Color Seamlessness in Multi-Projector Displays using Constrained Gamut Morphing

*IEEE Transactions on Visualization and Computer Graphics, Vol. 15, No. 6, 2009*

Behzad Sajadi, Maxim Lazarov, Aditi Majumder, and M. Gopi

Presented by Il-Su Park

*School of Electrical Engineering and Computer Science Kyungpook National Univ.*
Abstract

Proposed method

- New constrained gamut morphing algorithm
  - Remove spatial variation in 3D color gamut
    - Variation in chromaticity gamut across projector
    - Vignetting effect of each projector
    - Overlap across adjacent projector
  - Resulting in true color seamlessness
  - Adjust intensities of light from each pixel of each projector
    - Achieve smooth morphing from one projector’s gamut to others through overlap region
Introduction

◆ Tiled multi-projector display
  – Reason of spatial color variation in multi-projector display
    • Spatial variation in color gamut across display
    • Classify three different categories
      – Intra-projector variation within single projector
      – Inter-projector variation across different projectors
      – Overlap variation
– Previous method
  • Gamut matching method
    – Achieve color balancing across projector
    – Ignore intra-projector luminance and chrominance variation
    – Reducing color quality and resolution of display
  • Matching luminance transfer functions
    – Achieve luminance balancing across projectors
    – Ignore both inter and intra projector chrominance variations
  • Blending or feathering technique
    – Ramping intensity from each projector smoothly from 0 to 1 in overlap region
    – Ignore both intra and inter projector chrominance and luminance variation
• Majumder and Stevens
  – Addressing spatial variation of luminance
    » Achieve perceptually smooth variation across display
  – Not addressing spatial variation in chrominance

– Proposed method

• Address spatial variation in both luminance and chrominance in tiled projection-based displays
• Morph spatially varying color gamut of display in smoothly constrained manner
  – Retaining white point
• Gamut morphing
  – Smooth morphing of chrominance gamut
  – Smoothing of luminance
– Resulting image

![Curved Screen and Planar Screen Images](image.png)

**Fig. 1.** This figure shows our results on two multi-projector displays on a curved screen (left) and a planar screen (right).

– Comparison of previous work

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Intra Lum</th>
<th>Intra Chr</th>
<th>Inter Lum</th>
<th>Inter Chr</th>
<th>Overlap Lum</th>
<th>Overlap Chr</th>
</tr>
</thead>
<tbody>
<tr>
<td>[15, 16, 11, 17]</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[8]</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[14, 6]</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>[10]</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Our Method</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
CIE XYZ color space

- Color in CIE XYZ color space
  - Defining 3D coordinate \((X, Y, Z)\)
- \(Y\)
  - Luminance of color
- Chrominance of color
  - Chromaticity coordinate

\[
\begin{align*}
(x, y) &= \left( \frac{X}{X+Y+Z}, \frac{Y}{X+Y+Z} \right) \\
(X, Y, Z) &= (xB, yB, (1-x-y)B)
\end{align*}
\]

\(B = X + Y + Z\) which we call the tristimulus brightness.
— Scaling of color

• All colors lying on vector \((X, Y, Z)\)

\[ k(X, Y, Z) \]

where \(k\) is a scale factor.

— Addition of colors

• Addition of two colors, \((X_1, Y_1, Z_1)\) and \((X_2, Y_2, Z_2)\)

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

• Y and B of new color

\[ Y_3 = Y_1 + Y_2; \quad B_3 = B_1 + B_2 \] (4)

• Chrominance of new color

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

• Barycentric coordinates of new chrominance

\[ \frac{B_1}{B_1 + B_2} \text{ and } \frac{B_2}{B_1 + B_2} \]
Definition

- Display made of M projectors
  - Each denoted by $P_j$, $1 \leq j \leq M$
- Relating display coordinate $(s,t)$ to coordinate $(p_j,q_j)$ of projector $P_j$
  $$(s,t) = G_j(p_j,q_j)$$
- Color formed by channel input $i_l = 1$
  $$(X_l,Y_l,Z_l)$$
  $$B_l = X_l + Y_l + Z_l$$
  - Assuming linear displays
    - Changing $i_l$ from 0 to 1
      $$i_l(X_l,Y_l,Z_l)$$
– Remaining chrominance
– Scaling $B_i$

– **3D color gamut of display**
  • 2D chrominance gamut
    – Given by triangle T given by $(x_l, y_l)$
  • Tristimulus brightness of white
    $$B_W = \sum_l X_l + Y_l + Z_l = \sum_l B_l$$
  • Chrominance of white
    $$(x_w, y_w) = \sum_l \frac{B_l}{B_W} (x_l, y_l)$$ (6)
Single projector display

- Vignetting effect
  - Spatial fall off of brightness from around center to fringes of projector
  - Channel independent effect
    - Affecting $B_i$ of all different channels in same manner
      
      \[
      B_i(p, q) = V(p, q) B_i
      \] (7)
      
      where $V(p, q)$ is the vignetting of the projector at pixel $(p, q)$.
  
  - Spatially varying tristimulus brightness of white
    
    \[
    B_w(p, q) = \sum_i V(p, q) B_i
    \] (8)
◆ Multi-projector display

– Overlap of N projectors and coordinate \((s,t)\)

  • \(B_l\) for each channel
    \[
    B_l(s,t) = \sum_{j \in N} V_j(p_j, q_j) B_{l_j}
    \]  
  
  • Spatial variation in \(B_w\)
    \[
    B_w(s,t) = \sum_l \left( \sum_{j \in N} V_j(p_j, q_j) B_{l_j} \right)
    \]  

  • Spatially varying chrominance gamut \(T(s,t)\)
    – Defining three primaries
    \[
    (x_i(s,t), y_i(s,t)) = \sum_j \frac{B_l(s,t)}{B_w(s,t)} (x_{l_j}, y_{l_j})
    \]
Per projector white point balancing

- Desired white point

\[ \frac{\sum_i \alpha_i B_i (x_i, y_i)}{\sum_i \alpha_i B_i} = (x_D, y_D) \]  \hspace{1cm} (12)

where \( \alpha_i \) is a per-channel scale factor, \( 0 \leq \alpha_i \leq 1 \).

- Procedure
  - Fix \( \alpha_r = 1 \)
  - Solve two linear equations
Per pixel chrominance gamut morphing

– Morphing two-dimensional chrominance gamut
  • Chrominance of primaries
    \[ r_k = (1 - \tau) R_1 + \tau R_2 \]  
    \[ g_k = (1 - \tau) G_1 + \tau G_2 \]  
    \[ b_k = (1 - \tau) B_1 + \tau B_2 \]

  where \( 1 - \tau \) is given by the proportions of the is a per-channel scale \( B_i \) of the channel.

  • Scale factor \( \beta_{l_1} \) and \( \beta_{l_2} \) between 0 and 1
    \[ \frac{\beta_{l_1} B_{l_1}}{\beta_{l_1} B_{l_1} + \beta_{l_2} B_{l_2}} = 1 - \tau \]  
    \[ \frac{\beta_{l_2} B_{l_2}}{\beta_{l_1} B_{l_1} + \beta_{l_2} B_{l_2}} = \tau \]

  – Ruin white point balancing
• Retain white point by computing one common factor
  – Seeking $\beta_1$ and $\beta_2$

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

$$\frac{\beta_2 B_{l_2}}{\beta_1 B_{W_1} + \beta_2 B_{W_2}} = \tau$$  \hspace{1cm} (19)

• Transition from $T_1$ to $T_2$ through $n$ steps

$$T_1 \rightarrow t_1 \rightarrow t_2 \cdots t_n \rightarrow T_2$$

$$n = \frac{\max (|R_1 R_2|, |G_1 G_2|, |B_1 B_2|)}{\delta}$$  \hspace{1cm} (20)
**Fig. 2.** The morphing of the chrominance gamut in the horizontal and vertical direction in a display made of rectangular projectors and overlaps.

**Fig. 3.** The chrominance gamut morphs with 2 intermediate steps (across 2 overlapping pixels)
– Chrominance gamut morphing step
  • Generate two attenuation maps for each projector $P_j$
– Final attenuation map
  \[
  \beta_j(p_j, q_j) = \beta_j^H(p_j, q_j) \times \beta_j^V(p_j, q_j)
  \]
– New $B_w$
  • Following application of attenuation map
  \[
  B_{w_c}(s, t) = \sum_{j \in N} \left( \sum_l \beta_j(p_j, q_j) B_{l_j} V_j(p_j, q_j) \right)
  \]  \hspace{1cm} (21)

where $N$ is the set of projectors that overlap at pixel $(s, t)$ (Figure 4).
Per pixel perceptual brightness constraining
  – Applying perception based gradient constraint to $B_{WC}(s,t)$
    \[
    \zeta(s,t) = \frac{B_{WE}(s,t)}{B_{WC}(s,t)} \tag{22}
    \]

Per pixel bezier based brightness smoothing
  – Assuring $C^1$ continuity
    • Fit higher order $C^1$ continuous 2D Bezier surface to $B_{WE}(s,t)$
      \[
      \eta(s,t) = \frac{B_{WS}(s,t)}{B_{WE}(s,t)} \tag{23}
      \]
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 smoothened during the course of our method. On the left, we show the attenuation map for a single projector after every step.
◆ Image correction

- Linearize image $I(p,q)$
  - Using gamma function of 2
- Multiply different attenuation map
  - Creating final attenuation map
    $$A_j(p,q) = \beta_j(p,q) \times \zeta_j(p,q) \times \eta_j(p,q)$$
- Multiply $I$ with attenuation map $A_j$
  - Achieving color correction
    $$I_s(p,q) = I(p,q) \times A_j(p,q)$$
– Apply channel dependent white point correction
  • Generating white point corrected image
    
    \[ I_{w_i}(p,q) = \alpha_l I_{s_i}(p,q) \]

    where \( l \in \{r,g,b\} \).

– Final correction
  • Applying inverse of \( h_l \) to \( I_w \)
    – Achieve desired changes in non-linear device
      
      \[ I_{c_i}(p,q) = h_l^{-1}(I_{w_i}(p,q)) \]

      where \( l \in \{r,g,b\} \).
Implementation

◆ Implementation of proposed method on two displays
  – Planar rear projected display of 3x3 array of nine very low-end projectors
  – Cylindrical front projected display of 2x4 array of eight relatively higher-end projectors

◆ Reconstruct spatially varying color gamut
  – Using sRGB camera as sensor
Results

◆ Comparison of proposed method with existing work

Fig. 5. Comparison of our method with existing work on the most difficult case of white on the planar display. Before any correction; After simple RGB blending; After applying Majumder and Stevens 2005 photometric seamlessness algorithm; after our gamut morphing algorithm.
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; after our gamut morphing algorithm
Fig. 7. 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.
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.
Discussion

◆ Minimum size of the overlap and $\delta$
  – Maximum LAB distance between adjacent pixels in overlap region

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. Right: Comparison of change in chromaticity coordinates across an overlap region achieved by RGB blending with our chrominance morphing.
Difference from the traditional RGB blending

– Traditional blending methods
  • Feather RGB input of projectors in linear or cosine manner in overlap region

– Proposed chrominance morphing
  • Constrain change in chrominance easily to be within human tolerance
  • Retaining more of brightness of display
◆ Effect on the Display Quality
  
  – Impose some constraints on spatial color variation
    
    • Reduction in brightness of display

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 method and other existing methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean DR Loss</th>
<th>Brightness STD</th>
<th>Chrominance STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Correction</td>
<td>0%</td>
<td>0.044</td>
<td>0.035</td>
</tr>
<tr>
<td>After Chrominance Gamut Morphing</td>
<td>13.49%</td>
<td>0.035</td>
<td>0.0089</td>
</tr>
<tr>
<td>After Perceptual Brightness Constraint</td>
<td>19.33%</td>
<td>0.024</td>
<td>0.0088</td>
</tr>
<tr>
<td>After Bezier Based Smoothing (Our Method)</td>
<td>25.86%</td>
<td>0.019</td>
<td>0.0086</td>
</tr>
<tr>
<td>Majumder &amp; Stevens 2005</td>
<td>10.07%</td>
<td>0.031</td>
<td>0.026</td>
</tr>
<tr>
<td>RGB Blending</td>
<td>7.49%</td>
<td>0.038</td>
<td>0.029</td>
</tr>
</tbody>
</table>
Is the perceptual brightness constraining required?

- Bezier based smoothing
  - Way to fit smooth function to spatially varying brightness

**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.
Conclusion

- Color morphing algorithm for multi-projector displays
  - Smoothing both chrominance and brightness across entire display resulting in true color seamlessness
    - Morphing of 2D chrominance gamut in overlap region
    - Brightness smoothing across entire display
  - Achieving seamlessness even for difficult case of flat colors
– Method to find rectangular projections and overlaps

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

- $C^{-1}$: curves include discontinuities
- $C^0$: curves are joined
- $C^1$: first derivatives are equal
- $C^n$: first through $n^{th}$ derivatives are equal
Perceptual uniformity constraint

\[
\frac{\partial W'_l}{\partial x} \leq \frac{1}{\lambda} \times W'
\]

where \( \lambda \) is the smoothing parameter and \( \frac{\partial W'_l}{\partial x} \) is the gradient of \( W' \), along any direction \( x \).

\[
\frac{|W'[u][v] - W'[u'][v']|}{\sqrt{|u - u'|^2 + |v - v'|^2}} \leq \frac{1}{\lambda} \times W'[u][v], \quad \forall u, v, u', v'
\]
**Beziers curve**

- **Linear Bezier curves**
  \[ B(t) = P_0 + t(P_1 - P_0) = (1-t)P_0 + tP_1, \quad t \in [0,1] \]

- **Quadratic Bezier curves**
  \[ B(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2, \quad t \in [0,1] \]
  \[ B(t) = (1-t)((1-t)P_0 + tP_1) + t((1-t)P_1 + tP_2), \quad t \in [0,1] \]