Multi-projector seamless display system based on PC-cluster

Lingyun Yu, Cui Yu, Yigang Wang
Institute of Graphics and Image Hangzhou Dianzi University
Xiasha high education zone, Hangzhou, 310018, Zhejiang, China
lyy@hziee.edu.cn, yc@hziee.edu.cn, wangyg@cad.zju.edu.cn

Abstract

A large display system usually consists of many projectors, whose projections are merged together to form a large screen. In this paper, we discuss technologies to join multiple projections into a large display system for multimedia entertainment and virtual reality applications. In contrast to using expensive hardware-based seamlessly merging modules, we adopt a software-based seamlessly merging solution, which is used to calibrate the color brightness and the geometry in superposed regions. And in order to enlarging its application, we adapt technology to several kinds of screens (plane screen and camber screen). We use PC cluster with accelerated graphics cards to lower rendering costs. Our software merge system can run on a variety of large display platforms. The experiment shows our system obtain the satisfying results. It can seamlessly play videos with different encoding formats, such as avi, wmv, rm etc, and can seamlessly render 3D scenes in real time.

1. Introduction

Due to its large displaying area and high resolution, a large-scale displaying system could be used to provide a big audience with visual content. Moreover the screen could be enlarged to match the original size of a large object to provide the viewer with the best possible visual experience and a familiar ratio of the object. In the ideal case, we want to have a projection plane which could be extended to an infinite width and height.

Therefore a large screen created by many small screens is used more and more. At present most popular large screens have seams among its sub-displays. Absolute seamless display systems are rare and very expensive. MIT and UNC have developed software approaches to automatically "distort" the images prior to projection to match edges using calibrated cameras[1][2]. Princeton has extended the ideas to work with un-calibrated cameras[3]. This technique is used in addition to mechanical alignment, the processing of the images may introduce however a performance penalty. University of North Carolina's Office of the Future Group are exploring physical "freeform" projector placement and are handling image alignment completely in software [2][4]. Argonne and LLNL[5] use the method of adjusting projector's characteristics. There are many other papers to describe the way of seamless display system[6][7][8]. This paper presents a software adjustment of the splicing parameters which is used to calibrate the color brightness and the geometry in superposed regions. We use PC clusters with accelerated graphics cards and correct the image computationally before sending it to the projectors. This technology not only enhances the resolution of screens with different shape and scale, it also reaches the requirements for very large 3D scenes and cinematic movie-playback.

Our hardware platform consists of three projectors. Many technologies are adopted in our software system. First, we make use of DirectSound and DirectShow (which are both provided by Microsofts DirectX) to play videos and match the appropriate audio. Second, we propose a mechanism to synchronize video playing in different PCs. Third, we use 3D geometry transformation to realize geometry calibration and texture mapping to realize image blending. And we use plane and camber screens for displaying. Finally, we use OSG technology[9] to render 3D graphics.

2. PC-Cluster Displays

In order to provide exhibit halls with large visual content, the screen for the multiple projections could be a plane screen, camber screen or spheroid screen. For example, according to the vision area of the normal people, we can build a camber screen of 150°

Proceedings of the 16th International Conference on
Artificial Reality and Telexistence--Workshops (ICAT'06)
0-7695-2754-X/06 $20.00 © 2006 IEEE

(Figure 1) by putting two(or more) screens side by side.

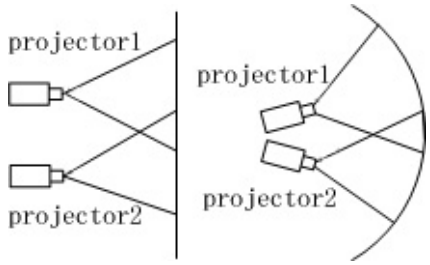

Figure 1. Plane screen and camber screen

Most of large scale displays use multi-processor, multi-piped expensive rendering engines to generate per projector output. Those expensive machines always require specialized application programming interfaces or customized software to display multi-piped imagery, existing applications typically not supported.

We can use PC cluster with accelerated graphics cards as well. Each projector is controlled by a PC, so projectors are controlled by a PC cluster. Each of computers render to a logical "tile". These distributed tiles can be efficiently reassembled to create an output image for many output devices. For example, while displaying large-screen movie, we must let each computer to display one part of the entire picture.

Although PC cluster lower rendering costs, their usage in large scale displays typically require precise projector alignment. For integrating image "tiles" into a seamless whole, several fundamental technological problems need to be addressed. These include:

(1) Frame synchronization

(2) Image and projector alignment

(3) Edge blending: color and luminosity matching and calibration techniques

3. The multi-projector seamless display system

Our system is built with low-cost commodity components: a cluster of PCs and portable presentation projectors. Synchronization, Image blending, alignment and calibration are the key major technical challenges to producing practical tiled display systems.

First we use DirectShow methods to play videos. Then we use the network to eliminate the joining of pictures from different time-frames synchronization. We use OSG technology to render 3D graphics. At last we use geometry splicing way and color blending method to solve the problem of luminance and color variations between color channels of each projector.

3.1. Synchronization

While displaying large-screen movie, we must let each computer to display one part of the entire picture. During this process, the movies picture is changing all the time and each computer also needs to renew the displaying content unceasingly. To eliminate the joining of pictures from different time-frames synchronization through the network is used.

During movie playback the times between both movies playback position are checked unceasingly. When the time interval (so called lag) between both movies time-frames exceeds the critical level of

$$|t_1 - t_2| \geq \lambda \qquad (1)$$

The frames are reordered due synchronization over the network. This prevents the movies from departing from each other. If the chosen λ is too large, then the slave machine might be delayed in playback or the display of the pictures will not seem smoothly. If the chosen λ is too small, the machines will adjust too often and seek the frame from the streams constantly. This also results in a waste of CPU time. A good choice is $\lambda = 0.1s$.

The advantage of time synchronization is that the amount of data transmitted over the network is small. The packages only carry two pieces of information, the display matrix and film show time. So this method also works in relative slow network-surroundings, as long as they don't suffer from an insufficient lag. But it is still quite insufficient, because every computer needs to read the film. The best results can be obtained by the following setup: Only one computer (the master) is responsible for reading the film. It delivers the picture information to the other computer(s) (slaves). Then all the slaves need to do is to receive the picture and display it.

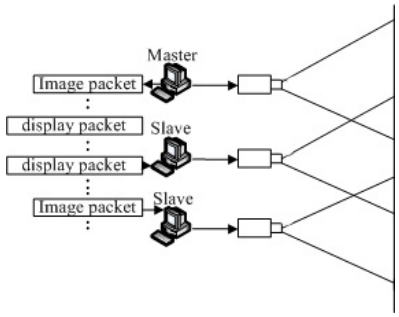

Figure 2. Synchronization of PC cluster

Proceedings of the 16th International Conference on
Artificial Reality and Telexistence--Workshops (ICAT'06)
0-7695-2754-X/06 $20.00 © 2006 IEEE

To do so, we think of using a router with one Gbit/s network speed (Figure 2). There are two kinds of packets transmitted over the network when using this method. The first kind carries the frame (the image). The second one carries the display matrix. Those packages are sent out by the master machine without a specified order, the slave machine(s) reorder the packages. If the received package is a package containing a frame, the slave machine will display the frame on a plane. If the package contains the display matrix it will rotate or enlarge the plane according to the matrix.

3.2. Image Overlapping

When tiling a surface, the tiles must be accurately aligned so that accumulated alignment errors do not propagate to the point where the overall geometry is distorted. At present most popular large screens have seams among its sub-displays. Absolute seamless display systems are rare and very expensive. If the sub-images are just side by side, there is a very bright seam between them. So they need to have an overlapping area, which makes overall-screen looks natural and fluent, just like from a single projector. But when two projectors are used side by side, the most casual observer notices that the images do not align, their colors and geometries do not match, and they are probably not the same brightness (Figure 3). Besides, the tiles overlap slightly, causing the bright lines or seams.

Figure 3. The alignment errors

It is better to keep the overlapping areas at about ten to 25 percents of the whole projection area. Edge blending techniques overlap the edges of the projected and tiled images to blend the overlapped pixel to smooth the luminance and chromaticity transition from one image to another. For example we can take a 2×1 array of projectors (two projectors next to each other). Each projection area is 1024×768 pixels, and the overlapping area takes up 256×768 pixels. The resolution of the resulting image is 1792×768 pixels.

3.3. Geometry splicing

The goal of the geometry splicing technique is to guarantee that there are no double images of the object in the projections overlapping regions. Because in software splicing, each projector projects a part of the image, the overlapping regions from both projectors should show the same content.

The plane screen

According to the Figure 4 which shows the position of right projector and the screen, we need to transfer the content of area 1to area 2.

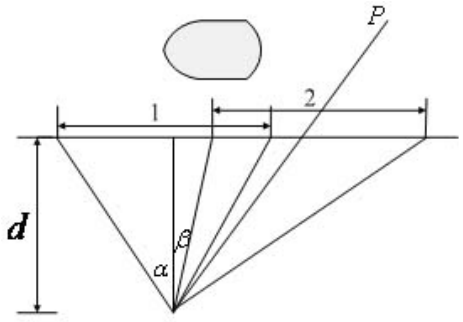

Figure 4. Geometry correction

Make the right projector for example. Here is a formula to calculate P 's position on the windows:

$$P^* = PM_v M_p M_s \tag{2}$$

and

$$M_s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{tg(\beta)}{tg(\alpha)}-1 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

In this way we can transfer the point of $x = d*tg(\beta)/(d*tg(\alpha)) = tg(\beta)/tg(\alpha)$ to -1. In another words, the content is moved by the distance of $-(\frac{tg(\beta)}{tg(\alpha)}+1)$.

The camber screen

When several projectors are all putted in the arc-center and we observe the object at the same place, the images' positions on the camber screen are completely right. But if the eye point is the arc-center, but

projector is not, we would have wrong image and position on the screen. In this way we would get double images in overlapping areas. From figure 5, we would see, if observing form the center, we output the image at the right position of the screen. But if we observe the object from different place, the images on the screen are distorted.

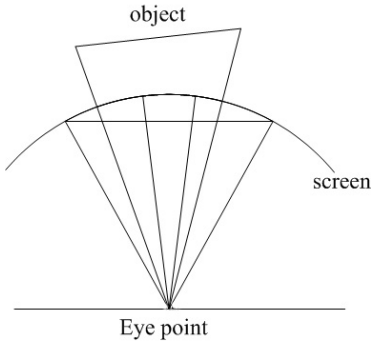

Figure 5. Observing from the center

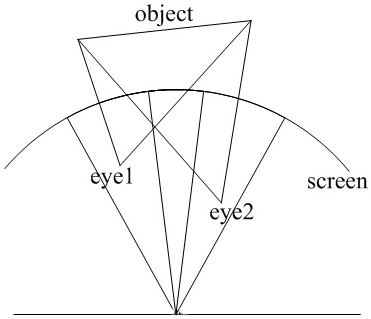

Figure 6. Observing from two different place

So we can only assume that we observe the object from the center and then set a right observing direction. But different position of outputs rely on different positions of projectors, so in order to keep the right result no matter where the projector is, firstly we need to get the projector's position (a,b,c) .

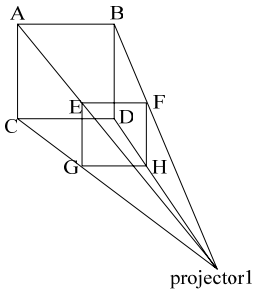

Figure 7. Get the position of projector

We can calculate the projector's position using its own result (Figure 7). We draw a quadrangle on the screen, and then display it on two plane screens. Then we can measure positions of each point and the screens, at last we can get the position of projector after calculating. The next step, we take one point from the arc-screen, and as an example its coordinate position is

$$(R \times \cos\theta, y, -R \times \sin\theta) \quad (4)$$

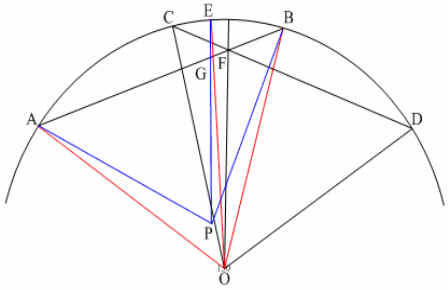

Figure 7. Correct the texture coordinates

If we display it from the center, as showed in figure 8, F is the point of E on the screen. If we display it from P (the projector's position) and want to get the right imagine like observing from the center (Figure 5), we should correct its texture coordinates. First we get E's position (G) on the planer by connecting E and P , then we need to set E 's texture coordinate at G . So what we need to do is changing the texture coordinate, like transferring the texture coordinate from F to G . And other projectors also make the similar transformation. After that, the projectors will output the images on the position of screen.

3.4. Color and luminosity matching

If the neighbor projection screens are put together, the overlaying area in the middle would leave a bright lane. So it is necessary to use a method of mixing the overlaid parts together to adjust the luminance and color variations. The mixing technique is applied to the every pixel in the overlapping region(s) (Figure 8).

Figure 8. Color and luminosity matching

Proceedings of the 16th International Conference on
Artificial Reality and Telexistence--Workshops (ICAT'06)
0-7695-2754-X/06 $20.00 © 2006 IEEE

We would like to edit the value of each pixel, by multiplying a factor. If the factor is 0, then the pixel's color becomes black, and accordingly, the color wouldn't change when the factor is 1. And because the projectors' colors and they are probably not the same brightness, in order to make the color of image much more frequent, we can edit the value of RGB of each pixel separately.

This can be realized by defining a function

$$f(x) = 1 - x^p \qquad (4)$$

with $x \in [0,1]$ (Figure 9). To adjust the brightness levels of two given pixels on both sides of the lane, multiply the right pixels brightness level with $f(x)$ (Figure 10) and the left pixels level with $1 - f(x)$.

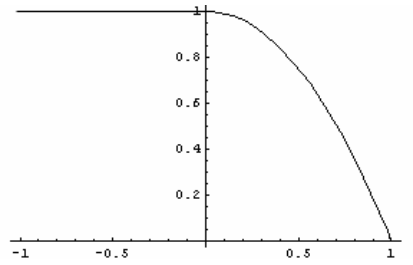

Figure 9. The function of left projector

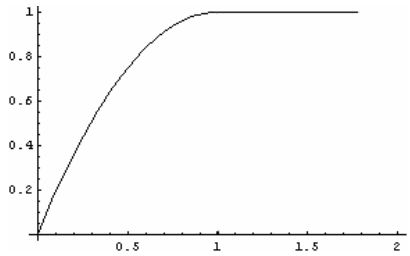

Figure 10. The function of right projector

Now the important thing is to choose the right parameter p. When $p = 1$, the accurate curvature lies on the parameter p, when $p = 2$, the mixing is quadratic. When increasing of the value of p, the curve keeps the same brightness level for a more pixels, while then lowering it very fast. So a high p is a good choice for sharp crossings.

4. OSG programming platform

The OpenSceneGraph (OSG) is an Open Source graphics toolkit for the development of high performance graphic applications such as flight simulators, 3D games, virtual reality applications and scientific visualizations. It is available on many platforms. Based around the concept of a SceneGraph, it provides an object oriented framework on top of OpenGL, giving the developer a lot of freedom, while implementing and optimizing low level graphic calls. More information on the project could be found on openscenegraph.org.

In our system, OSG technology is used to set up a scene tree and add call-back functions in some nodes of the scene tree. Then, when the picture needs to be renewed the system is searching for the certain node and will use the callback function of the node to update the frame. The callback functions have the special effects in the different nodes. (Figure 11) shows a typical partial scene tree, which is used in our setup.

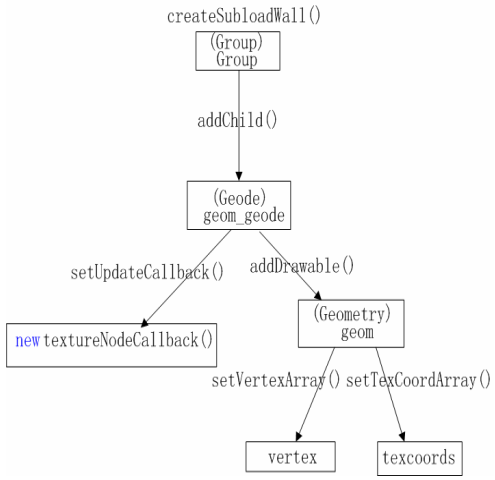

Figure 11. Part of OSG tree

Our program is realized by setting up a plane which covers the sub-screens. Firstly we set up a root node of the Group type. By using the *addchild*() command, we add a new node of the type Geode. It is also the son-node of the original node. By using the *setUpdateCallback*() command, we set the callback function for the Geode node. The callback function sets the texture of the plane. We add a geometry node directly beneath the Geode node, then set up some attributes for it like vertex and texoords.

5. Result

The experiment shows our system obtains the satisfying results no matter on plane (Figure 12) or camber screens. Using our Multi-projector seamless display system, we can watch the films just like in the cinema. We almost remove the bright lines at the

seams and display the same frame of the film at one time (Figure 13).

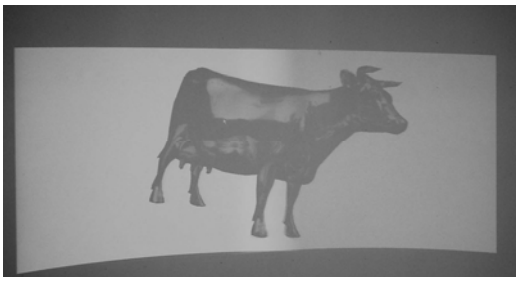

Figure 12. Observe 3D scene on camber screen

Figure 13. Use two PCs to play a film

6. Conclusion and future work

This paper introduces some general thoughts on seamless display systems. It shows how to adjust the geometry model as well as color and brightness, so that the sub-screens fit perfectly together. A way to provide a large scaled screen with a high resolution was presented. It is highly variably, since more and more projectors could be added to the setup easily. As an advantage it is cheap, since normal-priced projectors could be used.

Multi-projector seamless display systems have a big field of application and the next job would be to expand the results of the projection.

Acknowledgements: This work is supported by National Science Foundation of China (No. 60303028) and Young Scientist Science Foundation of Zhejiang Province (No.R603046).

The authors would like to acknowledge the Institute of Graphics and Image of Hangzhou Dianzi University for providing support and motivation for some of this work.

7. References

[1] R. Surati, "Scalable Self-Calibration Display Technology for Seamless Large-Scale Displays", PhD Thesis, *Massachusetts Institute of Technology*, 1999.

[2] R. Rasker, M.S.Brown, R.Yang, W.C.Chen, G.Welch and H.Towles. "Multi-Projector Displays Using Camera-Based Registration," *In Proceedings of IEEE Visualization 1999*, Oct 1999.

[3] Yuqun Chen, Douglas W.Clark, Adam Finkelstein, Timothy Housel, and Kai Li, "Methods For Avoiding Seams On High-Resolution Multi-Projector Displays Using An Un-calibrated Camera", Technical Report TR-618-00, Department of Computer Science, Princeton University, April 2000.

[4] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs, "The office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays". In Proceedings of ACM SIGGRAPH 98, pp179-188, 1998.

[5] D. Schikore et. al. "High-resolution multiprojector display walls." *IEEE Computer Graphics and Applications*,20(4):38-44, 2000.

[6] H. Chen, R. Sukthankar, G.Wallace, and T. Cham. "Calibrating scalable multi-projector displays using camera homographytrees". In IEEE Computer Vision and Pattern Recognition,2001.

[7] G. Humphreys and P. Hanrahan. "A distributed graphics system for large tiled displays." In *IEEE Visualization*, pages 215-223, San Francisco, CA, October 1999.

[8] R. Samanta et al., "Load Balancing for Multi-Projector Rendering Systems," *Proc. Eurographics/Siggraph Workshop on Graphics Hardware*, ACM Press, New York, Aug. 1999, pp. 107-116. *Chapel Hill in 1993 and 1997, respectively.*

[7] OpenScene Graph, http://www.openscenegraph.org.