

# Color Seamlessness in Multi-Projector Displays Using Constrained Gamut Morphing

Behzad Sajadi, Maxim Lazarov, M. Gopi, and Aditi Majumder, *Member, IEEE*



Fig. 1. This figure shows our results on two multi-projector displays on a curved screen (left) and a planar screen (right). Can you tell the number of projectors making up the display?

**Abstract**— Multi-projector displays show significant spatial variation in 3D color gamut due to variation in the chromaticity gamuts across the projectors, vignetting effect of each projector and also overlap across adjacent projectors. In this paper we present a new constrained gamut morphing algorithm that removes all these variations and results in true color seamlessness across tiled multi-projector displays. Our color morphing algorithm adjusts the intensities of light from each pixel of each projector precisely to achieve a smooth morphing from one projector’s gamut to the other’s through the overlap region. This morphing is achieved by imposing precise constraints on the perceptual difference between the gamuts of two adjacent pixels. In addition, our gamut morphing assures a C1 continuity yielding visually pleasing appearance across the entire display.

We demonstrate our method successfully on a planar and a curved display using both low and high-end projectors. Our approach is completely scalable, efficient and automatic. We also demonstrate the real-time performance of our image correction algorithm on GPUs for interactive applications. To the best of our knowledge, this is the first work that presents a scalable method with strong foundations on perception and realizes, for the first time, a truly seamless display where the number of projectors cannot be deciphered.

**Index Terms**—Color Calibration, Multi-Projector Displays, Tiled Displays

## 1 INTRODUCTION

Spatial variation in color is arguably the most challenging remaining obstacle towards achieving seamlessness in tiled multi-projector displays. Color is a three dimensional quantity defined by 1D luminance (defining brightness) and 2D chrominance (defining hue and saturation). Variation in color across a tiled display can break the illusion of a single display despite perfect geometric registration. This problem is becoming more pronounced with the advent of commodity projectors that cannot afford to use expensive optical elements like precision filters and fresnel lenses to address the problem optically.

The entire range of color (both chrominance and luminance) that can be reproduced by a display is called its 3D color gamut. The color gamut can also be specified separately by a chrominance gamut (defining the entire range of chrominance that can be reproduced by the device), dynamic range (defining the range of luminance that the device can produce), and the white point (the chrominance of the white). In [9, 7] Majumder et al. show that the spatial color variation in multi-projector displays is due to spatial variation in the color gamut across the display and classifies it in three different categories: intra-projector (within a single projector), inter-projector variation (across different projectors), and overlap variation. Existing methods address only parts

of these problems and hence cannot provide a general and comprehensive solution. [15, 16, 11, 17] ignore intra-projector luminance and chrominance variation and proposes a gamut matching method to achieve color balancing across the different projectors. [1] shows that achieving gamut matching is not scalable for many projectors. Further, gamut matching considerably restricts the final achievable gamut reducing the color quality and resolution of the display [19]. [8] ignores both inter- and intra projector chrominance variations and matches the luminance transfer functions (commonly called gamma functions) to achieve a luminance balancing across different projectors. Blending or feathering techniques usually do not capture the spatial color variation across the display and hence ignore both intra and inter projector chrominance and luminance variations. They ramp the intensity from each projector smoothly from 0 to 1 in the overlap region. Multiple projectors overlapping in the region have complementary blending functions to assure a smooth transition from the color of one projector to another. This can be done either in software [14] or using specialized hardware mounted on the light path from the projector [6]. Since these methods do not address the spatial variation in the color in a comprehensive manner, they result in softening of seams rather than achieving a seamless result (Figure 5). In [10] Majumder and Stevens address the spatial variation of luminance to achieve a perceptually smooth variation across the display. However, since the spatial variation in chrominance is not addressed, color blotches are still visible, especially for flat colors when humans are more sensitive to artifacts (Figure 5).

Commercial vendors who build such displays for entertainment and

---

• Behzad Sajadi, Maxim Lazarov, M. Gopi, and Aditi Majumder are in Computer Science Department of University of California, Irvine, E-mail: {bsajadi,mlazarov,gopi,majumder}@uci.edu.

simulation applications have also been trying to address the color variation problem for a long time. Most earlier systems (e.g. the ones built by Barco and Fakespace Labs) used very high-end projectors, almost a couple of orders of magnitude higher in cost than commodity ones, that assure higher color quality. For example, precision filter manufacturing assures better color matching across projectors, and the use of Fresnel lens assures significant spatial uniformity in intensity within a projector. Following this, a standard color management system is used to achieve color uniformity. Note that such systems may use camera assistance to achieve registration, but are not automated via a camera feedback. Since the color variation is not even captured at a high resolution, these systems cannot correct the spatial variation in color entirely, especially for testing cases of images with white backgrounds. More recently, some vendors have started using automated camera based registration techniques (e.g. Scalable Displays and Mersive). In this case also, to assure better color uniformity, projectors that are at least an order of magnitude higher in cost are used. These systems usually use a combination of gamut matching, edge blending and luminance correction methods [16, 10, 14] to remove the remaining color variations, acceptable for less testing patterns. However these methods restrict the chrominance gamut of the projectors to be within the common chrominance gamut of all of the projectors, degrading the resulting color quality. Though this yields acceptable results when using higher end projectors, it can be absolutely crippling for commodity projectors. On the contrary, our work uses extremely commodity projectors and demonstrates much superior color seamlessness than when using high-end projectors, in the presence of much more severe spatial color variations. Hence, our work is critical to push the frontier of the technology for entirely seamless and yet extremely affordable multi-projector displays.

In this paper, we present a new method that addresses spatial variation in both luminance and chrominance in tiled projection-based displays. Our method morphs the spatially varying color gamuts of the display in a smoothly constrained manner while retaining the white point. We use perception based constraints that automatically assure a visually pleasing appearance. The gamut morphing is achieved by a smooth morphing of the chrominance gamut followed by a smoothing of the luminance. Interestingly, the chrominance morph can also be achieved by changing only the luminance at every pixel, but in a manner that is carefully controlled across different projectors in the overlap region. Together they achieve, for the first time, a true color seamlessness following the application of our method where the number of projectors making up the display cannot be detected (Figure 1 and 7). The luminance correction proposed by Majumder and Stevens in [10] results in per projector luminance attenuation map (LAM). Since our chrominance morphing is also achieved by modification of the luminance, our method also results in per projector attenuation maps, but achieves chrominance correction also. Hence integration of our algorithm in the online process to a system that has already used the luminance correction is trivial and can be achieved at an interactive rate of 30 frames per second on a GPU. We demonstrate our method on both planar and cylindrical displays made of a large number of projectors and show their superior results when compared to the state of the art methods. Table 1 summarizes the different color variations we address when compared to other existing methods. Our assumption of negligible intra-projector chrominance variation is due to earlier empirical studies that show chrominance to be nearly constant spatially within most projectors.

## 2 NOTATION

We use CIE XYZ space for all of our color computations. A color in the CIE XYZ color space is defined by its 3D coordinates  $(X, Y, Z)$ , more commonly called the tristimulus values.  $Y$  is the *luminance* of a color and the *chrominance* of a color is given by its chromaticity coordinates,  $(x, y)$ , defined as

$$(x, y) = \left( \frac{X}{X+Y+Z}, \frac{Y}{X+Y+Z} \right). \quad (1)$$

Table 1. Comparison of previous work with our method in handling different types of luminance and chrominance variation in tiled displays.

Method	Intra		Inter		Overlap	
	Lum	Chr	Lum	Chr	Lum	Chr
[15, 16, 11, 17]			X	X		
[8]			X			
[14, 6]					X	X
[10]	X		X		X	
Our Method	X		X	X	X	X

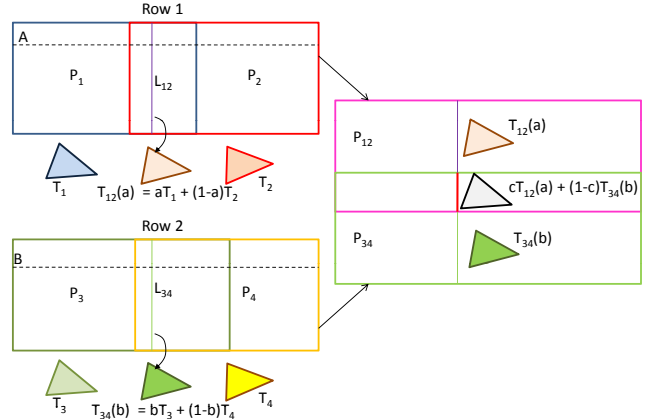


Fig. 2. The morphing of the chrominance gamut in the horizontal and vertical direction in a display made of rectangular projectors and overlaps.

In other words,

$$(X, Y, Z) = (xB, yB, (1-x-y)B) \quad (2)$$

where  $B = X + Y + Z$  which we call the tristimulus brightness. The range of  $(x, y)$  visible to the human eye is the chromaticity chart.

**Scaling of Color:** All colors lying on the vector  $(X, Y, Z)$  can be represented as  $k(X, Y, Z)$  where  $k$  is a scale factor. From Equation 1, it can be seen that the chrominance of all these colors are the same, but the  $B$  and  $Y$  are scaled to,  $kB$  and  $kY$  respectively. Conversely, for any two colors with the same chrominance, the ratio of their  $Y$  and the ratio of their  $B$  are the same.

**Addition of Colors:** The addition of two colors,  $(X_1, Y_1, Z_1)$  and  $(X_2, Y_2, Z_2)$  to create a new color  $(X_3, Y_3, Z_3)$  is given by the vector addition of these two colors in the CIE XYZ space, i.e.,

$$(X_3, Y_3, Z_3) = (X_1 + X_2, Y_1 + Y_2, Z_1 + Z_2) \quad (3)$$

Clearly, the  $Y$  and  $B$  of the new colors are given by

$$Y_3 = Y_1 + Y_2; \quad B_3 = B_1 + B_2. \quad (4)$$

Further, using equations 1 and 2, the chrominance of the new color is given by

$$(x_3, y_3) = \left( \frac{x_1 B_1 + x_2 B_2}{B_1 + B_2}, \frac{y_1 B_1 + y_2 B_2}{B_1 + B_2} \right) \quad (5)$$

Note that  $\frac{B_1}{B_1+B_2}$  and  $\frac{B_2}{B_1+B_2}$  are the barycentric coordinates of the new chrominance  $(x_3, y_3)$  with respect to  $(x_1, y_1)$  and  $(x_2, y_2)$ . This result can be generalized to  $n$  colors, where the chrominance of the new color lies within the convex hull of the chrominance of the constituting  $n$  colors.

## 3 ALGORITHM

A multi-projector display is a display made of  $M$  projectors, each denoted by  $P_j$ ,  $1 \leq j \leq M$ . We denote the pixel coordinates of each projector with  $(p, q)$  and the display coordinates by  $(s, t)$ . We assume

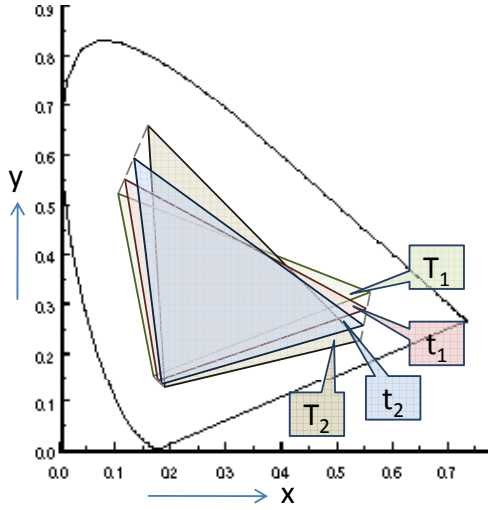


Fig. 3. The chrominance gamut morphs with 2 intermediate steps (across 2 overlapping pixels)

that the relationship between the projector and the display coordinates has been discovered apriori using a geometric registration method like [4, 2]. Hence, the display coordinate  $(s, t)$  is related to the coordinates  $(p_j, q_j)$  of projector  $P_j$  by  $(s, t) = G_j(p_j, q_j)$ .  $G_j$  can be deciphered using any standard geometric calibration technique [4, 18, 12, 2].

In a display, let the color formed by a channel input  $i_l = 1$  be  $(X_l, Y_l, Z_l)$  and hence  $B_l = X_l + Y_l + Z_l$ . Assuming linear displays, as  $i_l$  changes from 0 to 1, the colors are given by  $i_l(X_l, Y_l, Z_l)$ . Hence, from Section 2, their chrominance remains constant and  $B_l$  is scaled. The three channels of the display form three vectors in the XYZ space. Hence, all the of colors reproduced by the display are given by the parallelepiped spanned by the three vectors  $(X_l, Y_l, Z_l)$ ,  $l \in \{r, g, b\}$  where  $\{r, g, b\}$  are red, green, and blue color channels. This is called the *3D color gamut* of the display. Alternatively, this gamut can also be defined by a 2D chrominance gamut, the tristimulus brightness and the chrominance of white produced by  $i_r = i_g = i_b = 1$ . The 2D chrominance gamut is given by the triangle  $T$  given by  $(x_l, y_l)$ ,  $l \in \{r, g, b\}$ . The tristimulus brightness of white,  $B_W$ , is given by  $B_W = \sum_l X_l + Y_l + Z_l = \sum_l B_l$ . And the chrominance of the white,  $(x_W, y_W)$  is given by

$$(x_W, y_W) = \sum_l \frac{B_l}{B_W} (x_l, y_l). \quad (6)$$

**Single Projector Display:** Empirical data in [9] shows that the color within a projector shows spatial dependency due to the vignetting effect. The vignetting effect is a spatial fall off of brightness from around the center to the fringes of the projector. However, this is a channel independent effect and hence affects the  $B_l$  of all the different channels in the same manner. Hence,  $B_l$  at any pixel  $(p, q)$  is given by

$$B_l(p, q) = V(p, q)B_l. \quad (7)$$

Where  $V(p, q)$  is the vignetting of the projector at pixel  $(p, q)$ . Thus, the spatially varying tristimulus brightness of white is given by

$$B_W(p, q) = \sum_l V(p, q)B_l. \quad (8)$$

However, since the vignetting is a channel independent scale factor, it does not affect the 2D chrominance gamut  $T$  which is still spatially constant.

**Multi-Projector Display:** Let us now consider the overlap of  $N$  projectors and a coordinate  $(s, t)$  in the overlap region. Thus, the  $B_l$  for each channel in the overlap region is given by

$$B_l(s, t) = \sum_{j \in N} V_j(p_j, q_j)B_l. \quad (9)$$

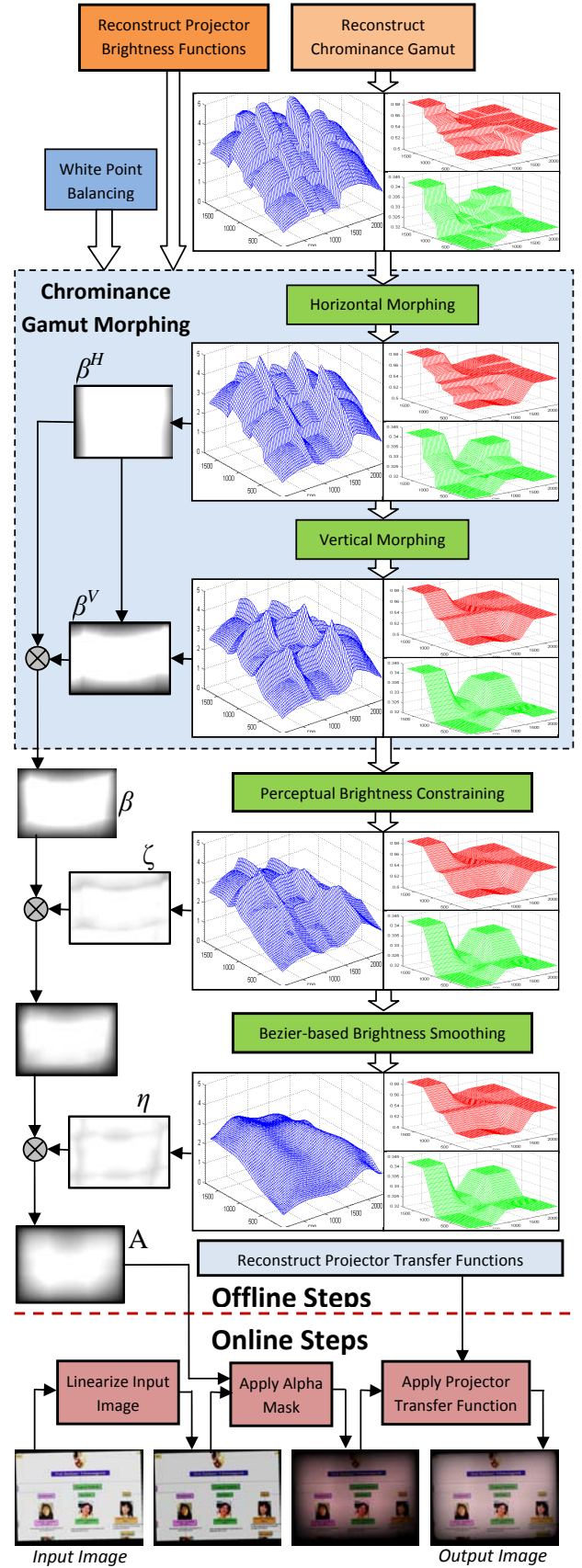


Fig. 4. The complete flowchart of our algorithm. We show the spatial variation in brightness (blue), the x (red) and y (green) of the red primary of the entire display after every step of our algorithm. Note how all of these are smoothed during the course of our method. On the left, we show the attenuation map for a single projector after every step.

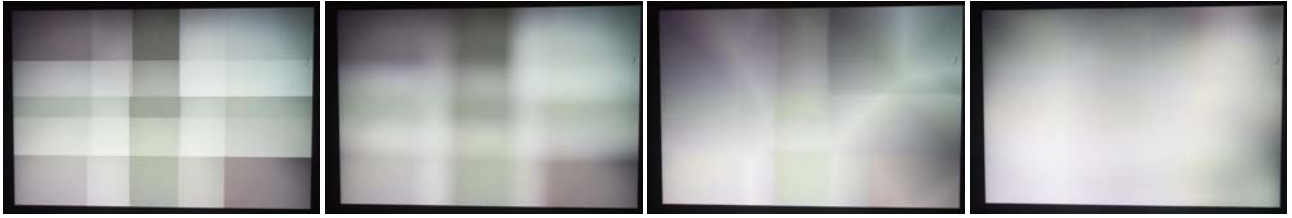


Fig. 5. Comparison of our method with existing work on the most difficult case of white on the planar display. Note that the white is the most difficult test case showing both chrominance and brightness variation. From left to right: Before any correction; After simple RGB blending; After applying Majumder and Stevens 2005 photometric seamlessness algorithm with  $\lambda = 400$ ; after our gamut morphing algorithm where the perceptual brightness constraining uses the same method by Majumder and Stevens 2005 with  $\lambda = 250$ . Please zoom in to see details.

This indicates a spatial variation in  $B_W$  given by

$$B_W(s,t) = \sum_l \left( \sum_{j \in N} V_j(p_j, q_j) B_{l_j} \right). \quad (10)$$

However, the chrominance of each channel will now depend on the proportions of the of the resulting  $B_l$  from different channels. Since the vignetting is different for different projectors,  $B_l$  will change in a channel dependent manner in the overlap region thus creating spatially varying chrominance gamut  $T(s,t)$ .  $T(s,t)$  is defined by the three primaries  $(x_l(s,t), y_l(s,t))$  as

$$(x_l(s,t), y_l(s,t)) = \sum_j \frac{B_l(s,t)}{B_W(s,t)} (x_{l_j}, y_{l_j}) \quad (11)$$

Note that the variation in the  $T(s,t)$  stems from the variation in the  $B_l$  and  $B_W$  rather than the  $(x_{l_j}, y_{l_j})$  which are constant for each projector  $P_j$ . From Equation 9 and 10, we see that this variation in turn depends on the relative shape of vignetting effect,  $V_j(p_j, q_j)$ , and the brightness,  $B_{l_j}$ , of the contributing projectors, both of which can show a large variation in any tiled display [9] leading marked variation and discontinuities of the chrominance in the overlap region. Similarly,  $B_W(s,t)$  also shows visible discontinuities (Figure 4).

Our goal is to remove such discontinuities to create a perceptually seamless appearance. The key to this lies in controlling the  $V_j$  and  $B_{l_j}$  from each projector, by scaling their contributions at each pixel precisely, to realize a smooth change in both  $B_W(s,t)$  and  $T(s,t)$ . Further, we want to constrain this per-pixel scaling based on strong foundations of limitations of human perception, so that they are not distracting to the human eye. This will result in a smooth morphing of the 3D gamut from one pixel to another ensuring perceptual seamlessness. We present a new algorithm to achieve such a per-pixel scaling that assures a smooth 3D gamut morphing. Our algorithm has four different steps: (a) *Per Projector White Point Balancing*; (b) *Per Pixel Chrominance Gamut Morphing*; (c) *Per Pixel Perceptual Luminance Constraining*; and (c) *Per Pixel Bezier Based Luminance Smoothing*.

### 3.1 Per Projector White Point Balancing

It is a well known fact that we humans are very sensitive to the white point of displays [5]. So, the first step in our method is a white point balancing for each projector. Let  $(x_D, y_D)$  be the desired white point. Using equation 6, we find for each projector a per-channel scale factor  $\alpha_l$ ,  $0 \leq \alpha_l \leq 1$  such that

$$\frac{\sum_l \alpha_l B_l(x_l, y_l)}{\sum_l \alpha_l B_l} = (x_D, y_D), \quad (12)$$

where  $(x_D, y_D)$  is the desired white point. We first fix  $\alpha_r = 1$  and then solve the two linear equations resulting from the above equations to find  $\alpha_g$  and  $\alpha_b$ . If this leads to an  $\alpha$  which is not within 0 and 1, then we repeat the process with either  $\alpha_g = 1$  or  $\alpha_b = 1$ . Note that since the vignetting effect scales  $B_l$  and  $B$  similarly at any pixel, the above equation will yield the same  $\alpha_l$  irrespective of the pixel at which it is computed.

### 3.2 Per Pixel Chrominance Gamut Morphing

In this step we morph the two-dimensional chrominance gamut of one projector to that of the other across the overlap region. However, the morph we design is constrained to retain the white point that was achieved in our earlier step. Further, this morph assumes rectangular projections and horizontal/vertical overlaps. An algorithm to assure this in any general tiled display is described in the appendix. Following is the description of our chrominance gamut morphing method.

A chrominance gamut  $T$  is denoted by a triangle whose vertices are given by  $R$ ,  $G$  and  $B$ . Let us now consider two chrominance gamuts  $T_1$  and  $T_2$ . We desire to define a morph that takes  $R_1$  to  $R_2$ ,  $G_1$  to  $G_2$  and  $B_1$  to  $B_2$ . For an intermediate chrominance gamut  $t_k$ , we would like the chrominance of the primaries  $r_k$ ,  $g_k$  and  $b_k$  to be a *constrained* linear combination of the chrominance of the primaries of  $T_1$  and  $T_2$ . Hence,

$$r_k = (1 - \tau)R_1 + \tau R_2 \quad (13)$$

$$g_k = (1 - \tau)G_1 + \tau G_2 \quad (14)$$

$$b_k = (1 - \tau)B_1 + \tau B_2 \quad (15)$$

Note that from Equation 5,  $1 - \tau$  is given by the proportions of the  $B_l$  of the channel. Hence, we need to find scale factors  $\beta_{l_1}$  and  $\beta_{l_2}$  between 0 and 1 such that

$$\frac{\beta_{l_1} B_{l_1}}{\beta_{l_1} B_{l_1} + \beta_{l_2} B_{l_2}} = 1 - \tau \quad (16)$$

$$\frac{\beta_{l_2} B_{l_2}}{\beta_{l_1} B_{l_1} + \beta_{l_2} B_{l_2}} = \tau \quad (17)$$

Clearly, computing  $\beta_{l_1}$  and  $\beta_{l_2}$  from the above proportions is an underconstrained system. So, we first fix  $\beta_{l_1}$  and compute the  $\beta_{l_2}$ . If this exceeds 1, we reverse the computation by setting  $\beta_{l_2} = 1$ . However, note that this process results in per channel scale factor at every pixel which would ruin the white point balancing achieved in the previous step. So, we make a conscious choice here to retain the white point by computing one common factor for all channels. Hence we seek  $\beta_1$  and  $\beta_2$  such that

$$\frac{\beta_1 B_{W_1}}{\beta_1 B_{W_1} + \beta_2 B_{W_2}} = 1 - \tau \quad (18)$$

$$\frac{\beta_2 B_{W_2}}{\beta_1 B_{W_1} + \beta_2 B_{W_2}} = \tau \quad (19)$$

Note that this means that the intermediate chrominance gamut  $t_k$  is still a linear combination of  $T_1$  and  $T_2$ .

The next question is how many such  $t_k$ s are required so that the transition between  $T_1$  and  $T_2$  is imperceptible. Let us assume that we make the transition from  $T_1$  to  $T_2$  through  $n$  steps  $T_1 \rightarrow t_1 \rightarrow t_2 \dots t_n \rightarrow T_2$  such that each of the transitions  $t_k \rightarrow t_{k+1}$  is imperceptible. To assure this, we first find the the smallest imperceptible distance  $\delta$  in the chromaticity chart. Then, we choose  $n$  as

$$n = \frac{\max(|R_1 R_2|, |G_1 G_2|, |B_1 B_2|)}{\delta} \quad (20)$$



Fig. 6. Comparison of our method with existing work on the most difficult case of white on the curved display made of  $2 \times 4$  array of eight displays. In scanline order: Before any correction; After simple RGB blending; After applying Majumder and Stevens 2005 photometric seamlessness algorithm with  $\lambda = 400$ ; after our gamut morphing algorithm where the perceptual brightness constraining uses the same method by Majumder and Stevens 2005 with  $\lambda = 250$ .

We discuss the value of  $\delta$  in Section 6. This chrominance gamut morphing is illustrated in Figure 2 and 3.

Thus, following the horizontal and vertical morphs as described above, a scale factor for each projector is generated for each pixel in the overlap region. These two are multiplied to create one spatially dependent map of per pixel attenuation factors per projector that defines how the inputs of the projectors at every pixel of the overlapping region should be scaled to achieve a smooth morphing of the chrominance gamut. These maps after each of the horizontal and vertical chrominance gamut morph is shown in Figure 4.

It is important to note that  $n$  decides the number of pixels required to achieve an imperceptible morph. If this  $n$  is greater than the size of the overlap region in pixels, we may not be able to achieve an imperceptible morph. This demands a minimum size of the overlap to achieve chrominance gamut morphing and is discussed in details in Section 6. However, if  $n$  is less than the size of the overlap region, then we morph in the first and last  $\frac{n}{2}$  pixels in the overlap. For the remaining pixels in the center of the overlap, we compute and maintain the chrominance gamut  $t_{k_{n/2}} = 0.5(T_1 + T_2)$ . This assures that the brightness of the overlap region following the chrominance gamut morph is maximized. Note that, since the scale factors  $\beta_1$  and  $\beta_2$  are chosen to be same across all channels as opposed to different for different channels, the transition from  $t_k$  to  $t_{k+1}$  will not assure equal changes in the chromaticity coordinates of the primaries, rather a monotonically increasing or decreasing change in the chromaticity coordinates of the primaries. The effect of this in the choice of  $\delta$  is discussed in Section 6.

Thus, the chrominance gamut morphing step generates for each projector  $P_j$  two attenuation maps,  $\beta_j^H(p_j, q_j)$  and  $\beta_j^V(p_j, q_j)$ , following horizontal and vertical morphs respectively. These two are multiplied to generate the final map

$$\beta_j(p_j, q_j) = \beta_j^H(p_j, q_j) \times \beta_j^V(p_j, q_j)$$

as shown in Figure 4. Following the application of the above attenuation maps, the new  $B_W$ , denoted by  $B_{W_C}(s, t)$ , at every pixel  $(s, t)$  on the display becomes

$$B_{W_C}(s, t) = \sum_{j \in N} \left( \sum_I \beta_j(p_j, q_j) B_{I_j} V_j(p_j, q_j) \right) \quad (21)$$

where  $N$  is the set of projectors that overlap at pixel  $(s, t)$  (Figure 4).

### 3.3 Per Pixel Perceptual Brightness Constraining

Our chrominance gamut morphing step assured that the chrominance changes smoothly across the entire display. But no constraints have yet been put on the brightness variation of the display. Thus, the display still shows sharp brightness changes resulting in seams. In this step, we apply the perceptual constraining of Majumder and Stevens 2005 on  $B_{W_C}(s, t)$ . By this, a perception based gradient constraint is applied to  $B_{W_C}(s, t)$  resulting in  $B_{W_E}(s, t)$  as shown in Figure 4. To achieve this  $B_{W_E}(s, t)$ , the contributions from all channels at that pixel should be multiplied by

$$\zeta(s, t) = \frac{B_{W_E}(s, t)}{B_{W_C}(s, t)} \quad (22)$$

This attenuation map  $\zeta(s, t)$  in the display coordinates is then broken into attenuation maps for each projector  $\zeta_j(p_j, q_j)$ . Note that in this case,  $N_p$  projectors overlapping in a pixel  $(s, t)$  all get the same  $\zeta(s, t)$ . Hence, this step retains the chrominance achieved by the chrominance gamut morphing step and changes only the brightness. Both the attenuation map and the brightness constraining are illustrated in Figure 4.

### 3.4 Per Pixel Bezier Based Brightness Smoothing

The luminance obtained after the perceptual luminance constraining still has  $C^1$  discontinuity. This step is designed to seek a brightness  $B_{W_S}$  which is closest to  $B_{W_E}$  but assures  $C^1$  continuity. For this, we fit a higher order  $C^1$  continuous 2D Bezier surface to  $B_{W_E}$ . However, to assure  $C^1$  continuity in the perceptual domain, we apply this operation in log scale. We call the function thus generated  $B_{W_S}(s, t)$ . To achieve the  $B_{W_S}(s, t)$ , the contributions from all channels at the pixel should be multiplied by

$$\eta(s, t) = \frac{B_{W_S}(s, t)}{B_{W_E}(s, t)} \quad (23)$$

$\eta(s, t)$  is then again broken into attenuation maps for each projector  $\eta_j(p_j, q_j)$ . Note that since this is a surface fitting step,  $\eta(s, t)$  can be slightly greater than 1.0 at places. So, we normalize  $\eta(s, t)$  to assure no saturation artifacts. In this case also, the  $N$  projectors overlapping in a pixel  $(s, t)$  all get the same  $\eta(s, t)$ . Hence, this step also retains the chrominance achieved by the chrominance gamut morphing step and changes only the brightness. Both the attenuation map and the result of the Bezier fitting is illustrated in Figure 4.



Fig. 7. A large variety of images corrected using our method on our 9 projector planar (top) and eight projector curved display (bottom). Note that the number of projectors are not visible in any of them. Please zoom in to see details.

### 3.5 Image Correction

Let  $I(p, q)$  be the image generated in the projector coordinates after applying the geometric function  $G$ . First, we linearize the image  $I(p, q)$  using a gamma function of 2. We multiply the different attenuation maps to create a final attenuation map  $A_j(p, q) = \beta_j(p, q) \times \zeta_j(p, q) \times \eta_j(p, q)$ . To achieve the color correction, we pixel-wise multiply  $I$  with the attenuation map  $A_j$  to generate an appropriately scaled  $I$ ,  $I_S(p, q)$ , such that

$$I_S(p, q) = I(p, q) \times A_j(p, q)$$

Note that  $A_j$  is channel independent and hence multiplies all the channels of  $I_j$  similarly. Following this, we apply the channel dependent white point correction to generate the white point corrected image  $I_W$  as

$$I_{W_l}(p, q) = \alpha_l I_{S_l}(p, q)$$

where  $l \in \{r, g, b\}$ . Now note, all the above corrections described till now assumes a linear projector. However, in practice the projectors have a channel input transfer function, denoted by  $h_l$ . So, to achieve the desired changes in this non-linear device, the final correction is achieved by applying the inverse of  $h_l$  to  $I_W$ . Hence, the final corrected image  $I_C$  is given by

$$I_C(p, q) = h_l^{-1}(I_{W_l}(p, q)),$$

where  $l \in \{r, g, b\}$ .

## 4 IMPLEMENTATION

We have implemented our method on two displays: (a) a planar rear projected display of  $3 \times 3$  array of nine very low-end projectors; (b) a cylindrical front projected display of  $2 \times 4$  array of eight relatively higher-end projectors. Since (a) is rear-projection system with a screen that deviates considerably from a Lambertian surface and uses old (about 3-4 years) low-end projectors, this shows severe color variation and is an excellent test case for our algorithm. Although (b) uses a diffuse front projection screen with relatively new (1 year) projectors, they also show considerable color variation.

To reconstruct the spatially varying color gamut for the projectors, we use an sRGB camera as a sensor. Ideally, a spectroradiometer should be used to measure the chrominance. But since most projectors have a chrominance gamut within the sRGB gamut [9], using a camera

yields sufficient accuracy in measurement. To measure the chromaticity gamut of each projector, we put up for each projector the maximum intensity red, green and blue values. We used a small region near the center of each projector where sRGB measurements are averaged, converted to XYZ using standard sRGB to XYZ conversions and then the chromaticity coordinates are computed. The vignetting effect is measured by projecting white from each projector and then capturing it using the high resolution camera. The geometric relationship between the camera, projector and the display are recovered using methods proposed in [2]. The camera RGB values at corresponding locations for the projector are then converted to XYZ and then to tristimulus brightness to recover the vignetting effect  $V_j$ . To recover  $h_l$  we use the technique proposed in [10] where we project a uniform sampling of inputs for each channel  $l$  and find their corresponding tristimulus brightness from the captured RGB value at the center of the projector.

**Real-time Image Correction on GPU:** Our method has an offline calibration step and an online correction step (Figure 4). We used MATLAB for offline computation of the attenuation maps and takes about 5 minutes; this can be easily ported to any other language such as C++ for greater efficiency. This offline calibration generates three alpha masks (one for each channel) and three inverse transfer functions (one for each channel) for each projector, which are then used for image correction. The online image correction is achieved in real-time (30 fps) using GPUs through Chromium, an open-source distributed rendering engine for PC clusters [8]. A module for Chromium is written that first multiplies the image with the alpha masks and then uses an 1D look-up-table to apply the inverse transfer functions. These are done using a fragment shaders on the GPUs at interactive rates.

## 5 RESULTS

In this section, we show our results on both the planar and the curved screen. Note that the acid test for any color correction method is white and flat colors. Results on such images are not shown in almost any prior work. Figure 5 compares our method with traditional RGB blending [14] and the method of photometric seamlessness proposed by Majumder and Stevens [10] on white on our worst display (planar one) made of old low-end projectors on rear-projection screen. We also compare our results on the much better projectors on the curved screen in Figure 6. In both cases, our results show much superior seamlessness. Figure 7 shows the results of our algorithm on many different images both on the planar and the curved screen display. Figure 8 shows the different steps of our process and how they progres-

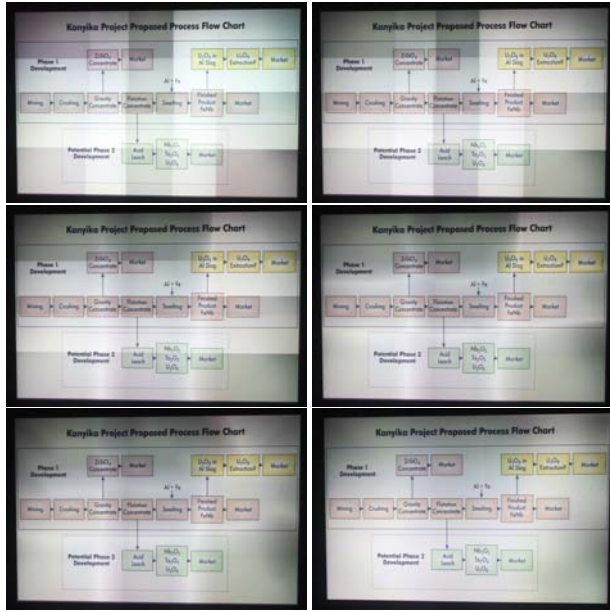


Fig. 8. The results from the different steps of our method illustrated on the planar display of 9 projectors. In scanline order from top left: Before correction; after white point balancing; After chrominance gamut morphing in the horizontal direction; After chrominance gamut morphing in the vertical direction; after perceptual luminance constraining; final result after Bezier based smoothing.

sively improve the quality of seamlessness resulting in the final one where the projectors cannot be detected.

Our method assumes no spatial chrominance variations within a single projector. However, in practice, commodity projectors does show small smooth spatial variations in chrominance [9]. Further, our method *morphs* the 3D color gamut across the display to make local color variations imperceptible. If we attempt to *match* the 3D color gamut at every pixel the brightness of the display will be limited to the brightness of the darkest pixel and the chrominance gamut will be limited to the common chrominance gamut of all of the projectors which is usually very small. Instead, we smoothly morph the 3D color gamut in order to remove the local variations while retaining as much of the display color gamut and dynamic range as possible. Hence, global color variations still exist. Note that these global variations are not the result of applying our method, but were camouflaged in presence of more drastic local variations. Further, human perceptibility of these global variations is closely tied to the angle subtended by the display in the human eye. This angle is significantly smaller when presented as a printed picture in the paper than in reality. So, these global variations, that are not disturbing when seen in person, can become perceptible when presented in the paper.

## 6 DISCUSSION

In this section, we discuss the different system issues, accuracy and performance.

### 6.1 Minimum size of the overlap and $\delta$

Consider the overlap of two projectors  $P_1$  and  $P_2$ . For projector  $P_1$ , following the chrominance gamut morph,  $\tau$  defines the contribution of  $P_1$  to a pixel in the overlapping region. Hence,  $\frac{B_{W_1}}{B_{W_1} + B_{W_2}} = \tau$ . Now, let us consider the implication of this on the change in the proportion of  $B_l$  for a single channel from  $P_1$  and  $P_2$ . Let us consider channel  $l$ , and let  $B_{l_1} = k_1 B_1$  and  $B_{l_2} = k_2 B_2$ . Hence, the proportion of the brightness of one channel  $l$  for  $P_1$  is given by

$$\frac{k_1 B_{W_1}}{k_1 B_{W_1} + k_2 B_{W_2}} = \frac{k_1 \tau}{k_1 \tau + k_2 - k_2 \tau}$$

after replacing  $B_{W_2} = \frac{1-\tau}{\tau} B_{W_1}$ . Taking the derivative of this we find that the speed of this will vary between  $\frac{k_1}{k_2}$  and  $\frac{k_2}{k_1}$ . Note that usually  $k_1$

and  $k_2$  vary from projector to projector but not much since these signify the proportion of a primary in white. Hence, both these functions are close to 1. Thus, the chrominance of each primary will not see a linear change but a monotonically increasing/decreasing change that is close to linear. This is also evident from the chromaticity plots in Figure 4.

During the chrominance gamut morph, ideally, we want  $\delta$  to be less than 3 or 4 in the CIE LAB space in order for the color morph to be imperceptible. We ran the following analysis to find how much overlap we need to assure such a  $\delta$ . Consider the channel  $l$  on a pair of projectors,  $P_1$  and  $P_2$ . Knowing the distance between the primary  $l$  on these two projectors, we plotted the maximum LAB distance between adjacent pixels in the overlap of  $P_1$  and  $P_2$  for varying size of the overlap region. We found this curve for all pairs of projectors (in our pool of 17 projectors) and then found the maximum of the LAB distance for each overlap size. The resulting plot (Figure 9) provides us with a good estimate of the maximum LAB distance that would result between adjacent pixels with varying overlap size. We know that a LAB distance of 3 or less is within the just noticeable distance (JND). In the plot we see that to be within the JND, we need a minimum of 90 pixels overlap for the green primary. As expected, the overlap region required for red and blue to assure the same is much smaller. This is due to the fact that the relatively higher luminance of green makes it easier to perceive chrominance variations. An overlap region of about 90 pixels is around 10% of the projector resolution in any direction and less than 5% of the screen resolution in any direction. It is often required to overlap projectors much more to alleviate the fall-off due to vignetting effect [9, 19]. Hence, the overlap region required by our method is reasonable in almost all tiled displays.

### 6.2 Difference from Traditional RGB Blending

Traditional blending methods feather the RGB input of the projector in a linear or cosine manner in the overlap region [14]. Our chrominance morphing achieves a superior result than such traditional RGB blending. The primary reason for that is that since we use a precise measurement of the projector's spatial brightness variation we can constrain the chrominance gamut morphing speed to be as close to linear as possible. Hence, we constrain the change in chrominance easily to be within the human tolerance (Figure 9). Traditional blending does not consider the brightness falloff of the projectors and just uses a linear or cosine function to blend the brightness of each channel in the overlap region. This does not assure a linear change in the  $(x, y)$  space and becomes especially pronounced when the projectors do not have similar brightness falloffs. An additional advantage of our chrominance gamut morph is that, we may not use the entire overlap region for morphing and hence retain more of the brightness of the display.

### 6.3 Effect on the Display Quality

It is evident that every step of any color calibration method, including ours, imposes some constraints on the spatial color variation that would invariably lead to some reduction in the brightness of the display. Table 2 evaluates this reduction in dynamic range versus the achieved uniformity through the different stages of our method and with other existing methods. As a metric, for dynamic range we use the mean of brightness of the white at all pixels of the display and for uniformity we use standard deviation from the mean brightness and standard deviation of the euclidian distance of the chrominance of each pixel from mean chrominance. Note that our method shows a reduction in dynamic range when compared to the photometric seamlessness method of Majumder and Stevens [10] and edge blending. However, we achieve greater smoothness in both chrominance and brightness variation as indicated by the reduction in the standard deviations by an order of magnitude.

### 6.4 Is the perceptual brightness constraining required?

In our method, since the last step of Bezier based smoothing anyway smooths the spatial brightness variation, one may question the utility of the perceptual brightness constraining. We emphasize this in Figure 10, which shows the result with and without the use of this method.

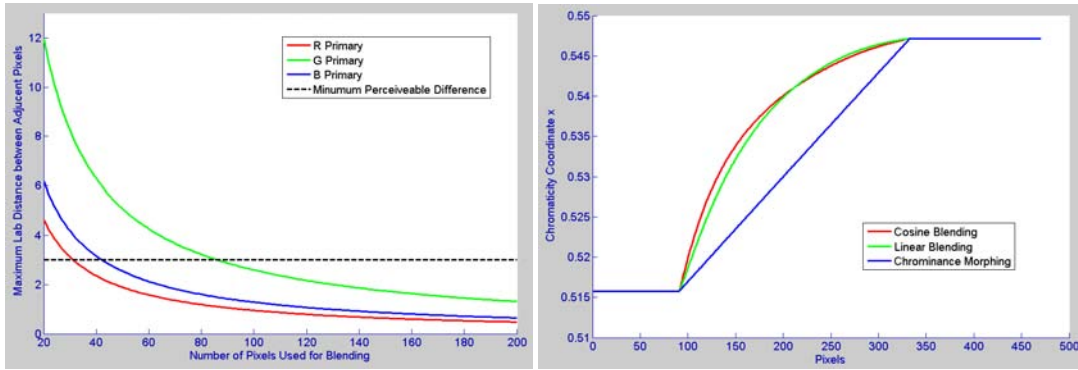


Fig. 9. Left: The plot of the size of the overlap region vs the maximum LAB distance between the chrominance of the primaries of adjacent pixels. Note that for less than a 90 pixels, the LAB distance is within the threshold of just noticeable difference. Right: Comparison of change in chromaticity coordinates across an overlap region achieved by RGB blending with our chrominance morphing.

Table 2. Evaluation of the percentage reduction in dynamic range(DR) and the uniformity achieved (measured by the standard deviation of the variation in brightness and chrominance from the mean across the display) by different steps of our methods and other existing methods.

Methods	Mean DR Loss	Brightness STD	Chrominance STD
Before Correction	0%	0.044	0.035
After Chrominance Gamut Morphing	13.49%	0.035	0.0089
After Perceptual Brightness Constraint	19.33%	0.024	0.0088
After Bezier Based Smoothing (Our Method)	25.86%	0.019	0.0086
Majumder & Stevens 2005	10.07%	0.031	0.026
RGB Blending	7.49%	0.038	0.029

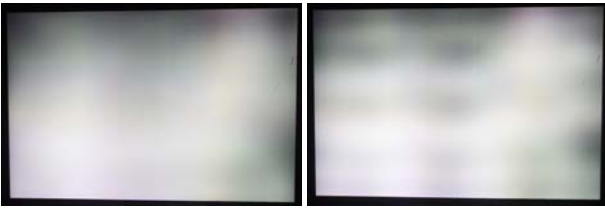


Fig. 10. The comparison of our method with (left) and without (right) applying the perceptual brightness constraining. Note that not applying perceptual brightness constraining cannot yield the desired smoothness.

Note that the results are smoother and visually more pleasing when the perceptual brightness constraining is applied. Further, the Bezier based smoothing is just a way to fit a smooth function to the spatially varying brightness. If the underlying variation is perceptible to the human eye, the Bezier based smoothing seldom makes it imperceptible. Applying the perceptual brightness constraining step already assures that the spatial variation is within the human tolerance and hence the Bezier smoothing will provide a good fit to that.

## 7 CONCLUSION

In conclusion, we have presented, for multi-projector displays, the first method that addresses 3D color variations within and across projectors and in the overlap region simultaneously. Our method smooths both the chrominance and the brightness across the entire display resulting in a true color seamlessness where the number of projectors are absolutely invisible in the display. This is achieved by a morphing of the 2D chrominance gamut in the overlap regions followed by brightness smoothing across the entire display, both guided by perceptual parameters to assure that the variation is within the human tolerance and is hence not detected. Unlike previous methods, our method is successful to achieve seamlessness even for the difficult cases of flat colors, especially flat white. Our work demonstrates, for the first time, that inexpensive multi-projector displays can indeed provide similar seamlessness as the cost-prohibitive high-resolution single projector displays.

However, in this work, we assume that the chrominance gamut remains constant across a projector. But, in reality, the chrominance gamut does show some variation within a single projector. We would like to remove this constraining assumption in future work to improve the results. There has been earlier work in achieving geometric registration in a distributed manner [3]. We would like to explore our method to achieve color seamlessness in a distributed manner in large multi-projector displays.

## REFERENCES

- [1] M. Bern and D. Eppstein. Optimized color gamuts for tiled displays. *ACM Computing Research Repository, cs.CG/0212007, 19th ACM Symposium on Computational Geometry, San Diego*, 2003.
- [2] E. Bhasker, R. Juang, and A. Majumder. Registration techniques for using imperfect and partially calibrated devices in planar multi-projector displays. *IEEE TVCG*, 13(6):1368–1375, 2007.
- [3] E. Bhasker, P. Sinha, and A. Majumder. Asynchronous distributed calibration for scalable reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics (Visualization) - To Appear*, 2006.
- [4] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable alignment of large-format multi-projector displays using camera homography trees. *Proc. of IEEE Vis*, 2002.
- [5] E. J. Giorgianni and T. E. Madden. *Digital Color Management : Encoding Solutions*. Addison Wesley, 1998.
- [6] M. Hereld, I. Judson, and R. Stevens. Introduction to building projection-based tiled display systems. *IEEE Computer Graphics and Applications*, 2000.
- [7] A. Majumder and M. Gopi. Modeling color properties of tiled displays. *Computer Graphics Forum*, June 2005.
- [8] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. *Proceedings of IEEE Vis*, 2000.
- [9] A. Majumder and R. Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. *IEEE Transactions on Vis and Computer Graphics*, 10(2), March–April 2003.
- [10] A. Majumder and R. Stevens. Perceptual photometric seamlessness in tiled projection-based displays. *ACM TOG*, 24(1), January 2005.
- [11] B. Pailthorpe, N. Bordes, W. Bleha, S. Reinsch, and J. Moreland. High-resolution display with uniform illumination. *Proceedings Asia Display IDW*, pages 1295–1298, 2001.



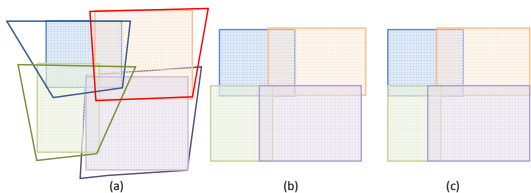


Fig. 11. Method to find rectangular projections and overlaps from a set of overlapping keystoneed projectors.

- [12] R. Raskar, M. Brown, R. Yang, W. Chen, H. Towles, B. Seales, and H. Fuchs. Multi projector displays using camera based registration. *Proc. of IEEE Vis*, 1999.
- [13] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: Geometrically aware and self-configuring projectors. *ACM TOG*, 22(3), 2003.
- [14] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: A unified approach to image based modeling and spatially immersive display. In *Proceedings of ACM Siggraph*, pages 168–176, 1998.
- [15] M. C. Stone. Color balancing experimental projection displays. *9th IS&T/SID Color Imaging Conference*, 2001a.
- [16] M. C. Stone. Color and brightness appearance issues in tiled displays. *IEEE Computer Graphics and Applications*, 2001b.
- [17] G. Wallace, H. Chen, and K. Li. Color gamut matching for tiled display walls. *Immersive Projection Technology Workshop*, 2003.
- [18] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. S. Brown. Pixelflex: A reconfigurable multi-projector display system. *Proc. of IEEE Vis*, 2001.
- [19] R. Yang, A. Majumder, and M. Brown. Camera based calibration techniques for seamless multi-projector displays. *IEEE TVCG*, 11(2), March–April 2005.

## APPENDIX

We describe here an algorithm to achieve rectangular projections and overlaps for our chrominance gamut morphing method. For illustration of the process, please refer to Figure 2. Let us consider a single row of two *rectangular* projectors  $P_1$  and  $P_2$ . Their 2D chrominance gamut in the non-overlapping regions are denoted by the  $T_1$  and  $T_2$  respectively. Now, consider a horizontal scanline  $A$  through these two projectors. At any point in the overlap region in this scanline, we want to morph the chrominance gamut in the horizontal direction. This is achieved by linear morphing of  $T_1$  to  $T_2$  controlled by the parameter  $a$ . Details of this morphing algorithm is provided later in this section. Since the overlap between the projectors is rectangular, any vertical line  $L_{12}$  will have the identical morphed gamut,  $T_{12}(a)$ . Following the horizontal morph, we can consider  $P_1$  and  $P_2$  together to form a large projector  $P_{12}$  where every vertical line has identical chrominance gamut. Similarly, horizontal chrominance gamut morphing can be applied independently to another row of projectors  $P_3$  and  $P_4$  to form a large projector  $P_{34}$  where the same property holds. Now, in a display made of two rows of projectors  $P_{12}$  and  $P_{34}$  with a overlap between them every vertical line follows the property that it has constant chrominance gamut in the region with no overlap. Hence, they can be morphed in the vertical direction linearly exactly the same way as the horizontal morph. This method can extend to  $n$  projectors in each row and  $m$  rows in the display. Thus, by performing two passes of chrominance gamut morphs in each of the horizontal and vertical direction, we can produce a smoothly morphed chrominance gamut at any point on the display.

However, note that the key aspect of the above method is that the projectors have to be rectangular with rectangular overlap regions. This is hardly the case, especially since projectors have off-axis projection leading to keystoneing effect. Hence, we first turn off some pixel at the boundaries to make each projection rectangular. Figure 11 describes this process. First, we find the largest inscribed rectangle for each projector using methods proposed in [13], as shown in Figure 11(a). Next, we consider every row of projectors independently as a single projector and repeat the process of the finding the largest

inscribed rectangle in them. All pixels outside this rectangle are again turned off. Multiple such rows of projectors are arranged in a column and the above process is repeated again. The above process ensures a rectangular projection, display, and horizontal/vertical overlaps. This process is illustrated in Figure 11. Note that the keystoneing showed for each projector is exaggerated for illustration purpose. In reality, the keystoneing is much smaller and this does not lead to much pixel wastage. We measured these wasted pixels for both our set up. On the planar display, where it is easy to achieve a close to rectangular grid, the wastage was about 9.6%. On the curved display, where the keystoneing is more severe, we see a wastage of 19.8%.