 cores), and graphic card technologies are developing rapidly too. The latest graphic cards have multiple rendering pipelines, programmable Shaders, multiple GPUs on one card, and it is even possible to combine two graphic cards for one display device. Technical developments of PCs and graphic cards mean that there is no need that use dedicated graphic hardware. Consequently, PC clusters become more and more popular for large screen systems.

In this paper, we suggest very simple implementation techniques with fragment shader technology, advantage of GPU capabilities. The suggested techniques will save budget, time, and effort for building projector-based PC clustered large screen display systems.

At last this paper shows a guide to build projector-based PC clustered large screen display systems. We believe this paper is useful to understand building projector-based large screen display systems.

7 Acknowledgement

This work was supported by the IT R&D program of MIC/IITA. [2005-S-604-02, Realistic Virtual Engineering Technology Development]

8 References

[1] Jeroen van Baar, Thomas Willwacher, Srinivas Rao and Ramesh Raskar.: Seamless Multi-Projector Display on Curved Screens. In: J. Deisinger, A. Kunz, Eurographics Workshop on Virtual Environments (EGVE), Zurich, Switzerland, May 2003, pp. 281-286, ACM Press (2003)

[2] Barco, http://www.barco.com

[3] Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, Carolina Cruz-Neira.: VR Juggler: A Virtual Platform for Virtual Reality Application Development. In: Proceedings of the Virtual Reality 2001 Conference, Yokohama, Japan, March 2001, pp.215-222, IEEE Computer Society (2001.)

[4] Christie Digital Systems,Inc.,

http://www.christiedigital.com

[5] Mark Hereld, Ivan R. Judson, Rick L. Stevens.: Introduction to Building Projection-based Tiled Display Systems. IEEE Computer Graphics and Applications, Vol. 20, No. 4, pp 22-28 (2000)

[6] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D. Kirchner, James T. Klosowski.: Chromium: A Stream Processing Framework for Interactive Graphics on Clusters of Workstations. In: Proceedings of ACM SIGGRAPH 2002, San Antonio, Texas, July 2002, pp.693–702 (2002.)

[7] INFITEC GmbH, http://www.infitec.net

[8] Byungil Jeong, Ratko Jagodic, Luc Renambot, Rajvikram Singh, Andrew Johnson, Jason Leigh.: Scalable graphics architecture for high-resolution displays. In: IEEE Information Visualization Workshop 2005, Minneapolis, MN, USA, (2005)

[9] Tyler Johnson, Florian Gyarfas, Rick Skarbez, Patrick Quirk.: Multi Projector Image Correction on the GPU. In: Poster presentation at Workshop on Edge Computing Using New Commodity Architectures (EDGE), Chapel Hill, North Carolina, May 2006, (2006)

[10] Naveen K. Krishnaprasad, Venkaatram Vishwanath, Shalini Venkataraman, Arun G. Rao, Luc Renamhot, Jason Leigh, Andrew E. Johnson.: Juxtaview: A tool for interactive visualization of large imagery on scalable tiled displays. In: 2004 IEEE International Conference on Cluster Computing, pp.411-420, IEEE Computer Society (2004)

[11] Kai Li, Yuqun Chen.: Optical Blending for Multi projector Display Wall Systems. In: 12th Lasers and Electro-Optics Society 1999 Annual Meeting (LEOS '99), November 1999, pp. 281-282, IEEE Press (1999)

[12] Aditi Majumder and Rick Stevens.: Perceptual Photometric Seamlessness in Tiled Projection-Based

Displays. ACM Transactions on Graphics, Vol. 24, No. 1, pp. 118-139 (2005)

[13] Theo Mayer.: New Options and Considerations for Creating Enhanced Viewing Experiences. Computer Graphics, Vol. 31, No. 2, pp. 32-34 (1997)

[14] NVIDIA corporation, http://www.nvidia.com

[15] http://www.evl.uic.edu/pape/CAVE, Dave Pape, Carolina Cruz-Neira, Marek Czernuszenko.: CAVE User's Guide. May 1997,

[16] Matsushita Electric Industrial Co., Ltd. http://panasonic.co.jp

[17] Changhoon Park, Sang Chul Ahn, Yong-Moo Kwon, Hyoung-Gon Kim, Heedong Ko.: Gyeongju VR Theater: A Journey into the Breath of Sorabol. Presence, Vol. 12, No.2, pp.125-139. (2003)

[18] Bruno Raffin, Luciano Soares, Tao Ni, Robert Ball, Greg S. Schmidt, Mark A. Livingston, Oliver G. Staadt, Richard May.: PC clusters for virtual reality. In: IEEE Virtual Reality Conference(VR2006), pp.215-222, IEEE Computer Society (2006)

[19] Rajvikram Singh, Byungil Jeong, Luc Renambot, Andrew Johnson, Jason Leigh.: TeraVision: a Distributed, Scalable, High Resolution Graphics Streaming System. In: Cluster Computing, 2004 IEEE International Conference, September 2004, pp.391-400, IEEE CNF, (2004.)

[20] Ramesh Raskar, Michael S. Browny, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Sealesy, Henry Fuchs.: Multi-projector displays using camera-based registration. In : IEEE Visualization, San Fransisco, CA, USA, October 1999, pp.161-522, IEEE CNF(1999)

[21] Ramesh Raskar, Jeroen van Baar.: Low-Cost Multi-Projector Curved Screen Displays. International Symposium Scociety for Information Display (SID), May 2005 (2005)

[22] D. Reiners.: Opensg: A scene graph system for flexible and efficient reltime rendering for virtual and augmented reality applications., Dissertation, TU Darmstadt, Germany (2002)

[23] L. Renambot, A. Rao, R. Singh, B. Jeong, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, A. Spale, C. Zhang, G. Goldman, J. Leigh, A. Johnson.: Sage: the scalable adaptive graphics environment. In: WACE 2004, Nice, France, September 2004 (2004)

[24] Benjamin Schaeffer, Camille Goudeseune.: Syzygy: Native PC Cluster VR. In: IEEE VR Conference, Los Angeles, CA, USA, March 2003, pp.15-22, IEEE Computer Society (2003)

[25] Luciano P. Soares, Marcelo K. Zuffo.: Jinx: an X3D Browser for VR Immersive Simulation Based on Clusters of Commodity Computers. In: ninth international conference on 3D Web technology, Monterey, California, USA, April 2004, pp.79-86, ACM(2004)

[26] SGI, http://www.sgi.com

[27] Alexander Streit, Ruth Christie, Andy Boud.: Understanding next-generation vr: classifying commodity clusters for immersive virtual reality. In: 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, New York, NY, USA, pp.222–229, ACM Press (2004)

[28] Rajeev J. Surati.: A Scalable Self-Calibrating Technology for Seamless Large-Scale Displays. PhD thesis, Department of Electrical Engineering and Computer Sceince, Massachussetts Institute of Technology, (1999)

[29] VISIONMAX International LTD,

http://www.visionmaxint.com

[30] Z. Li and A. Varshney.: Calibrating Scalable Multi-projector Display Using Camera Holography Trees. In: Seventh Annual Symposium on Immersive Projection Technology (IPT 2002), Orlando, FL, March 2002, IEEE Computer society (2002)

[31] 3Dlabs Inc., http://workstation.3dlabs.com

[32] 3Digm Inc., http://www.3digm.com

[33] 3D PERCEPTION,

http://www.3d-perception.com/Issue No., Page numbers (eg.728—736), Month, and Year of publication (eg. Oct 2006).