

Analysis of a Scalable Multi-Projector System for Virtual Reality Environments



Fernando Teubl, Marcio Cabral, and Marcelo Zuffo
University of São Paulo, Brazil

Celso Kurashima
Federal University of ABC, Santo André, São Paulo, Brazil

Guest Editors: Luciano P. Soares and Liliane S. Machado

ABSTRACT — Virtual reality environments with multi-projector systems provide better visual quality, higher resolution and more brightness than traditional single-projector systems. Moreover, using multiple low-cost projectors is economically advantageous in comparison to an expensive high-end projector for equivalent visual performance. This article presents the research and development of a scalable multi-projection system that enables the construction of virtual reality systems with a large number of projectors and graphics computers, and that is capable of achieving a high resolution display. We demonstrate the viability of such system with the development of a camera-based multi-projector system library called FastFusion, which automatically calibrates casually aligned projectors to properly blend different projections. Our system software improves known algorithms in the literature for projector calibration and image blending. As a result, FastFusion improves system scalability and calibration reliability. In a detailed analysis of the visual performance of FastFusion in a CAVE system with three walls, eighteen projectors and nine computers, we achieved a satisfactory result for variance in geometric calibration and for graphics performance. Thus, our library is suitable for building complex projector systems and with retina resolution.

INDEX TERMS: Multi-projector System; Virtual Reality; Visual Computer

1. Introduction

Virtual reality applications with large projection area such as virtual simulation, gaming animation, and scientific visualization have been increasing in recent years. These systems are motivated by the high quality visual imagery that provides a comfortable immersive environment to the user. However, these are usually complex systems, which require high performance graphics computing and high-resolution projectors. Therefore, the need for high investments limits the deployment of such systems. The approach to lower these costs is to distribute the computational work.

Monolithic graphics supercomputers may be replaced by a cluster of commodity computers connected by a high-speed network. An individual computer has lower performance than a graphics supercomputer, however, the graphics power of clustered computers exceeds that of the monolithic machine. Each computer becomes responsible for the graphics processing of a particular group of objects. Then, the separated parts are joined together in order to form the final image. This process is known as distributed graphics processing [1].

Research on graphics clusters considers two issues: (a) how to distribute the rendering process among each computer and, (b) how to regroup the contribution of each computer to reconstruct the final image. One of the solutions implemented to solve this latter issue is the use of special libraries for distributed graphics processing, such as the

Chromium [2] and Turbo TechViz [3]. They act as a mediator - or middleware - between the graphics application and the OpenGL library. Chromium replaces the OpenGL library, in the sense that it intercepts the graphics application calls, and distributes the graphics processing among the network cooperative computers. One of the advantages of Chromium is that the graphics application code does not need to be changed, i.e. one can use proprietary application without the source code.

Another approach to promote distributed graphics processing is to consider that each computer is connected to a projector, and that all the projectors together compose a single image on the projection display. Thus, each computer naturally contributes with a smaller part of the projection image area.

Figure 1 shows the distribution of graphics processing and contents in a multi-projector environment. A multi-projection system can promote not only the distribution of application processing, but also provides an environment of high resolution and graphics performance at a reduced cost.

The major advantage of using graphics cluster with the multi-projection technique is the scalability. We only need to add new computer-projector pairs to increase visual system performance. Furthermore, in environments that require high availability, such as concert halls and museums, the cost to store equipment and spare parts is lower. Finally, advanced multi-projection systems allow arbitrary projection screens, offering new possibilities for projection. For example,

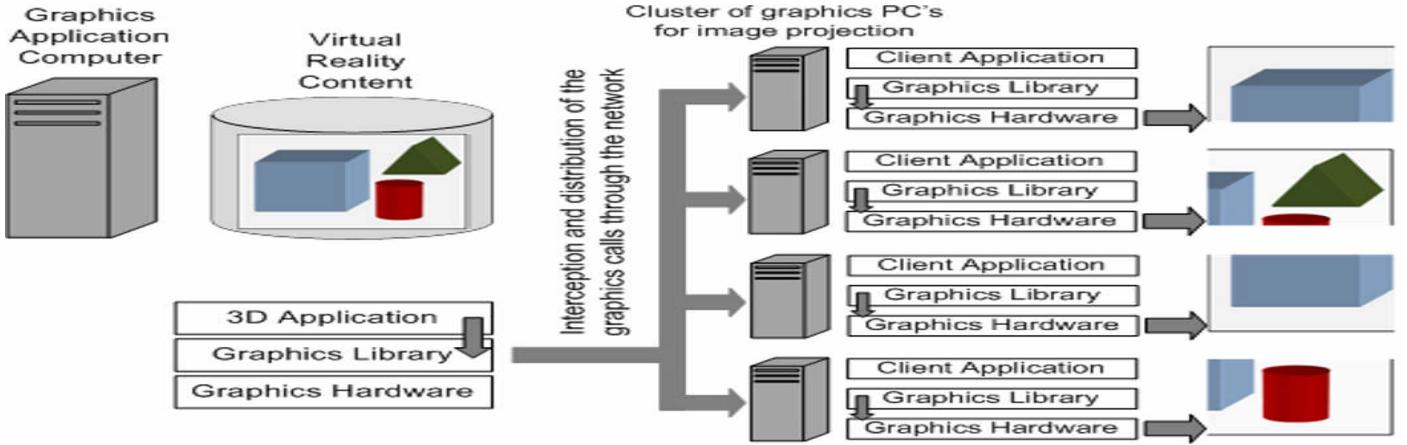


Fig. 1. Distribution of graphics processing in multi-projector environment

facades of buildings, CAVE's with igloo shape, projection in stadiums, among others.

FastFusion is a scalable multi-projection system library, which is aimed to the construction of virtual reality systems with high number of projectors and graphics computers [4]. The main feature of FastFusion is the support to a huge number of projectors and a feasible retina display solution for projection-based Virtual Environments. This article presents an analysis of this multi-projector library for building a CAVE system.

The following Section 2 brings a brief background on multi-projection systems and the related work. Section 3 presents the software architecture of FastFusion. Section 4 describes the image processing key algorithms of our multi-projection library. Part V presents the implementation and results on multi-projection environments applications. Part VI concludes this article.

2. Background and Related Work

In a multi-projection system, each projector fills a region of the total image area onto the screen. It is essentially necessary that the projected image be geometrically aligned between all the projectors. The transition from one projector to its neighbor must not show discontinuities on the displayed image.

The automatic camera-based calibration of projectors is used to align images and set the brightness and color of projectors without manual settings. The goal of automated calibration methods is to minimize the effort of the operators and provide sophisticated corrections, beyond those supported by conventional projectors.

Figure 2 shows a typical automatic camera-based multi-projection system. In this example, the system is composed of four projectors arranged in a 2x2 grid. Each projector is connected to a graphics computer. All graphics computers are interconnected to each other with a high speed local network. These graphics computers synthesize all the images to be projected onto the screen. These images must be synchronized among all computers since temporal inconsistencies between

the individual images causes wrong presentation of the global image in a given frame.

In the example of Figure 2, the graphics computers are managed by the server, which is also connected to the local network. The server includes a camera and it is responsible to manage the whole calibration process.

The basic concept of calibration is to figure out the relationship between the projectors and the image, in the form of a mapping function. Obtaining the mapping of each projector P_n to image I consists of two steps: (a) mapping each individual projector P_n to camera C coordinates; and (b) mapping camera C to image I coordinates.

Once the mapping function $P_n(x,y) \rightarrow C(u,v) \rightarrow I(s,t)$ is obtained, the image generated by each projector P_n is mapped onto image I in runtime. So, in this process all pixel coordinates of projectors P_n must be converted to image I coordinates.

Our system architecture is related to several multi-projector frameworks and algorithms that appear in the literature. In the following, we refer to important works in this field.

An early work on multi-projection was published in 1999 by Raskar et al. [6], in which they developed a multi-projector system with edge blend correction and automatic geometric aligning based on cameras. After that, Yang et al. [7] proposed a tiled display system for planar surfaces.

In 2007, Yang et al. [8] showed a flexible pixel compositor for plug-and-play multi-projector displays, which was patented in 2010 [9].

In Raskar and Van Baar [10], and Harville et al. [11] multi-projector systems with edge blend and support to curved surfaces of projections were demonstrated.

Johnson et al. [12][13] proposed a calibration framework for multi-projector that archives continuous geometric calibration for complex geometry.

Majumder has published several articles that demonstrate and propose to correct the variation in luminance and chrominance in multi-projector display [14] [15][16][17].

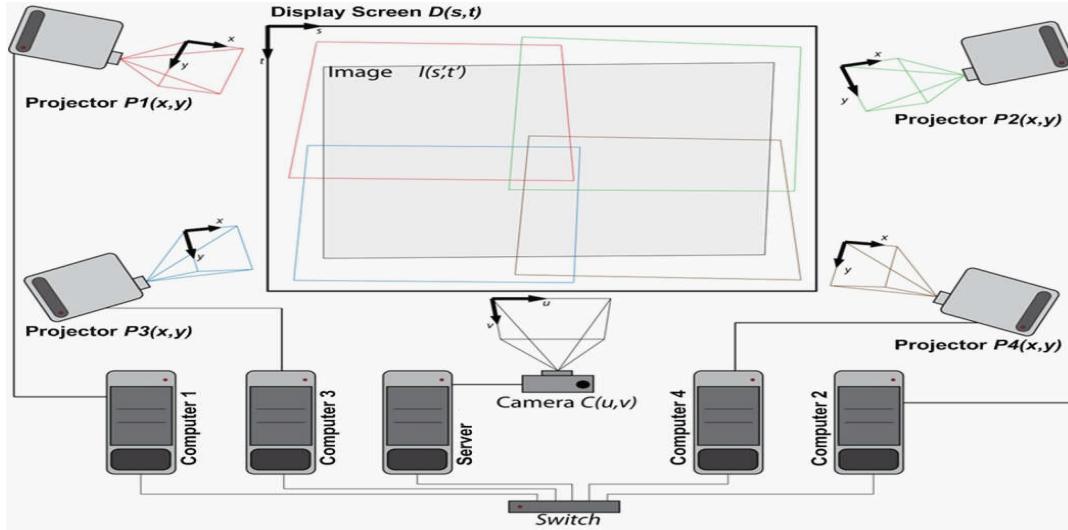


Fig. 2. Camera-based multi-projection system (adapted from [5])

Damera-Venkata et al. [18][19] presented a framework for the modeling and optimization of scalable multi-projector displays, with algorithms that can optimize the visual quality.

Sajadi et al. [20] proposed a new method that can archive an accurate geometric registration even when the field-of-view of the calibration camera cannot cover all display at a time.

A complete multi-projector system implementation of virtual reality application is reported by Yin et al. [21]. They use twelve projectors for a 360 degrees cylindrical CAVE system. Their geometric correction of projection alignment needs manual interaction from the user. And, overlapping regions are blended by a previously established linear function. In contrast, our approach uses automatic geometric calibration of projectors and blending based on actual brightness measurements.

In conclusion, a multi-projector system includes aspects on software framework, algorithms, and calibration techniques among others. Building such a system may be a complex task for the developer of virtual reality applications. The ability to support new algorithms and techniques, while offering a simple programming interface, is within the scope of our proposed architecture for multi-projector systems.

3. FastFusion: A Scalable Seamless Multi-Projector System

This part describes the system architecture of a highly scalable multi-projector systems, called FastFusion [4]. The software architecture allows scaling the number of projectors and cameras. Therefore, FastFusion is able to support tiled displays of any size and any format. In this work, our specific application is a CAVE environment, with several projectors being used for each display wall.

The next sections describe the concept and architecture of FastFusion.

3.1 CONCEPTUAL FEATURES

Virtual reality application software has a complex nature itself due to the interaction with the user, the generation of 3D content and effects, the physical simulation, among other factors. By adding a multi-projection library the application complexity is quite increased, specially concerning to development, testing and maintenance. The Fast Fusion multi-projector library simplifies this by providing a transparent graphical interface to the application. Therefore, graphics applications become exempt from the responsibility to provide support to the number and the arrangement of projectors. When application calls instances of image displays, FastFusion library creates an abstraction of the projectors, as shown in Figure 3.

Before virtual reality images are seamlessly displayed on a screen, the multi-projection library must execute a calibration step, in order to acquire the projectors displaying features. This is an independent process from the graphics image generation. Hence, the graphics application does not need to verify whether the system is being calibrated, or it is already calibrated, or it is not calibrated. And the system users can interact with the FastFusion library without interrupting the graphics application.

Additionally, the FastFusion library offers a set of functions that retrieve projection information and interact with the multi-projection library. For example, the graphics application can capture information about the contribution of each associated projector and generate the image only in the areas of interest, as well as learn about the current state of the system (it is calibrated already, it is going to be calibrated, or it is in calibration procedure). This information is useful to save computational resources, for graphics application can avoid generation of images that will never be displayed.

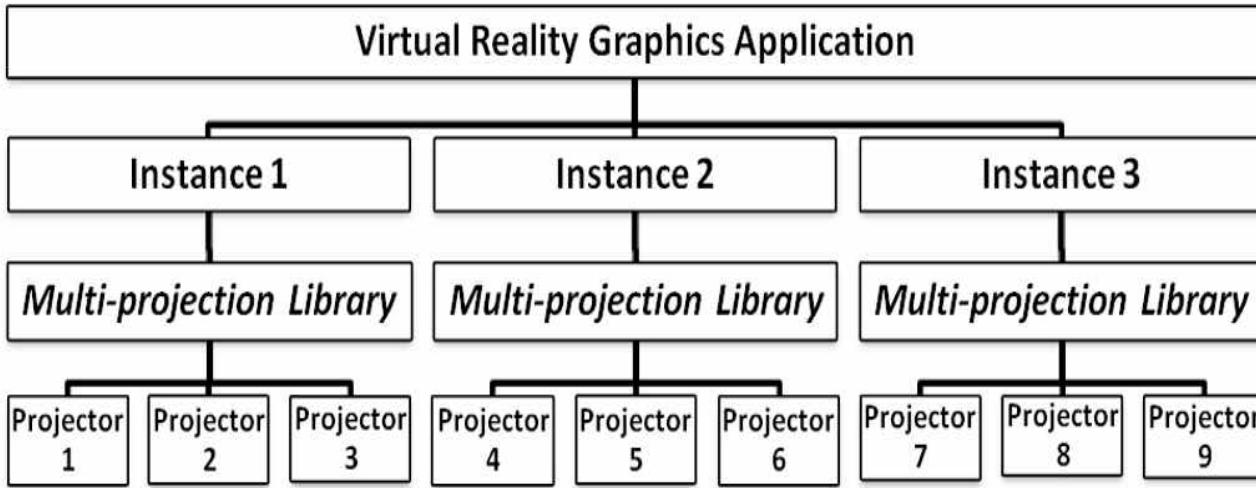


Fig. 3. Abstraction of projectors in a virtual reality application.

Another feature of the FastFusion multi-projection library is the turn-on/turn-off capability of projection portions on execution time. The instances of graphics applications can be added or removed by the virtual reality application even if the multi-projection system is already running. This feature is very important in systems with large number of computers and projectors. For example, when one graphics instance fails, the application may simply re-start this particular instance to go back to normal projection state. This increases reliability and robustness of the multi-projection environment.

3.2 HIERARCHY OF PROJECTORS USING CAMERAS

FastFusion library achieves a high degree of scalability by automatically creating the hierarchy of projectors using cameras. Therefore, FastFusion can be used in multi-projection environments that ranges from a few to a huge number of projectors and rendering computers.

The developed software library applies an automated calibration method that identifies the projectors and groups them into the cameras hierarchy. This organization is then used to reduce the steps needed to calibrate the projectors (both geometric align and color/brightness correction of projectors).

The first step of the calibration process is to identify which projectors are captured by each camera. For this task, each projector shows a previously established image pattern, allowing the cameras to search for portions of this pattern in their area of coverage. In the example shown in Figure 4(a), camera $C1$ captures images of projectors $P1$ and $P2$, camera $C2$ captures images of projectors $P2$, $P3$ and $P4$, and camera $C3$ captures images of projectors $P4$ and $P5$.

After sweeping the projectors regarded to the cameras, the system generates a hierarchy by grouping all the projectors that are inside the coverage area of each camera. Figure 4(b) illustrates the result of this hierarchy.

Once the hierarchy is completed, the system is able to identify groups of projectors that can be calibrated simultaneously. Instead of performing the calibration of each projector individually, the FastFusion library allows concurrent processes to take place: each camera calibrates their respective projectors.

The calibration sequence below illustrates all the steps executed by the FastFusion library for the example shown in Figure 4:

- 1) Calibrate projectors $P1$, $P3$ and $P5$ with cameras $C1$, $C2$ and $C3$ respectively;
- 2) Calibrate projector $P2$ with cameras $C1$ and $C2$;
- 3) Calibrate projector $P4$ with cameras $C2$ and $C3$.

This technique is called "Fast Calibration", and it can perform all the calibration in just three steps, instead of the five steps expected in a traditional calibration process. This improvement significantly reduces the overall calibration time in environments with a large number of projectors.

The information about the groups of projectors, identified in the calibration process, can be used by both the multi-projector library and the graphics application. This information is provided by the FastFusion library interface and the application can use it to improve the synchronization of neighbor projectors, the load balance or other tasks.

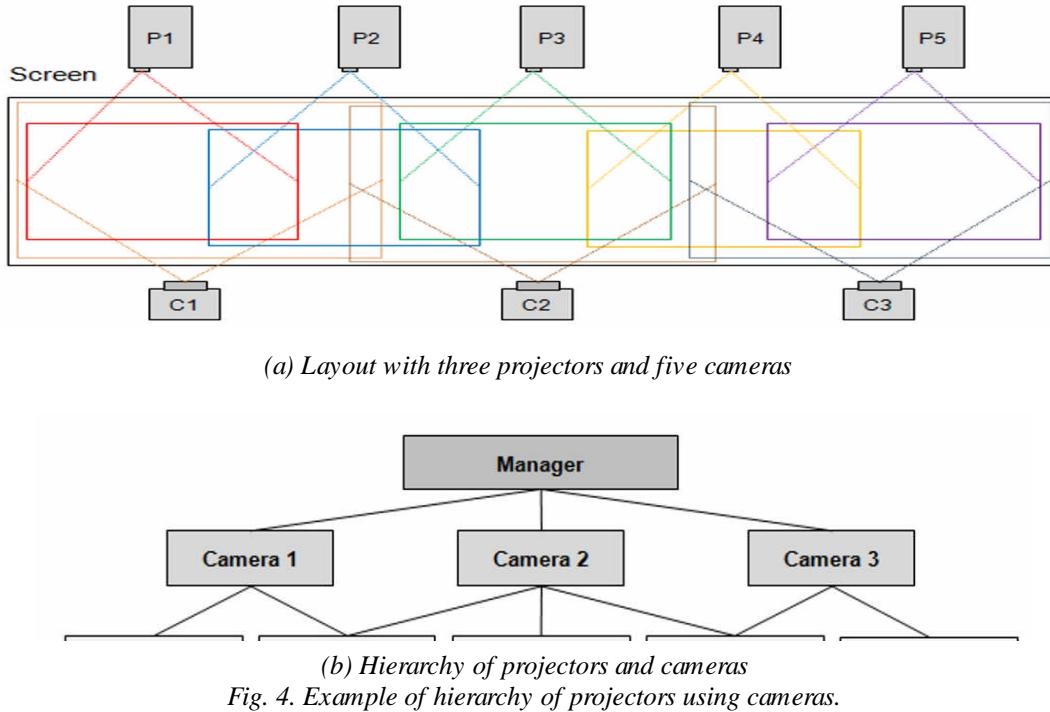


Fig. 4. Example of hierarchy of projectors using cameras.

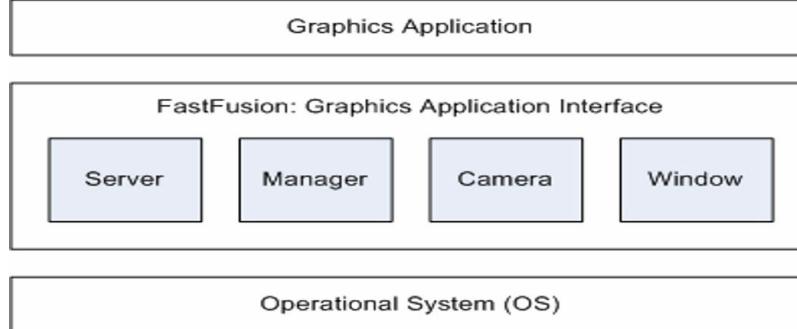


Fig. 5. FastFusion Classes for Graphics Application Interface

3.3 SOFTWARE ARCHITECTURE OVERVIEW

The FastFusion software has been developed with the object-oriented programming and it is coded with C++ language. Therefore, developers of virtual reality applications can easily use the multi-projection library by calling the available public classes from their own program code. This section presents an overview of these public classes of FastFusion library.

There are basically four public classes available in this multi-projection library, which are called Server, Manager, Camera, and Window, as shown in Figure 5. There is also an important internal class (private) called Projector. All these classes are responsible for interfacing with the graphics applications.

Figure 6 shows the internal variables and methods defined for each public class. In the following, a brief overview of these classes is presented.

- **Server:** This class is responsible for coordinating all calibration execution. Although FastFusion coordination is centered, the mathematical processing is distributed, enabling the use of several computers and projectors (see Figure 6(a)).
- **Manager:** This class allows the user's code to interact with the multi-projector library. *Manager* class can be integrated into the main application or be compiled as the separate dedicated application. This class can control all calibration process. There may be one or more managers simultaneously in the system (see Figure 6(b)).
- **Camera:** The *Camera* class is responsible for listing all devices and to registry them in *Server* class. There can be one or more instances of *Camera* class in the system. This class can be incorporated in any part of the system (see Figure 6(c)).

- **Window:** This class has to be used in the graphics application for each video output (projector). *Window* class instantiates an object of the *Projector* class for each projector of the system (see Figure 6(d)).
- **Projector:** This is an internal class (private), and it is managed by the *Window* class. The *Projector* class is responsible for performing all transformations and modifications needed to provide a continuous and uniform projected image. Each *Projector* class connects to the server and must perform all actions required by it.

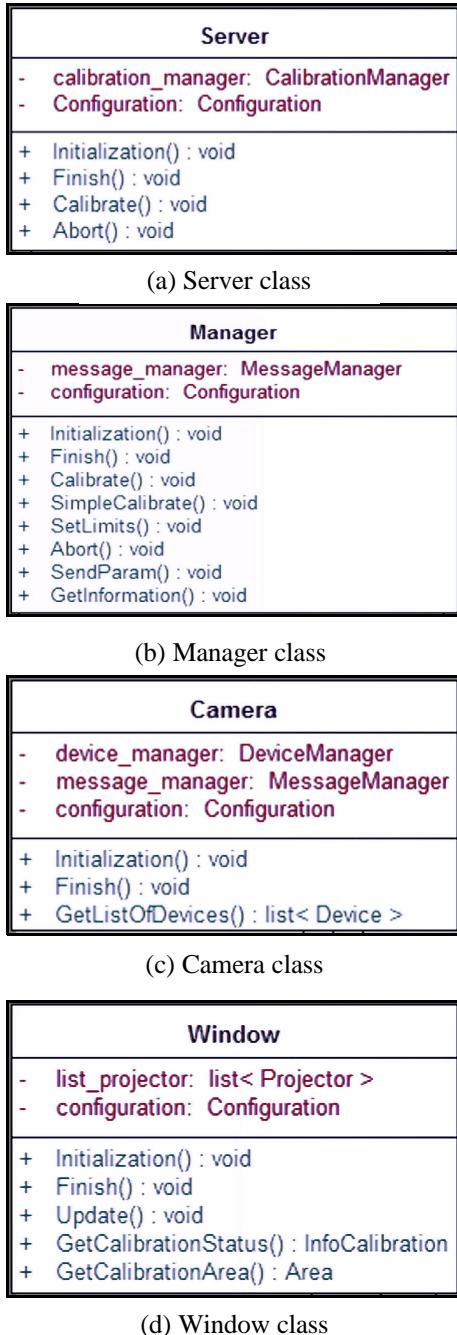


Fig. 6. Variables and methods of FastFusion public

3.4 INTEGRATION WITH GRAPHICS APPLICATION

The integration of FastFusion library with the graphics application is intended to be as simple as possible. It is necessary only three basic modifications:

- 1) To include header of the multi-projection library;
- 2) To create instance of the *display* object; and
- 3) To call the *update* method immediately before the *SwapBuffer* method.

The algorithm shown in Figure 7(a) presents a generic example of this integration. Notice the three modifications mentioned above within this code.

Alternatively, the graphics application may enhance performance by using information provided by the library, such as overlapping area, as shown in Figure 7(b). This case is a more advanced integration with FastFusion. However, it uses calibration resources for maximization of system performance.

4. Image Processing Algorithms

In this part we describe the main algorithms implemented in our scalable multi-projector system. These are techniques on the geometric calibration and the edge blending of neighbor images. Since the software architecture is composed by modules, these algorithms can be easily replaced by newer versions anytime.

4.1 GEOMETRIC CALIBRATION

The FastFusion system is based on the geometric correction algorithm demonstrated by Majumder and Brown [5, p.65]. This method was chosen because it allows a high precision calibration even with low-resolution cameras. Thus, it is possible to use conventional cameras, even in systems with large area of projection. FastFusion allows usage of others techniques to improve speed and accuracy without changing its structure.

In order to detect the blobs used in calibration pattern image, we use a technique of subtraction of images to reduce noise. This approach consists in obtaining a reference image with all projectors in black, and then subtract the calibration pattern images from the reference image. Thus, it is possible to reduce the interference of external light sources. Figure 8 illustrates this image subtraction process in a system with eight projectors (2x4 grid).

```
#include <GL/gl.h>
#include <FastFusion/Window.h>

void main()
{
    fastfusion::window ff_win( "configuration_file.conf" )
    // OpenGL initialization.

    while( 1 ) // OpenGL application main loop.

    /* SCENE RENDERING */

    ff_win.Update() // Applying Multi-projection corrections.

    SwapBuffer() // Swapping the OpenGL graphics memory buffer.
}
```

(a) Simple integration

```
#include <GL/gl.h>
#include <FastFusion/Window.h>

void main()
{
    fastfusion::window ff_win( "configuration_file.conf" )
    // OpenGL initialization.

    while( 1 ) // OpenGL application main loop.

    if( ff_win.is_calibrating() ) // Avoid image generation
        // during calibration process.
        a ← ff_win.area() // Getting overlapping area.

    SetFrustumCulling( a ) // Setting usefull overlapping area
        // to avoid unnecessary rendering.
    /* SCENE RENDERING */

    ff_win.Update() // Applying Multi-projection corrections.

    SwapBuffer() // Swapping the OpenGL graphics memory buffer.
}
```

(b) Advanced integration

Fig. 7. Examples of algorithms for integration of FastFusion

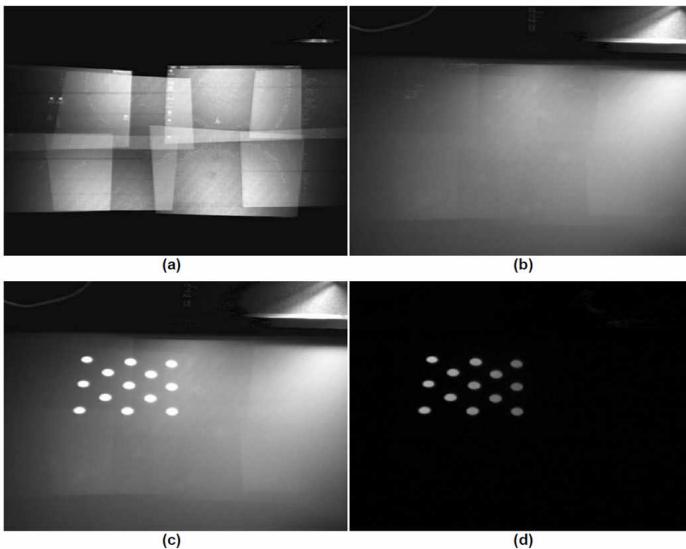


Fig. 8. Process to generate the final image of calibration: (a) initial display of all projections ate the same time, they do not need any special position alignment; (b) reference "black" image displayed by all projectors; (c) calibration pattern displayed by one projector only; (d) final pattern image obtained by subtracting image "b" from image "c".

In this example, Figure 8(a) is a picture captured by the calibration camera, and a casual arrangement of the projectors can be observed on the screen. Figure 8(b) shows the reference image, which has only black pixels. Figure 8(c) illustrates the original calibration image with the blob pattern displayed by one projector. And Figure 8(d) represents the difference between "c" and "b" images, that is actually used in the geometric calibration.

After getting the final image of calibration (Figure 8(d)), the center of each blob is calculated, and the 3x3 homography matrix for each segment is generated. Since the homography matrix is invertible, both mapping of projectoràcamera and cameraàprojector are obtained.

Using this method, it is possible to perform the calibration even when the camera is not able to capture the entire area of the projector. To estimate the projection areas that are not captured by the camera, FastFusion library extrapolates the mappings obtained from the nearest segments. This feature makes this calibration process more robust because it permits loss of blobs, and also, it may be used in environments where the camera is unable to capture the whole area of projection.

Figure 9 demonstrates the geometric calibration steps of the FastFusion. In this example, the multi-projection system is the same of Figure 8, which is composed by eight projectors in a 2x4 grid array. Figure 9(a) shows a picture captured by the calibration camera, so that we observe the random arrangement of eight projectors. Figure 9(b) shows the display result of a given image before geometric calibration is applied. Figure 9(c) demonstrates the calibration process in execution, when an array of blobs is being projected and captured by the camera. Figure 9(d) presents the image display using the complete geometric calibration.

4.2 EDGEBLEND AND COLOR CORRECTION

The edgeblend, brightness and color corrections are responsible for providing to the user an illusion of having a single projection on the entire display area. The process of brightness and color correction is performed in two steps: (a) calculation of the contribution of projectors in the overlapping areas, and (b) applying corrections.

4.2.1 Overlapping contribution

Calculation of overlap region is used to assist the process of correction of brightness and color. Our approach technique is an enhancement of the algorithm proposed by Yang et al [6] [7]. Since overlapping areas receive light from more than one projector, the individual brightness from each projector must be proportionally reduced within these areas. Using the projectoràcamera mapping obtained in previous step, actual distance between a pixel on the screen to the borders of projection area are calculated. This information is used to calculate the percentage of reduction to be applied on each original pixel value.

The contribution of each pixel is represented on 2D maps and it can be easily converted into a grayscale image. A white point represents 100% of contribution, and a black point represents 0% of contribution. Figure 10 shows the contribution maps generated for the projectors in the example of Figure 9.

Additionally, this overlapping contribution maps also incorporates the number of projectors in each position. This information is used in the process of color and brightness correction.

4.2.2 Color and brightness correction

A color and brightness correction function is applied to each pixel of the image generated by the application, providing uniformity along the final image. Such correction is multiplied to each projector pixel during all image frames. This could significantly reduce the application frame rate. In order to maximize runtime performance, FastFusion builds beforehand, during the calibration step, a correction texture map based on the color and brightness contribution maps. Then, during the application execution time, GPU uses this texture map to modulate each frame. Texture mapping is known to be much faster than CPU multiplication operations, since it is executed in GPU hardware. In summary, most calculations are performed during calibration step, so that runtime overhead is reduced.

A comparison of an image displayed with and without edgeblend correction can be seen in the example of Figure 11. Figure 11(a) shows the image after geometric correction but without edgeblend correction. The image of Figure 11(b) is the result after processing color and brightness correction. It is noticeable that this image has a smoother transition in the overlapping regions.

4.2.3 Screen area

At the end of the calibration step the image is resized to fit the camera space, because the screen limits rarely match the exact camera limits. In addition, if the screen is not planar a screen parameterization will be necessary.

FastFusion allows manual setting of the projection area through a graphical user interface. To set the projection area, the user must select four points that match the screen corners using the camera image as a reference. Figure 12 illustrates this process using the FastFusion Calibration Manager interface.

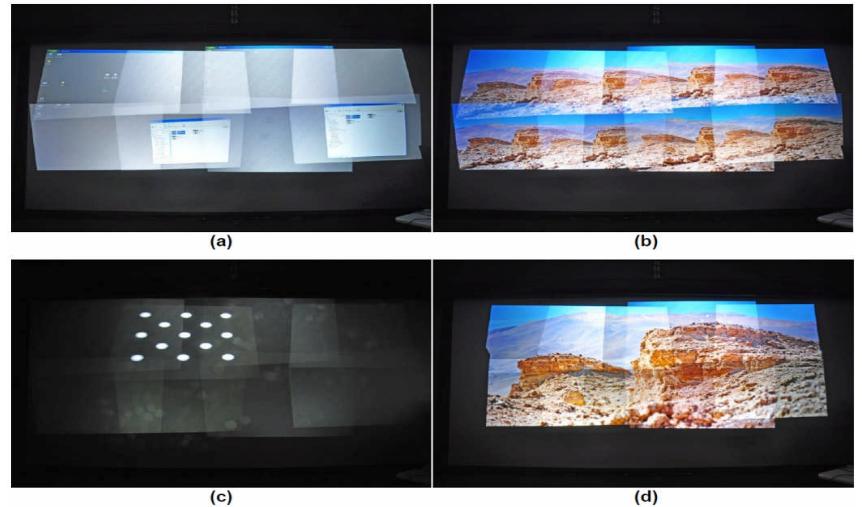


Fig. 9. Geometric calibration sequence: (a) initial display of all eight projectors; (b) projection of image without calibration of projectors; (c) execution of geometric calibration; and (d) display of image with geometric correction. Notice the natural continuity of image content.

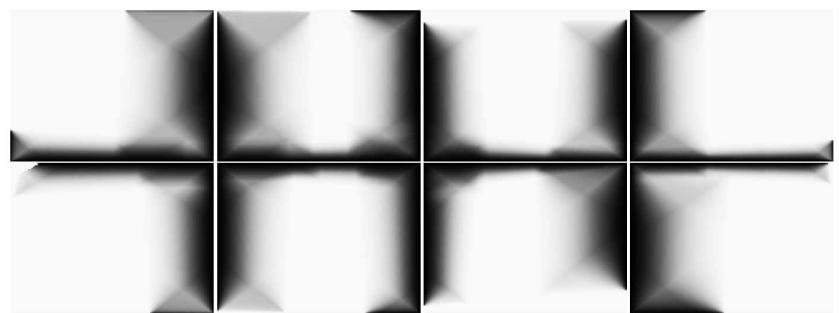


Fig. 10. An example of contribution maps In a 2x4 grid of projectors.



Fig. 11. Edgeblend correction. (a) Image with geometric correction. (b) Image with both geometric correction and edgeblend correction.

Figure 12(a) is a screenshot of the Manager interface, showing an image with geometric correction but without edgeblend correction. The image after calibration process is shown in the interface of Figure 12(b). Figure 12(c) illustrates the boundary of projection screen defined by the user, and Figure 12(d) shows the final result after calibration process and screen image adjustment.

This method has a known limitation because the camera must capture all the screen area. If a corner of the screen cannot be captured by the camera, the user cannot select the corner of the image screen. To solve this, we developed a second mode to set the screen area using markers on the projection screen. In this mode, the four corners of the screen must be covered by projectors. Figure 13 shows this procedure of adjusting the projection screen.

This method has a known limitation because the camera must capture all the screen area. If a corner of the screen cannot be captured by the camera, the user cannot select the corner of the image screen. To solve this, we developed a second mode to set the screen area using markers on the projection screen. In this mode, the four corners of the screen must be covered by projectors. Figure 13 shows this procedure of adjusting the projection screen.

Figure 13(a) shows the calibration mode by references. In this mode, four crosses are shown by projectors, while images with geometric calibration are also projected. Figure 13(b) shows the projection screen with the crosses only. These crosses can be move beyond of the area captured by the camera, as shown in Figure 13(c) and (d). After selecting the four screen corners, the relationship of camera→screen is calculated.

Figure 14 shows the final results of the screen area setting using the same scenario presented in Figure 9 and Figure 11. Figure 14(a) shows the calibration without adjustment of the screen area and Figure 14(b) shows the final result after setting the screen area.

This screen calibration process (camera→screen) has the advantage of setting the screen limits without the need to include special markers around the projection screen. Also, to flip the image projection in cases such as back projection or ceiling mount is a simple task with FastFusion calibration process.

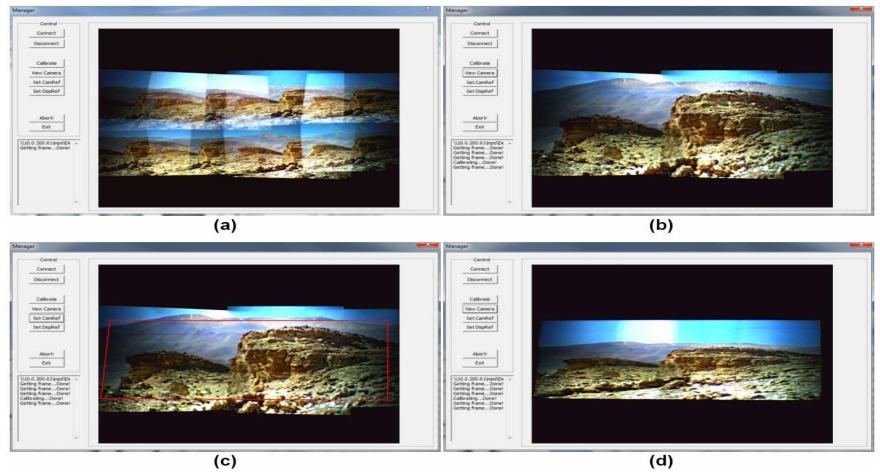


Fig. 12. Interface for screen adjustment. (a) Manager showing image with geometric correction. (b) Image with both geometric correction and edgeblend correction. (c) Boundary of projection screen defined by the user. (d) Final image

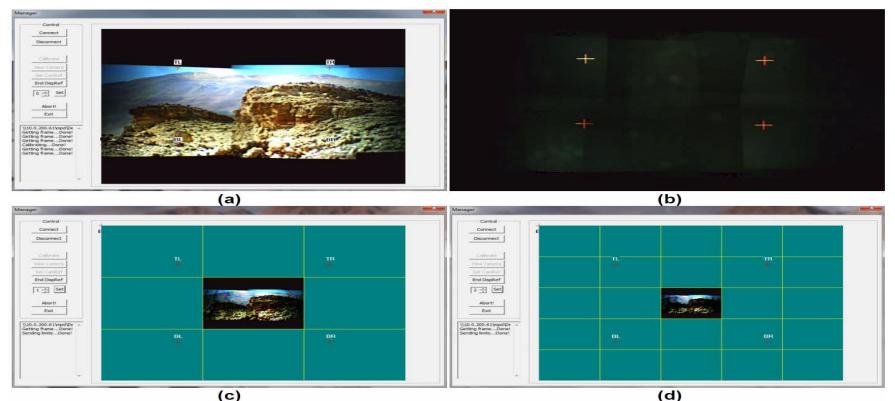


Fig. 13. Second mode of screen area adjustment. (a) Manager showing image with geometric correction. (b) Auxiliary crosses displayed by projectors onto the screen. (c) and (d) Markers position can be moved beyond camera are by the user.



Fig. 14. Screen area calibration. (a) Image displayed without screen area adjustment. (b) Result of final displayed image after setting the screen area.

5. Implementation and Results

This part presents an analysis of the visual performance of FastFusion library. The implementation aims to power a CAVE system with three walls, by using eighteen projectors and nine graphics computers.

5.1 Multi-projection in a CAVE Wall

We modified a five-walled CAVE system ($3m \times 3m \times 3m$) to test, validate and demonstrate the FastFusion library. In this experiment, only three walls are used to display virtual reality stereoscopic images. Each of these walls is powered by six stereoscopic projectors and three graphics computers. The projectors are the 3D DepthQ HDs3D-1 with 1280×720 of resolution and 2700 Lumens on 3D mode [22]. Three calibration cameras - one for each wall - are used to calibrate all eighteen projectors.

Figure 15 shows a graphics virtual model of our CAVE laboratory. Figure 16 is a picture taken at the back of a CAVE wall, and it shows the mounting of our 3×2 grid of casually aligned projectors and the three graphics computers in the bottom part. Each computer uses two video outputs, in order to drive images to two projectors simultaneously,

The geometric calibration of projectors on a screen of the CAVE wall resulted in sub-pixel precision, i.e., the geometric alignment between projectors is smaller than the size of a projected pixel.

Since the cameras do not capture two perpendicular walls simultaneously, the alignment of the corners of adjacent walls is not automatic in this version. Therefore, we improve the adjustment of screen area by precisely setting the limit of projection screen. Although this is a manual process, the result was a satisfactory alignment of the perpendicular images. The calibration of all projectors takes fifteen minutes in this experiment, which is a much faster time than traditional CAVE calibration.

Virtual reality demonstrations were developed for this CAVE environment, based on UNITY 3D tool [23]. In the integration of UNITY 3D with FastFusion multi-projector library, we used *hooking*



Fig. 15. Graphics model of our CAVE laboratory. Multi-projector systems are used in all sides of this environment. Images are back-projected on the wall screens.

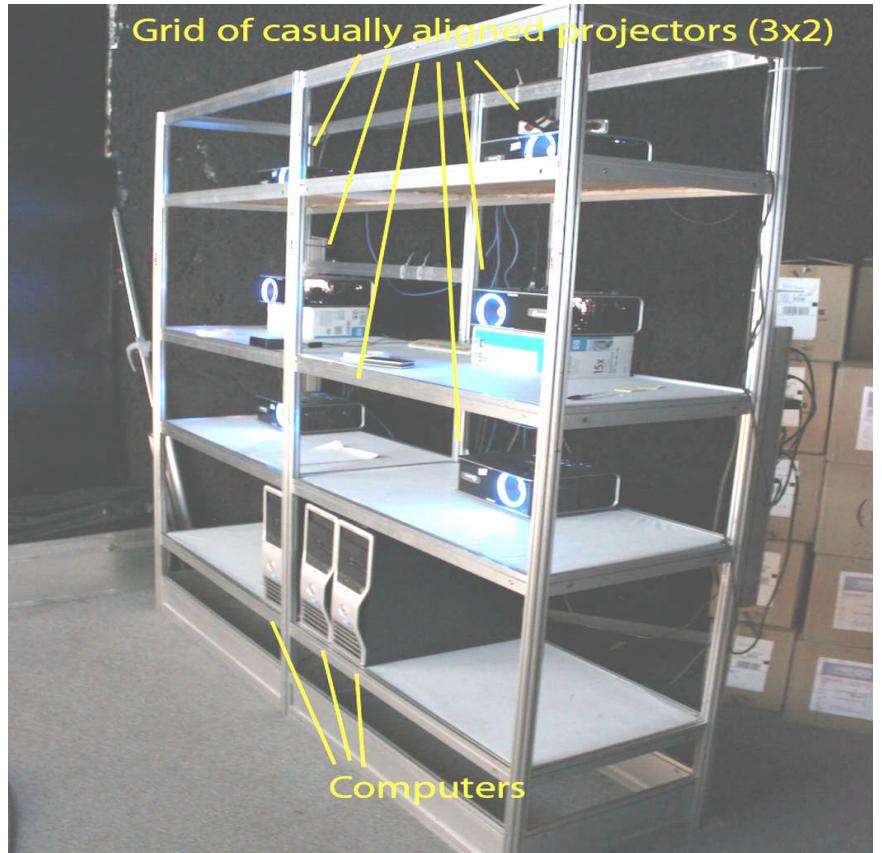


Fig. 16. Multi-projector system composed by six projectors and three graphics computers. This 3×2 projector array display images on a CAVE wall in back-projection mode.

techniques to modify the application behavior. OpenGL calls are intercepted in the execution time for applying image corrections. Therefore, we do not need to modify the UNITY 3D application to integrate it with FastFusion. Figure 17 shows the CAVE internal environment displaying a demonstration of our system running a virtual model with UNITY 3D.

This FastFusion library experiment resulted in a high quality seamless projection with a final image resolution of 2000x2000 pixels in each screen. This system configuration achieved high performance in graphics processing, while being able to open complex graphics models.

With the FastFusion scaling capability, CAVE environments powered by multi-projector systems are able to display retina resolution images.

5.2 QUANTITATIVE ANALYSIS

5.2.1 Geometric Variance

A precise geometric calibration allows a better relationship between the camera resolution and the projection area. The precision level directly interferes in the physical architecture of the system and its associated cost.

We implemented a test environment to evaluate the geometric error of FastFusion geometric calibration algorithm. The goal was to analyze the influence of the camera pixel density on the precision of the calibration procedure. A sketch of this test environment is shown in Figure 18.

The multi-projector system evaluated in this experiment was composed of two calibrated projectors (vertically aligned) and a calibration camera. The projections were adjusted to keep a maximum of overlapping area between projectors, which increases the analysis area. The test set also includes a high resolution photographic camera for capturing projected images for precision analysis.

Both projectors were a DepthQ model WXGA(DLP), with 2.500 ANSI lumens and a resolution of 1280x720 pixels. The projection on the screen occupied a rectangular area, with sides measuring 88cm x 50cm. Thus, one projected pixel had a size of 0.69mm x 0.69mm on the screen.

The photography camera was a Nikon D90 model, with resolution of 4288x2848 pixels. Since this camera covered an area of 1.0m x

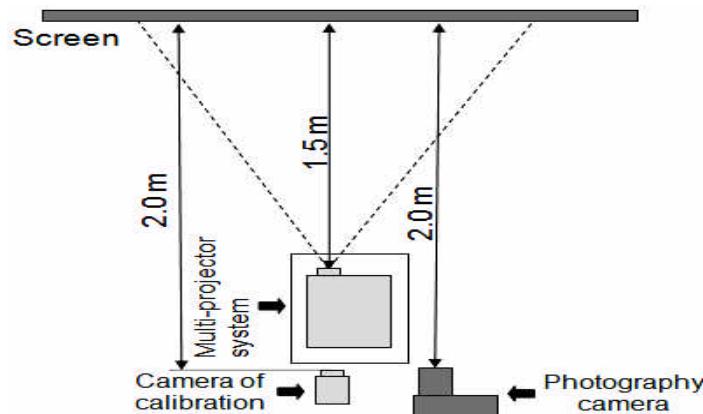


Fig. 18. Top view of laboratory environment for the evaluation of geometric variance in calibration procedure.

Table I
RELATIONSHIP BETWEEN MAXIMUM ERROR AND SIZE OF CAMERA PIXELS

Resolution		Pixel Size	Max. Error	Max. Error / Pixel Size
Horiz.	Vert.			
640 px	480 px	2.58 mm	0.00 mm	0.00%
560 px	420 px	2.95 mm	0.00 mm	0.00%
480 px	360 px	3.44 mm	0.25 mm	7.27%
400 px	300 px	4.13 mm	0.64 mm	15.50%
320 px	240 px	5.16 mm	0.98 mm	18.99%
240 px	180 px	6.88 mm	1.04 mm	15.11%
160 px	120 px	10.32 mm	1.78 mm	17.24%
120 px	90 px	13.76 mm	1.98 mm	14.39%

0.67m on the screen, each captured pixel of this camera corresponds to a square of 0.23mm x 0.23mm on the screen. Therefore, the camera pixel area is nine times smaller than the projected pixel area. This is small enough for calibration precision analysis.

The evaluation test used a range of calibration camera resolution to perform the geometric calibration. For each resolution, the whole procedure consists of the following steps:

- Calibrate the multi-projector system with FastFusion system.
- Set each projector to display a black background with five individual white pixels at these specific locations: upper-left, upper-right, lower-left, lower-right and center of screen projection. In a perfect calibration, pixels projected by different projectors must match to each other in the corresponding locations. Any misalign between these pixels means that there is a calibration error.
- Capture a picture of this projection with the photography camera.
- Measure the distance between misaligned pixels at the five locations. The greater misalignment distance is referred as the maximum calibration error. These measurements were performed by image processing algorithms.

The result of this experiment is listed in Table I. We used eight resolutions for the calibration camera, ranging from 120x90 pixels to 640x480 pixels. The calibration precision is expressed in terms of the maximum calibration error. Absolute and relative error values were computed.

It can be observed that the relative error was less than 20% in all cases. This error order is smaller than the projected pixel size. Hence, a subpixel geometric calibration can be achieved with a camera pixel size which is up to five times smaller in horizontal or vertical dimensions than the projection pixel size.

5.2.2 Brightness Variation

The distance between a projector and a given point on the screen is variable according to the light emission direction. In a CAVE with large screen dimensions this distance variation may cause non-uniform distribution of luminance along the display area. Moreover, low-cost commodity projectors usually have optical color variation, mainly in brightness across the projected images on the screen [14]. We investigated brightness variation FastFusion multi-projector system in comparison to a single-projector system.

The experimental set was composed of a 2x2 multi-projector array placed in front of the screen. Also, we have placed a single projector farther from the array so that its projected image area on the screen is approximately the same of the area used by the multi-projector array. A high-resolution photographic camera was placed behind the projectors for taking pictures of the projections on the screen.

The projectors used in this analysis were the DelphQ WXGA (DLP chip) with 1280x720 native resolution and 2.500 ANSI Lumens each, and the camera was a Nikon D90 model, with a resolution of 4288 x 2848 in raw mode. The test procedure consisted of projecting a solid white picture with edge blending enabled. Then projected image was captured from three points of views by a high resolution camera. The luminance values were obtained by converting the RGB color space to YUV color space, as in the process used in [15].

The luminance (Y) values were plotted to better visualize its amplitude variation. Figure 19(a) shows the brightness variation for a single-projection system. Figure 19(b) shows the brightness variation for a projection on the same screen area, using the multi-projection system without edge-blending. Figure 19(c) shows the brightness variations for the multi-projection system with edge-blending. Figure 19(d) shows the brightness of black-offset for the multi-projection system with edge-blending, although without efficiencies because of black image. Blue areas indicate that the left view has less brightness than the right view. Red areas indicate the opposite.

The variation in projection brightness is mainly caused by the physical characteristics of the screen. Our edge-blending algorithms demonstrated satisfactory results. However, the user can see the projectors areas in solid colors and dark images because of projectors black-offset.

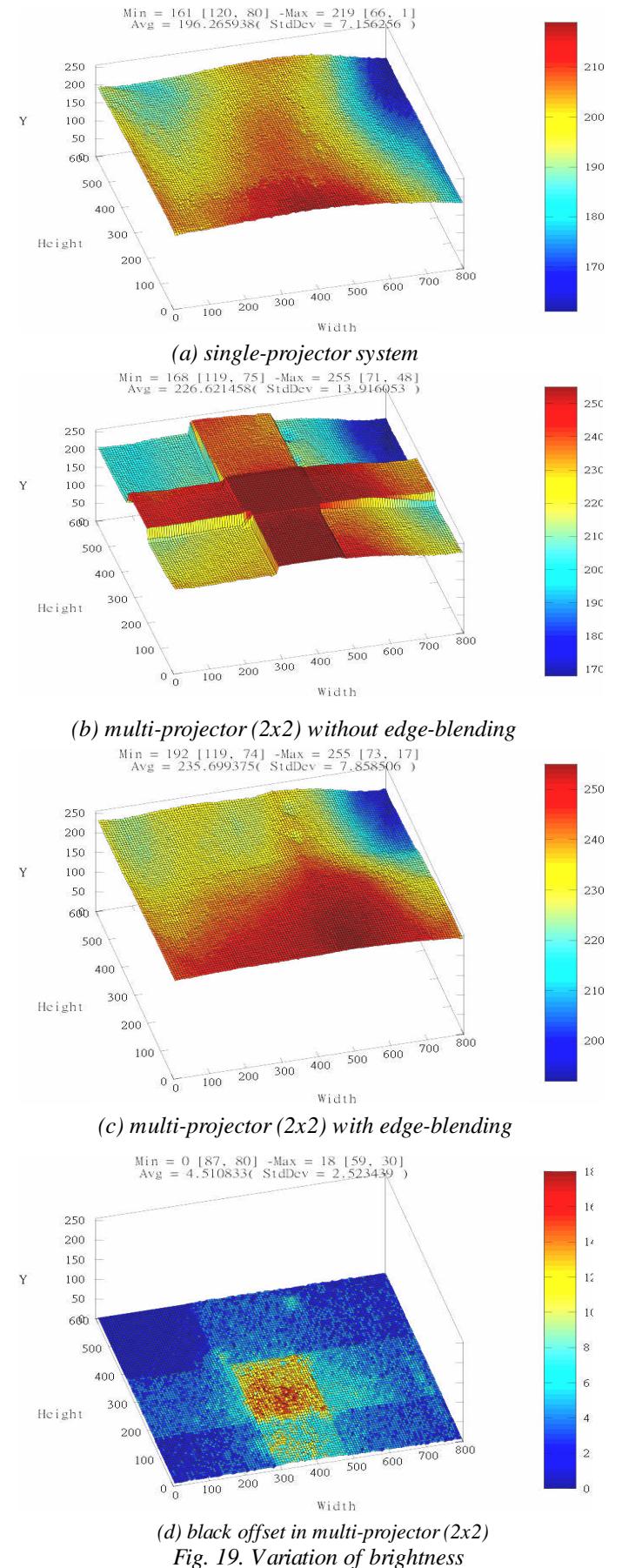


Fig. 19. Variation of brightness

5.2.3 Computational performance

Graphics computers are an important part of multi-projector systems because they perform the projectors calibration step and applies image correction algorithms when virtual reality application is being executed. The latter task were addressed by an analysis of the computational performance of FastFusion library. Test experiments were carried out on a multi-projector system composed by four projectors, one planar-screen, and a single computer. The computer model was an AMD Opteron Dual Core with four CPUs, 3.5 GB of RAM, and MS Windows XP Professional SP2 operational system. And it was also equipped with two graphics card, model NVidia Quadro FX 5600. Each projector had a resolution of 1280x720 pixels, and the total image resolution of the test applications was 2560x1440 pixels.

In order to analyze the computational impact of FastFusion in virtual reality applications, the frame rate of the graphics card output image was measured. This is based on the fact that, in this experiment, both the user application and the projection correction are performed on the same graphics card. Hence, output frame rates with and without FastFusion (FF) enabled were compared.

Table II presents the measurement results for seven different applications. Output frame rate values for applications with FastFusion are equal or lesser than without it. Frame rate with multi-projector library enabled is usually lower because the application of image corrections imposes an overhead to graphics processing unit. We achieved an overhead range of 0% to 17.8%.

Table II

APPLICATION PERFORMANCE: OUTPUT FRAME RATE

Application Name	Output Frame Rate		Overhead (%)
	Without FF	With FF	
Ogre - Demo Fresnel	62.5 fps	51.4 fps	17.8
Ogre - Demo Grass	51.5 fps	47.6 fps	7.6
Ogre - Demo Ocean	65.6 fps	58.9 fps	10.2
Ogre - Demo Terrain	71.1 fps	64.7 fps	9.0
OpenSceneGraph - osgview cesnafre.osg	60.0 fps	60.0 fps	0.0
OpenSceneGraph - osgview spaceship.osg	60.0 fps	60.0 fps	0.0
TuxRacer	80.0 fps	80.0 fps	0.0

Another performance analysis is the internal processing share of the Multi-Projector library, as shown in Table III. Percentage of processor time were measured for three internal tasks. The geometric correction process is the most time consuming part of the library with 62.5%. Then, brightness and color correction with 20.8%, and Framebuffer copy with 16.7% are lighter processes.

Table III
FASTFUSION INTERNAL PERFORMANCE SHARE

Internal task	%
Framebuffer copy (glCopyTexSubImage2D)	16.67%
Geometric correction (glCallList)	62.50%
Brightness and color correction	20.83%

6. Conclusion

This paper presents an analysis of FastFusion, a scalable multi-projection system library, for virtual reality applications in a CAVE environment. Quantitative analysis of system calibration was carried out in our virtual reality laboratory. This implementation displays multiple images in a CAVE system with three walls, by using eighteen projectors and nine graphics computers.

A comprehensive description of our scalable multi-projector system and the main processing algorithms are presented. This library enhances algorithms or techniques on the geometric calibration and the edge blending of neighbor images. As a result, the scalability and the calibration reliability of a multi-projector system are improved. Therefore, FastFusion is suitable for building complex projector systems and it is capable of achieving retina resolution.

A differential feature of FastFusion is the grouping of projectors based on camera location. This allows overall performance speedup, not only for the calibration process but also for the graphics application.

Future work includes porting this multi-projector library to other graphics libraries such as DirectX. Current version of the FastFusion uses the OpenGL library for generating graphics outputs. Since it executes image correction in the graphics card, this property provides a low impact on application performance.

ACKNOWLEDGEMENT

We thank the students and researches of CAVERNA Digital at USP who helped us during the experimental implementation and evaluation of FastFusion.

REFERENCES

- [1] Soares, L.P.; Cabral, M.C.; Bressan, P.A.; Fernandes, H.; Lopes, R.D.; and Zuffo, M.K. Management of Commodity Computer Cluster Oriented to Virtual Reality Applications. *Proceedings of the V Symposium on Virtual Reality (SVR'02)*, Fortaleza, CE, Brazil, 2002.
- [2] Humphreys, G., Houston, M., Ng, R., Frank, R., Ahern, S., Kirchner, P. D., and Klosowski, J. T. Chromium: a stream-processing framework for interactive rendering on clusters. *ACM Trans. Graph.*, vol. 21, pp. 693–702, July 2002.

- [3] Techviz. <http://www.techviz.net/>.
- [4] Teubl, F.; Kurashima, C.S.; Cabral, M.C.; and Zuffo, M.K. FastFusion: A Scalable Multi-Projector System. *Proceedings of the XIV Symposium on Virtual and Augmented Reality (SVR 2012)*. Niteroi, RJ, Brazil, 2012.
- [5] Majumder, A., and Brown, M. S. *Practical Multi-projector Display Design*. Natick, MA, USA: A. K. Peters, Ltd., 2007.
- [6] Raskar, T., Brown, M. S., Yang, R., Chen, W. C., Welch, G., Towles, H., Seales, B., and Fuchs, H. Multi-projector displays using camera-based registration, in *Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, ser. VISUALIZATION '99. Washington, DC, USA: IEEE Computer Society, 1999.
- [7] Yang, R., Gotz, D., Hensley, J., Towles, H., and Brown, M. S. Pixelflex: A reconfigurable multi-projector display system. *Proceedings of the Conference on Visualization 2001 (VIS '01)*, pp 167–174, 2001.
- [8] Yang, R., Rudolf, D. R., and Raghunathan, V. Flexible pixel compositor for plug-and-play multi-projector displays. *Proceedings of Computer Vision and Pattern Recognition*, ser. CVPR '07, Minneapolis, MN, USA, 2007.
- [9] Yang, R., and Lastra, A. Patent application title: *Source and output device-independent pixel compositor device adapted to incorporate the digital visual interface (dvi)*. Patent 20 100 039 562, 02 18, 2010.
- [10] Raskar, R., and Van Baar, J. Low-cost multi-projector curved screen displays. in *International Symposium Scoiety for Information Display (SID)*, ser. MERL Report, 2005.
- [11] Harville, M., Culbertson, B., Sobel, I., Gelb, D., Fitzhugh, A., and Tanguay, D. Practical methods for geometric and photometric correction of tiled projector. in *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, ser. CVPRW '06. Washington, DC, USA: IEEE Computer Society, 2006.
- [12] Johnson, T., and Fuchs, H. Real-time projector tracking on complex geometry using ordinary imagery. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [13] Johnson, T., Welch, G., Fuchs, H., la Force, E., and Towles, H. A distributed cooperative framework for continuous multi-projector pose estimation. in *Proceedings of the 2009 IEEE Virtual Reality Conference*. Washington, DC, USA: IEEE Computer Society, 2009.
- [14] Majumder, A., and Stevens, R. Color nonuniformity in projection-based displays: Analysis and solutions. *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, pp. 177–188, Mar-apr 2004.
- [15] Majumder, A., and Stevens, R. Perceptual photometric seamlessness in projection-based tiled displays. *ACM Trans. Graph.*, vol. 24, pp. 118–139, January 2005.
- [16] Majumder, A., and Gopi, M. Modeling color properties of tiled displays. *Computer Graphics Forum*, vol. 24, pp. 149– 163, 2005.
- [17] Majumder A., and Irani, S. Perception-based contrast enhancement of images. *Acm Transactions on Applied Perception*, vol. 4, 2007.
- [18] Damera-Venkata, N., Chang, N., and Dicarlo, J. A unified paradigm for scalable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1360–1367, November 2007.
- [19] Damera-Venkata, N., and Chang, N. L. Display supersampling. *ACM Trans. Graph.*, vol. 28, pp. 9:1–9:19, February 2009.
- [20] Sajadi, B., and Majumder, A. Scalable multi-view registration for multiprojector displays on vertically extruded surfaces. *Computer Graphics Forum*, vol. 29, no. 3, pp. 1063–1072, 2010.
- [21] Yin, Y.; Sun, X.; Zhang, X.; Liu, X.; Ren, H.; Zhang, X.; Jin, Y. Application of Virtual Reality on Marine Search and Rescue Simulator. *The International Journal of Virtual Reality*, vol.9, no. 3, pp.19-26, September 2010.
- [22] Depthq technology. <http://www.depthq.com/>.
- [23] Unity3d. <http://unity3d.com/>

BIOGRAPHIES



Fernando Teubl Ferreira received the B.Sc. degree in computer science from the Centro Universitário da FEI, Brazil, in 2004, and received the M.Sc. degree, and the Ph.D. degree in electrical engineering from the University of São Paulo (USP), Brazil, in 2007, and 2011, respectively. He is Professor at the Faculdade Impacta e Tecnologia (FIT), Brazil, since 2006. He is also with LSI-TEC, Brazil, as research engineer, since from 2003. His research interests include virtual reality, image processing and graphic computing. Email: ftf@lsi.usp.br



Celso Setsuo Kurashima (M'99) received the B.Sc. degree in electrical engineering from the Universidade Federal de Uberlândia (UFU), Brazil, in 1987, and received the M.Sc. degree, and the Ph.D. degree in electrical engineering from the University of São Paulo (USP), Brazil, in 1999, and 2005, respectively. He is Assistant Professor at the Universidade Federal do ABC (UFABC), Brazil, since its foundation in 2006. He was with LSI-TEC, Brazil, as research engineer, from 2003 to 2006. During his doctoral studies, he was Visiting Scholar at the University of North Carolina at Chapel Hill, NC, USA. From 1987 to 1999, he was with NEC Brazil, where he worked as development engineer for over than ten years. He also was a Research Student at the Tokyo Institute of Technology (Titech), Tokyo, Japan. He has experience in PCM circuits, low bit-rate speech coding algorithms, and video coding techniques. His research interests include immersive visual communications, interactive 3D video technology, and multimedia signal processing. Dr. Kurashima is member of ACM SIGGRAPH since 2001. Email: kurash@lsi.usp.br



Marcio Calixto Cabral received a PhD degree from INRIA-REVES in Sophia Antipolis, France, under the supervision of George Drettakis, in 2011. His doctoral studies focused on user-aided content creation, using techniques such as image-based relighting and interactive mesh editing with constraint-based systems. His current research is on 3D immersive interaction for content creation. Before joining INRIA, he completed his undergraduate and Master studies at University of São Paulo, Brazil, where he participated in the installation and development of applications for the first VR CAVE TM system in Latin America, called CAVERNA Digital. In 2005, he co-authored the ACM SIGGRAPH Emerging Technologies exhibit "Virtual hang-gliding over Rio de Janeiro". He is currently a principal investigator leading a team of researchers at CAVERNA Digital and a post-doctoral fellow at Polytechnic School of University of São Paulo.
Email: mcabral@lsi.usp.br



Marcelo Knörich Zuffo (M'88) received the B.Sc. degree, the M.Sc. degree, and the Ph.D. degree in electrical engineering from the University of São Paulo (USP), Brazil, in 1989, 1993, and 1997, respectively. He is Full Professor at the University of São Paulo, Brazil, since 2006, at the Department of Electronics Systems. He is also the chief of R&D activities of the Laboratory for Integrated Systems of USP. Since 2007, he is a Board Member of the Brazilian Digital Television System Forum, where the coordinates the standardization of security for digital television. In 2001, he developed the first Brazilian fully immersive CAVE environment called CAVERNA Digital. In 2010, he was General Chair of the ACM International Web3D 2010.
Email: mkzuffo@lsi.usp.br