IPT-EGVE Symposium (2007)
B. Fröhlich, R. Blach, and R. van Liere (Editors)

A GPU-Based Framework of Photometric Uniformity for Multi-Projector Tiled Display

Guodong Yuan[†] and Kaihuai Qin

Department of Computer Science & Tecnology, Tsinghua University, Beijing, China

Abstract

In this paper, we firstly propose a partial-sampling scheme to measure the intensity transfer function of a projector. Secondly, we implement the computation of luminance surface by rendering a texture rectangle on GPU. Thirdly, we generate an unified index data for all projectors. And then we compute the masks of photometric correction using a GPU based on topological consistency. Finally we integrate the masks with a GPU-based photometric correction pipeline to achieve the photometric uniformity for multi-projector display. The GPU-based framework is a combination of the GPU-based computation and the GPU-based photometric correction pipeline, which removes the bottleneck of the computation of luminance surface, attenuation mask and black offset mask. It is shown by experimental results prove that the GPU-based framework is effective and efficient.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Multi-Projector Tiled Display

1. Introduction

Multi-projector tiled system can provide high resolution and large area display, which is very useful in museums, exhibitions and cinemas etc. The main issues of using multiple projectors with seamless and smoothness display are of the geometric distortion and the photometric non-uniformity. Here we focus on how to achieve the photometric uniformity in a short time.

When multiple projectors are casually placed together, there are three regions existed: the intra-projector region, inter-projector region and overlapped region in the whole display region [BMY] [LLS]. According to the different type of the processed region, the previous methods can be classified into two categories: the blending method for the overlapped region and the photometric uniformity method for the whole display region. Further, the blending method can be classified into two sub categories: software blending [RBY*99] and optical blending method [LC99] [CJ01]. [RBY*99] precomputes the attenuation coefficient for each pixel in the overlapped region and incorporates it into the graphics pipeline using alpha blending to achieve the luminance blending in the overlapped region. However, it can not avoid the black offset issue when projectors emit the black (0) pixels. While the optical methods implement the blending using physical attenuation of lights to resolve the black offset issue. [LC99] mounts physical masks at the projector boundaries on the optical path to attenuate the light in the overlapped region. [CJ01] inserts optical masks in front of the projection lens to achieve the attenuation. However, the blending methods only work well if the overlapping projectors have similar luminance ranges. To assure this point, extensive manual labor must be applied to adjust the brightness and contrastness of projectors. Besides the overlapped region, [MS02] proposes a luminance attenuation map (LAM) method to process the intra-projection region and inter-projection region. They measure the nonlinear intensity transfer function (ITF) of each projector by using an expensive spectradiometer. Next, images are captured using a camera by the maximum luminance projected from each projector and the luminance surface is generated by adding up these images based on the geometric transformation. After achieving the common luminance response, the generated LAM for each projector is integrated with the

[†] Email: ygd02@mails.tsinghua.edu.cn

© The Eurographics Association 2007.

delivered by EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

graphics pipeline to achieve the photometric uniformity on the full display region. Since above spectradiometer is not only expensive but also uneasily used, [RGM*03] presents a method in which the high dynamic range (HDR) imaging method [DM97] is applied to measure the ITFs of all the projectors at a time using an inexpensive video camera. However, it adopts a full-sampling scheme to measure each input ranging from 0 to 255. Obviously, it takes much time to capture so great number of images.

In recent years, the raw computational power of graphics processing unit (GPU) has far surpassed that of CPU. Today's GPU has evolved into powerful and flexible streaming processors with fully programmable floating-point pipelines and tremendous aggregate computational power and memory bandwidth. With these advances, a modern GPU can now perform more functions than the specific graphics computations for which they were designed. In this paper we present a GPU-based Framework to achieve luminance uniformity for multi-projector display, which includes both the mask computation and the rendering pipleline using the GPU. Firstly, we adopt a partial sampling strategy to measure some luminance outputs for each projector, and fit the ITF of each projector using least squares curve approximation. Then we utilize the GPU to compute the data of luminance surface for all projectors based on the HDR technology. Finally, we utilize a GPU to generate the uniform luminance masks for each projector, and we integrate them into the GPU-based rendering pipeline to achieve the photometric uniformity.

Our main contributions are as follows:

1) As opposed to the full sampling technique [RGM*03], parts of samples ranging from 0 to 255 are used to measure the ITFs of projectors. Obviously, there are less amounts of images to be captured in our method.
2) Both the evaluation of the luminance surface and the generation of masks to achieve photometric correction are implemented on a GPU, which greatly reduces the time cost of setting up a casual multi-projector system.
3) The Integration of GPU-based computation of photometric uniformity and GPU-based pipeline of seamless display makes it practical to build up a low-cost multi-projector display system in a casual environment.

In Section 2, we give the overview of our GPU-based framework for achieving the photometric uniformity. In Section 3, we discuss the implementation of each step in the framework and provide the experimental results. We also present the performance comparison of the GPU-based computation with the CPU-based one. Finally the conclusion and future work are described in Section 4.

2. Framework Overview

Geometric registration is a necessary step to achieve seamless multi-projector display. Similar to [RBY*99] [BS], we use a single un-calibrated camera to capture the geometric arrangement of tiled projectors. Let us assume that the camera is of resolution $W_c \times H_c$ and each of n projectors is of resolution $W_p \times H_p$. We assume that the camera image plane is parallel to the planar screen plane. So the geometric registration can be performed by warping the coordinate information of the fiducials between the camera's image space (camera space) and the projector's framebuffer space (projector space).

The GPU-based Framework of photometric uniformity consists of four stages:

1) Measuring the Intensity Transfer Function of a Projector. The intensity Transfer Function (ITF) shows that the output luminance of projector is nonlinear response when the RGB channels of projector vary from 0 to 255. Besides of expensive photometers as measure devices, high dynamic range (HDR) image method can be adapted to measure the ITFs of projectors. We present a partial-sampling scheme to measure the luminance responses of specific sample points. And then achieve the ITF based on least squares B-Spline curve approximation.
2) Evaluating the display luminance surfaces using a GPU. If the inverse of the ITF is applied to the output image of a projector beforehand, the luminance response of a projector meets the linear relation. By the linear response we can measure only the luminances of the minimum (black) and the maximum (white) input using the HDR technology. We realize the evaluation of black and white HDR data on a GPU instead of a CPU, which highly improves the computation performance.
3) Generating Masks for photometric uniformity using a GPU. On the precondition that linearity is achieved, both the attenuation mask and the black offset mask in the projector space can be computed according to the relation of luminance uniformity [RGM*03]. For any pixel in a mask, we calculate which triangle the pixel locates. Since the triangle meshes in projector space have the same topology, so we can organise this information into a 2D texture of resolution $W_p \times H_p$, and take advantage of the 2D texture to implement the computation of masks on a GPU for each projector.
4) Photometric Correction Pipeline on GPU. For each projector, we sequentially integrate the attenuation mask, the black offset mask and the inverse of the ITF into the graphics pipeline using Cg. Due that the two masks are achieved using a GPU, so it is unnecessary to exchange the texture data between the CPU and the GPU.

2.1. Measuring the Intensity Transfer Function of Projector

Let us denote the relation between the projector's input I and its luminance response E as $E = f(I)$, where I ranges from 0 to 255. [RGM*03] measures the ITF using a full-sampling technique, which has stepped the input of projectors from black (0) to white (255) and taken one HDR image for each input. Thus the resulting ITF curve can be achieved by the 256 HDR samples. However, the noisy curve is not usable unless it is smoothed by a gaussian filter. It is obvious that the full-sampling technique brings on a large scale of captured images. In addition, the smoothed ITF curve is not editable. To reduce the amounts of captured images, we adopt a partial-sampling technique to take the HDR images of a projector's inputs. We select $m+1$ samples in the range from 0 to 255, and take $n+1$ images with different exposure time Δt_j for each sample. Using the HDR technology, we can achieve an array of 2D data (I_i, E_i). Then we apply the least square method to approximate the ITF curve using a B-Spline curve [LP95].

For $m+1$ E_i $(i=0,1,..,m)$, we seek a pth-degree B-Spline curve,

$$E(u) = \sum_{j=0}^{N} B_{j,p}(u)Q_j, u \in [0,1].$$

Satisfying that

1) $E_0 = E(0)$ and $E_m = E(1)$;
2) $Min(\sum_{k=1}^{m-1} |E_k - E(\overline{u}_k)|^2)$, where the $\overline{u}_k$ is the precomputed parameter value.

We can achieve a p-th B-Spline curve with $N+1$ control points Q, which is the approximation of the exact ITF curve. We will show our measurement results in Ssection 3.

2.2. Evaluating the Luminance Surface Using a GPU

Before generating the attenuation mask and the black offset mask for each projector, we need to evaluate the luminance surface by capturing the black and the white luminance of the entire display when turning on all projectors simultaneously. Let us take pictures for the luminance display with $n+1$ different exposure times Δt_j. We can get two series of surface images, one is the series of the black surface images and the other is the series of the white surface images. Different from previous methods, we extend the computation of the luminance surface onto the GPU. The details are as follows: for each pixel Z_{ij} in the exposure images, its irradiance value E_i meets with the Equation

$$\ln E_i = \frac{\sum_{j=0}^{n} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=0}^{n} w(Z_{ij})} \qquad (1)$$

where

$$w(Z) = \begin{cases} Z - Z_{min}, & Z \le \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - Z, & Z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

and g is a smooth and monotonic function for the digital camera, which can be achieved according to [DM97] in advance.

Since each captured image has the resolution of $W_c \times H_c$, the evaluation of the luminance surface has the complexity of $O(n^2)$. However, we accelerate such computation using the parallelism of a GPU instead of the serialism of a CPU. We make the discretion of the g function as one 1D texture. And we make each series of exposure images as 2D textures of resolution $W_c \times H_c$ separately. Thus, we can use the OpenGL render-to-texture technique to implement the computation of the luminance surface in the camera space.

2.3. Generating Masks for Photometric Uniformity Using a GPU

The geometric calibration is necessary to achieve the photometric uniformity. As shown in Figure 1, the $m_r \times n_c$ fiducials in the projector space are projected onto the planar screen by each projector. The coordinate of each fiducial in the projector space is denoted by $F(x,y)$. Then a single camera is used to observe the output of each projector in turn. The fiducial coordinate $C(u,v)$ in the camera space can be achieved using an image-processing technique. These fiducials of each projector can be separately organized into triangular meshes in the projector space and in the camera space. The triangular mesh in the projector space is denoted as $projectorTM$ and that in the camera space is denoted by $cameraTM$, which can be expressed as follows:

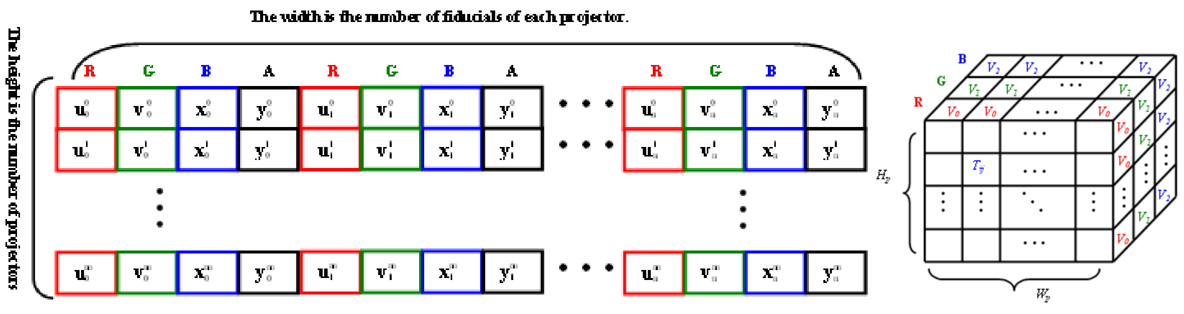
$$\begin{cases} projectorTM = \{projectorTri | (F_{s,t}, F_{s+1,t+1}, F_{s,t+1}), \\ \qquad \cup (F_{s,t}, F_{s+1,t}, F_{s+1,t+1})\} \\ cameraTM = \{cameraTri | (C_{s,t}, C_{s+1,t+1}, C_{s,t+1}) \\ \qquad \cup (C_{s,t}, C_{s+1,t}, C_{s+1,t+1})\} \end{cases},$$

where $0 \le s \le m_r - 1$ and $0 \le t \le m_c - 1$. We adopt the inscribed rectangle instead of the bounding box of all fiducials, as the display region, as shown in Figure 1. According to the area coordinate theory, the geometric calibration can be easily achieved. The details can be referred in [YQH].

For any given pixel P_{ij} in the masks, the attenuation coefficient α and the black offset coefficient β are computed using the Equation

$$\alpha = \frac{w_{min} - b_{max}}{w_r - b_r}, \beta = \frac{b_{max} - b_r}{w_r - b_r} \qquad (2)$$

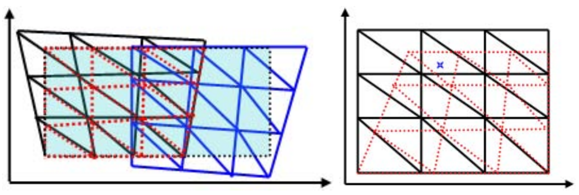
where w_{min} is the minimum value in the white luminance surface, b_{max} is the maximum value in the black luminance surface [RGM*03], w_r represents the response in the white luminance surface where the given pixel P_{ij} is transformed from the projector space to the camera space, and b_r represents the response in the black surface. As shown in Figure 1, if both $projectorTM$ and $cameraTM$ are ordered sequentially by the same rule, such that the triangle index is ordered from left to right and from bottom to top, we can find an important fact that the two meshes in the projector space and in the camera space have the same topology for any given projector. It means that for any given pixel P_{ij} in the projector space, once its triangle index k in the projector space is known, its corresponding triangle in the camera space can be directly accessed by the index k.


The width is the number of fiducials of each projector.
The height is the number of projectors
R
G
B
A
H_p
W_p
T_{ij}
V_0
V_1
V_2

(a) (b)

Figure 2: *(a) shows the data organization of fiducials for all projectors. (b) shows the data organization of triangular index data*

Although the triangle index data for all pixels can be calculated according to the geometrical transformation between the $projectorTM$ and the $cameraTM$, it is tedious and costs much time. Fortunately, due that any $projectorTM$ has the same regular geometry. We can generate the triangle index for each pixel in the projector space by rasterizing the $projectorTM$ in advance. As shown in Figure 1, It is assumed that there are $m_r \times n_c$ fiducials in the projector space. Then we can achieve $(m_r - 1) \times (n_c - 1)$ rectangles, and each rectangle consists of two triangles. For each rectangle, we can transverse the pixel in it and determine which triangle this pixel is located in. Thus we achieve the triangle index data for all pixels in the projector space. The performance of generating the data will be provided in Section 3. As we know, the coordinate of each fiducial (x, y, u, v) is a quadrilateral element. The coordinates of all fiducials of each projector are arranged along the row direction. Thus, the coordinate data can be organized as a 2D rectangular texture of size $N_f \times N_p$, where N_f is the number of fiducials of each projector and N_p is the number of projectors. Meanwhile, the triangular index data of each pixel in the projector space can also be organized as a $W_p \times H_p$ 2D rectangular texture. Figure 2 shows the arrangements of the two textures.



(a) (b)

Figure 1: *(a) is the meshes of 1 $\times$ 2 tiled mode in the camera space. The shadow region is the display region. (b) is a mesh in the projector space. The red mesh in (b) is achieved by warping each vertex in the red mesh in (a) based on the area coordinate theory.*

We render a $W_p \times H_p$ texture rectangle on the graphics pipeline. For each fragment, the steps of GPU-based computation are as follows:

1) Looking up the triangular index texture according to the texture coordinate and achieving an triple (V_0, V_1, V_2);
2) Using V_i to lookup the vertex texture and getting the a tetrad coordinates of each triangle's vertice (x, y, u, v);
3) Calculating the triangular area coordinate (c_0, c_1, c_2) of the current fragment in the projector space;
4) Calculating the corresponding coordinate (u, v) in the camera space based on the area coordinate theory;
5) Using (u, v) to look up the black and white surface textures and getting the w_r and b_r values;
6) Computing the attenuation and black offset coefficients using the Equation 2.

Combining the Multiple Render Target (MRT) and the Framebuffer object (FBO) extensions, the computation can be completed by one-pass rendering. Meanwhile both the attenuation texture and the black offset texture are stored in the texture memory on the GPU, which can be used directly in the GPU-based rendering pipeline.

2.4. Photometric Correction Pipeline On GPU

After the computations of the attenuation mask and the black offset mask are completed, we can integrate the two masks with the inverse of the ITF of each projector to implement the photometric correction pipeline on a GPU. Figure 3 illustrates our GPU-based framework to achieve photometric uniformity for the multi-projector display. Compared to previous photometric correction pipeline [RGM*03] [MS], our pipeline is incorporated with the GPU-based mask computation, which brings some benefit on the efficiency of the pipeline initialization in terms of binding the mask textures without the swap between the texture memory of GPU and the memory of CPU.

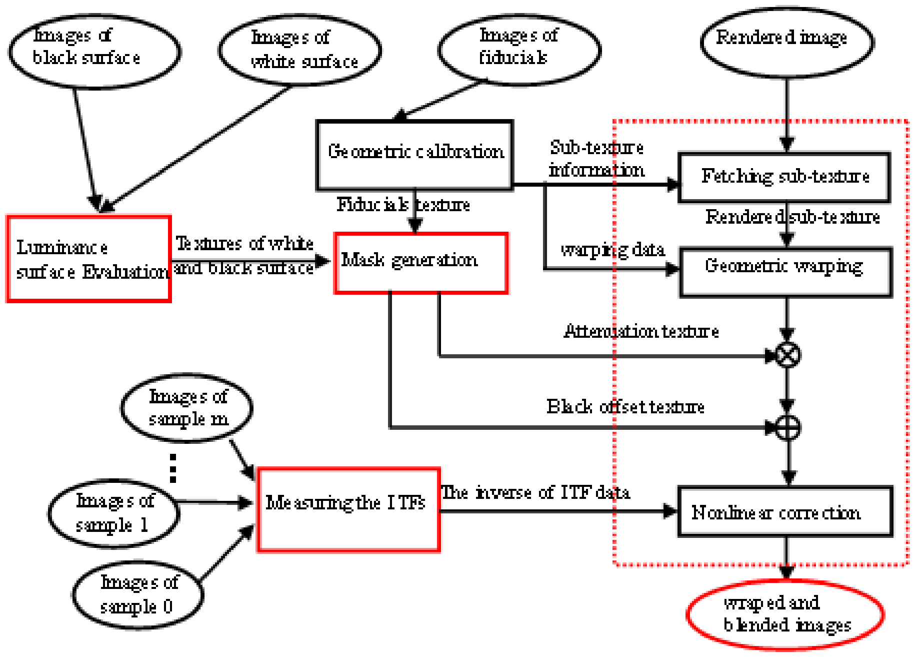


Figure 3: *The block diagram of GPU-based Framework. The part in the red-dotted-line rectangle is the photometric correction pipeline.*

3. Implementation & Experimental Results

We implement the GPU-based framework of the photometric uniformity for the multi-projector display on a display wall made up of 2×2 array of four projectors. In this section, we discuss the implementation details of our GPU-based framework and analyse the performances of the ITF measurements, the evaluation of the luminance surface and the generation of masks. Finally we provide some experimental results to show the rightness and efficiency of the GPU-based framework.

We select an array of samples [0,4*i,255] ($i = 1..63$) as the input of all projectors and the exposure time varies in the stops [6,4,2,1,0.6,0.4,0.2,0.1,0.05,0.02,0.01,0.005] with the fixed aperture. Thus we need to take the total 65×12 images to measure the ITFs of all projectors. The approximation of each actual ITF can be achieved according to our B-Spline fitting method. Figure 5 shows our experimental result. However, due to the limitation of our conditions, we can not provide the actual ITF measured by a expensive spectradiometer. Even if the partial-sampling method is adopted instead of the full-sampling one, the number of taken images is too large to be done in a short time. Fortunately, the shape of the ITF of a projector does not change significantly over time [MS02], so this measurement can be done offline. Once the ITF of a projector changes, we can modify the control points of the B-Spline curve to adjust the approximation of the ITF.

Table 1: *Performance of Luminance Surface Evaluation (ms)*

	800×600	1024×768	1600×1200
GPU	2.1	3.5	8.5
CPU	167.9	266.7	742.7

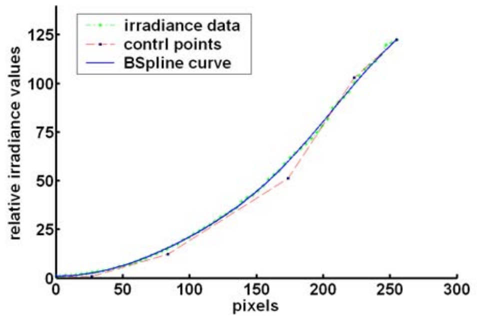


Figure 4: *Approximation of actual ITF using B-Spline curve.*

Table 1 provides the information of computing the luminance surface in the camera space. We can see that the computational time using a GPU is nearly one percent of the computational time using a CPU. In most general cases

Table 2: *Performance of mask generation (ms)*

	800×600	1024×768	1600×1200
GPU	1.88	3.05	7.42
CPU	188.46	316.16	803.23

where the resolution of the camera is 1024×768, the computational time of one luminance surface only needs 3.5 milliseconds (ms) using a GPU, whereas the time using a CPU is up to 300 ms.

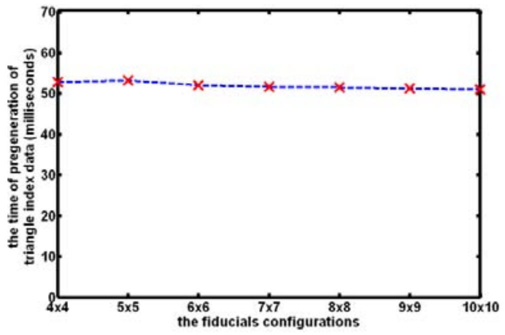


Figure 5: *Performance of generation of the triangle index data.*

We assume that the resolution of the framebuffer of a projector is 800×600, 1024×768, or 1600×1200. Then we test the performances of mask generation in different resolution for each projector using a CPU and a GPU, respectively. As shown in Table 2, the speed of generating the mask using a GPU is about 100 times faster than that using a CPU. By contrast with previous techniques [MS02] [RGM*03] to generate masks using geometric transformation, our technique generates the triangle index data for each pixel in the projector space, and uses the same topology to locate its corresponding pixel in the camera space. Moreover, the GPU can provide multiple rendering pipelines to compute the masks. Such advantages improve greatly the performance of the mask generation. In addition, the pre-generation operation for different fiducial configurations needs no more than 60 milliseconds as shown in Figure 5. It shows that our mask generation technique is efficient.

If the measurements of ITFs for projectors are done, we can see that using the GPU-based framework a multi-projector display system of photometric uniformity can be set up in several minutes. Figure 6 shows the image display after the photometric correction.

4. Conclusions & Future Work

In summary, we adopt a partial-sampling scheme to measure the ITFs of projectors. Moreover we use the B-Spline curve to approximate the actual ITF. In some cases, we can change the positions of the control points to modify the shape of ITF. We extend the computation of luminance surface onto a GPU. Especially, we propose a simple and efficient technique to calculate the masks of the photometric correction using a GPU based on invariant topology. Incorporating the computation of luminance surface and masks using a GPU with the photometric correction pipeline on a GPU, we present a GPU-based framework of photometric uniformity for multi-projector display. Experimental results verify that the performance of the masks' computation is highly improved. In future, we will investigate GPU-based techniques to improve the display quality of images on a multi-projector display system.

References

[BMY] BROWN M., MAJUMDER A., YANG R.: Camera-based calibration techniques for seamless multiprojector displays.

[BS] BROWN M. S., SEALES W. B.: Incorporating geometric registration with pc-cluster rendering for flexible tiled displays.

[CJ01] CHEN C., JOHNSON M.: Fundamentals of scalable high resolution seamlessly tiled projection system. In *Proc. SPIE Projection Displays VII* (2001), vol. 4294, pp. 67–74.

[DM97] DEBEVEC P. E., MALIK J.: Recovering high dynamic range radiance maps from photographs. In *Proc. SIGGRAPH '97* (1997).

[LC99] LI K., CHEN Y.: Optical blending for multi-projector display wall system. In *Proc. 12th Lasers and Electro-Optics Society 1999 Annual Meeting* (1999).

[LLS] LI C., LIN H., SHI J.: A survey of multi-projector tiled display wall construction. In *Third International Conference on Image and Graphics*, pp. 177–188.

[LP95] LES PIEGL W. T.: *The NURBS Book*. New York:Springer, 1995.

[MS] MAJUMDER A., STEVENS R.: Color nonuniformity in projection-based displays: Analysis and solutions.

[MS02] MAJUMDER A., STEVENS R.: Lam : Luminance attenuation map for photometric uniformity across a projection based displays. In *ACM Virtual Reality and Software Technology* (2002).

[RBY*99] RASKAR R., BROWN M. S., YANG R., CHEN W.-C., WELCH G., TOWLES H., SEALES B., FUCHS H.: Multi-projector displays using camera-based registration. In *Proc. Visualization '99* (1999), vol. 31, pp. 161–168.

[RGM*03] RAIJ A., GILL G., MAJUMDER A., TOWLES H., FUCHS H.: Pixelflex2: A comprehensive, automatic, casually-aligned multi-projector display. In *IEEE International Workshop on Projector-Camera Systems* (2003).

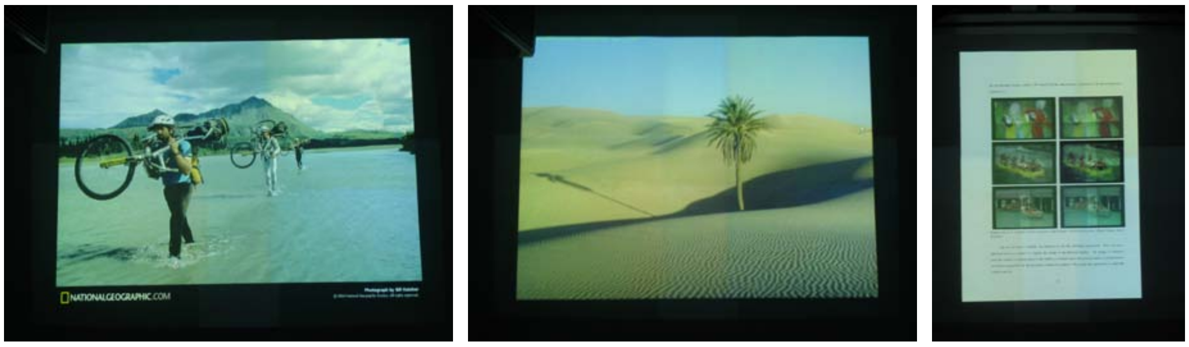


Figure 6: *Display images on the multi-projector system after photometric correction.*

[YQH] YUAN G., QIN K., HU W.: Geometric calibration for multi-projector tiled display based on vanishing point theory. In *International Conference on Computational Science (1)*, pp. 864–867.