# 3D Gamut Morphing for Non-Rectangular Multi-Projector Displays

Mahdi Abbaspour Tehrani, Muhammad Twaha Ibrahim, Aditi Majumder *Senior Member, IEEE* and M. Gopi *Senior Member, IEEE*

**Abstract**—In a spatially augmented reality system, multiple projectors are tiled on a complex shaped surface to create a seamless display on it. This has several applications in visualization, gaming, education and entertainment. The main challenges in creating seamless and undistorted imagery on such complex shaped surfaces are geometric registration and color correction. Prior methods that provide solutions for the spatial color variation in a multi-projector displays assume rectangular overlap regions across the projectors that is possible only on flat surfaces with extremely constrained projector placement. In this paper, we present a novel and fully automated method for removing color variations in a multi-projector display on arbitrary shaped smooth surfaces using a general color gamut morphing algorithm that can handle any arbitrarily shaped overlap between the projectors and assures imperceptible color variations across the display surface.

**Index Terms**—Color-correction, Multi-projector Displays, Gamut Morphing, Photometric correction

## 1 Introduction

Multi-projector systems can be used to create immersive high resolution displays on surfaces around us (e.g. a cylinder, wall or a dome). It can also be used to project static or dynamic imagery on real objects (e.g. a 3D relief, a life size statue or a tabletop object) [9], [10], [29] to change their appearance. Using multiple projectors (instead of a single projector) becomes essential in such applications when the projection surface cannot be illuminated with a single projector due to a surround shape or a large size. Therefore, multiple projectors are tiled with some overlap across their boundaries to increase coverage of the projection surface/object. The main goal of such systems is to create a single seamless imagery using multiple tiled projectors.

The key challenge for creating seamless imagery on such displays is to achieve accurate geometric and color registration. Geometric registration entails aligning the projected content (e.g. lines and curves) from multiple projectors. There are a variety of existing methods to achieve accurate geometric registration while calibrating the system (e.g. finding the pose and orientation of the devices), both on planar [5], [21], [22], [31] and non-planar surfaces [1], [2], [3], [12], [23], [26], [29]. However, achieving a color seamless display using such systems is still a big challenge. Spatial color variations across such displays are quite noticeable when using projectors of different make and model, or even projectors of the same make but different lamp ages. The 3D color gamut, that defines the range of colors that can be reproduced by a projector, given by the 1D maximum luminance and 2D chrominance for each channel, varies significantly from one pixel to another across the entire display. Further, the per channel transfer function (commonly called gamma function) can also differ from projector to projector. These result in significant spatial color variation across the multi-projector display.

In this paper, we present the first comprehensive content-independent real-time solution for removing the spatial color variation across a multi-projector display on arbitrary shaped smooth surfaces. Using Laplacian equations and a spherical space transformation that is critical for applying well-defined constraints on the smoothing in real time, we present a novel *gamut morphing* method that smoothly morphs the 3D color gamut volumes from one pixel to another in the display. Our method can address both rectangular and non-rectangular overlaps between projectors, unlike most prior methods, resulting in completely seamless imagery.

### 1.1 Related Work

Majumder and Stevens [16] categorize color variation in multi-projector displays into three categories: intra-projector, inter-projector and overlapping area variations.

*Intra-projector variation* refers to the variation of color in the projection area of a single projector itself. For example, in almost all projectors, the brightness is not constant across the projection area and luminance falls off at the periphery of the projector. This is known as *vignetting*. However, the per-channel chrominance (defined by 2D chromaticity coordinates) and transfer function is usually constant across the projection area of a single projector.

*Inter-projector variation* refers to the color variation of projectors with respect to each other. Both the 3D color gamut and the transfer function of projectors can vary based on their make and model or their lamp age creating the inter projector variation.

- *Mahdi Abbaspour Tehrani is with Genentech. E-mail: ma.tehrani86@gmail.com.*
- *Muhammad Twaha Ibrahim, Aditi Majumder and M. Gopi are with the University of California, Irvine. E-mail: muhammti@uci.edu, {majumder, gopi}@ics.uci.edu*
- *The first two authors contributed equally to this work.*

*Manuscript received May 4, 2023; revised May 4, 2023.*

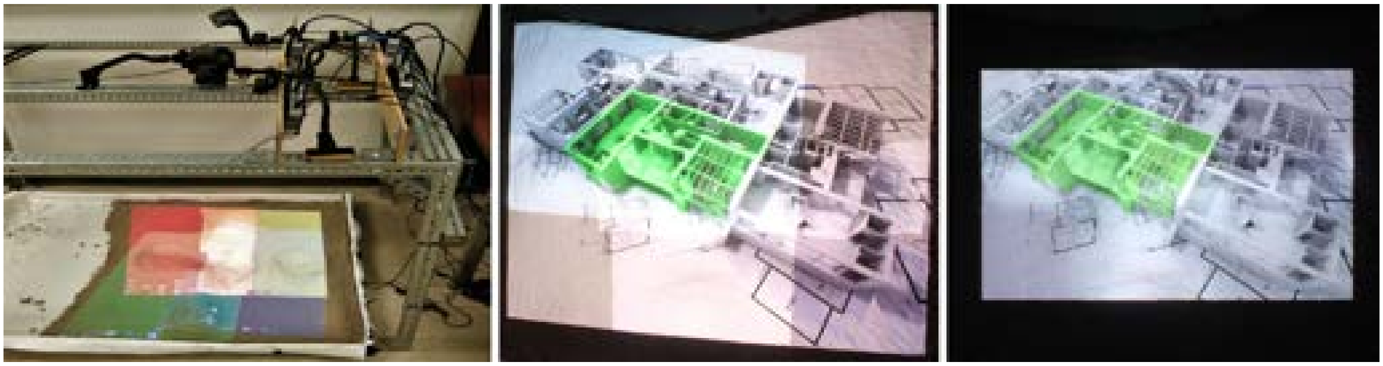


Fig. 1. *Left*: Our setup with four projectors projecting on a sandpit with non-linear geometry. *Middle*: the projected image without photometric correction and *Right*: the display after color calibration using our method.

*Overlapping area variation* results from **(a)** the significantly higher luminance due to addition of light from multiple projectors, and **(b)** the linear combination of the chrominance of the overlapping projectors making them different from the chrominance of the non-overlapping region of each constituting projector.

A large body of prior work have addressed the color variation, each focusing on only *parts of the problem*. Also, some works present content-dependent solution that cannot be implemented in real-time. Though we discuss such works in this section, we compare the proposed work with methods that achieve content-independent color seamlessness in real-time.

**Blending**: [8], [24] only address the overlap variation by blending the RGB values in the overlap region, either via hardware [8] or software [24]. This yields acceptable results only when the multiple projectors have identical 3D gamuts and transfer function at every pixel and therefore no spatial color variation which is almost an impossibility when using large multi-projector displays, especially with consumer devices. Nomoto et al. [18] perform intensity blending in a multi-projector system for dynamic, rigid objects. Assuming that the projectors have been photometrically calibrated to have similar brightness and chrominance, they determine the weight for the contribution from different projector pixels at a point on the surface where pixels from multiple projectors overlap. These weights are proportional to the angle between the ray through the projector pixel and the normal to the surface, whose shape is known. In order to achieve real-time performance, the weights are calculated in parallel.

**Gamut Matching**: [15], [19], [28], [30] address only the inter-projector variation assuming spatially constant color within each projector. [19], [28], [30] assumes identical transfer functions across different projectors and strive to match the 3D gamut (i.e. maximum luminance and chrominance for each channel across the projectors). However, since some colors within the 3D gamut of one projector may be out of the gamut of another projector, first a common gamut is defined as the intersection of the 3D gamuts of all the different projectors. Next, the 3D gamut of each projector is transformed to the common gamut using a linear transformation. [15] additionally matches the transfer functions for each channel. These methods completely ignore the intra-projector color variation that causes the large spatial color variation across the display. Thus, they seldom result in a seamless display. Finally, finding a common gamut is a computationally expensive problem ($O(n^3)$) and therefore cannot scale to a large number of projectors [4].

**Gamut Mapping**: Pjanic et al. [20] presents a content-dependent color gamut mapping method and addresses the challenging case when projector color channels are not independent. First, a color prediction model is computed by adaptively sampling the RGB cube of each projector using a colorimeter. Then, they apply a spatially-varying *gamut mapping* based on the projector overlapping regions. In a post-processing step, they compute the optimal color contribution for each projector pixel by jointly optimizing the color adaptation of all projectors using edge blend weights as a guide to reproduce the target image as closely as possible. This allows one projector to compensate for a target color that may be out-of-gamut of another projector in the overlapping region. As they note, this optimization does not scale well with a large number of projectors due to its non-real-time performance. Therefore they propose another real-time optimization that does not let one projector correct for out-of-gamut colors of another projector in the overlap region.

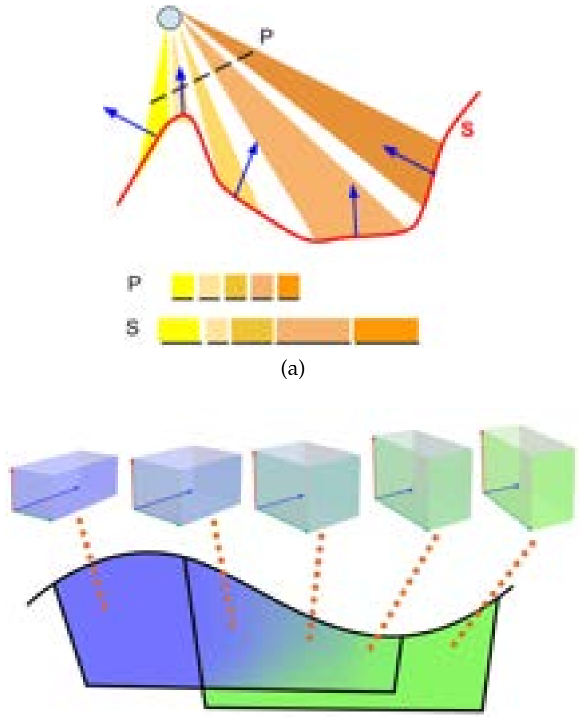
Kurth et al. [11] adjust the colors for dynamic target objects in a single-projector setup in a real-time feedback loop involving a camera. Their system tries to achieve a target appearance and corrects for changes to environmental lighting and surface color, neither of which is assumed to be uniform across the target object. They solve a system of linear equations that compensates for environment light and surface color and continue collecting samples of the object appearance (using the camera) to account for any changes over time. They sampled gamuts at discrete positions of the target object are mapped to achieve the perceptually closest colors that can be reproduced by the projector in order to achieve the target appearance. Since the gamuts are sampled at discrete positions around the target object, they blend them using tetrahedral interpolation to generate the gamuts at every location.

**Smoothing and not Matching**: [13] is the first method to address the spatial color variation. Considering the intra, inter and overlap variation in a unified manner, [13] captures the spatial variation at a high resolution using a camera. It then addresses only the luminance variation by matching the per-channel transfer function of all the pixels to the pixel with the smallest dynamic range since it cannot match the other pixels with higher dynamic range. However, this forces a large number of pixels with high dynamic range to match the dynamic range of the pixels with the smallest range, leading to under-utilization of the aggregated luminance capability of the entire display, compromising its brightness and dynamic range severely and rendering it almost useless when consumer devices with significant vignetting effects are used.

In order to alleviate this situation, [17] provides the unique insight that *complete color matching is not required to create a seamless display*. Instead, [17] smoothly changes the per channel transfer function staying within a perceptual tolerance while constraining every pixel to stay within their dynamic range. This is achieved with a dynamic programming optimization solution that assures retaining the brightness and dynamic range of the display maximally while maintaining imperceptible spatial variation across the display. Therefore, though the luminance response of every pixel is not identical, it leads to a perceptually seamless display, in luminance, that mimics the smooth luminance variation seen in any single projector. This led to the most comprehensive way to address spatial color variation at that time. However, [17] ignores the spatial chrominance variation of the display and therefore the results still show perceptible seams.

[25] further improves this solution by additionally smoothing the chrominance variation across the projectors through the overlap region (still assuming constant chrominance within each projector). This solution achieves smooth morphing of the 3D color gamut volumes from one pixel to another across the entire display, maximizing display quality (i.e. brightness, contrast and color vibrancy) while making spatial variations imperceptible. Though this solution provides astounding results, it can only handle rectangular tiling of projectors where the overlap areas across projectors are rectangular in shape. Such rectangular shaped overlaps are only possible in a very constrained condition when the projectors are placed in a grid-like manner on planar or vertically extruded surfaces with non-oblique projections. However, as we try to illuminate arbitrarily complex geometric shapes using different kinds of projectors, including non-linearly distorted short and ultra short throw projectors, constraining the overlaps to be rectangular is nearly impossible. Further, none of the previously solutions address spatial color variation resulting from the complex geometry.

[27] claims that luminance variation also occurs due to the projector pose (position and orientation) with respect to the surface geometry. This variation occurs because light from projector pixels hits the surface at different angles and gets distributed over different surface areas of the projection surface, leading to a spatial luminance variation due to the geometry (Figure-2a), especially apparent on complex, curved surfaces. [27] explores this luminance variation by presenting a real-time dynamic projection mapping system on complex shaped surfaces by simulating the diffuse direct light transport and providing a smooth blending of luminance across projectors. However, they ignore the spatial chrominance variation across the multi-projector display.



(a)

(b)

Fig. 2. Some challenges of color correction of multi-projector displays: **(a)** Same-sized pixels at the projector (blue circle) image plane (P) cover different areas on the surface (S). The area covered depends on the surface geometry and the distance between the projector and the surface point. Therefore, the color and brightness on the surface varies based on the area covered. **(b)** The variation of 3D color gamut of two projectors. The color of each 3D color gamut in this figure indicates the color of white. We smoothly morph the color gamut in the overlap of projectors to alleviate the perceived color difference.

## 1.2 Our Contribution

In this paper, we propose the first technique to comprehensively address all kinds of luminance and chrominance variations across different arbitrary shaped smooth surfaces illuminated by a multi-projector system. We present a fully automatic gamut morphing algorithm which achieves a smooth morphing of the 3D color gamut across the pixels of the multi-projector display, even in non-rectangular overlaps, resulting in completely seamless imagery on any arbitrary shaped smooth display surface. Table-1 shows the comparison of our method with previous content-independent real-time color correction methods. Note that methods like [20] can optimize the display quality for a particular target image but become impractical when playing a video or any other interactive applications. Our main contributions are as follows:

**Comprehensive Solution**: To the best of our knowledge, this is the first comprehensive color seamlessness solution for multi-projector displays that handles all kinds of spatial color variations on complex Lambertian shapes including intra and inter projector variations and variations due to rectangular and non-rectangular overlaps and complex 3D geometry.

**Scalable and Accurate Solution**: The most difficult aspect of a color seamlessness solution is the color cali-

TABLE 1
This table shows the comparison of our method with existing works.

| Method | Intra-Projector | Inter-Projector | | Overlap | | Geometry-aware |
|---|---|---|---|---|---|---|
| | Luminance | Luminance | Chrominance | Luminance | Chrominance | |
| [19], [28], [30] | | ✓ | ✓ | | | |
| [15] | | ✓ | | | | |
| [8], [24] | | | | ✓ | ✓ | |
| [17] | ✓ | ✓ | | ✓ | | |
| [27] | ✓ | ✓ | | ✓ | | Surface normals |
| [11] | ✓ | | | | | Surface normals |
| [18] | | ✓ | | ✓ | | Surface normals |
| [20], [25] | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Ours | ✓ | ✓ | ✓ | ✓ | ✓ | Surface distance |

bration of the devices that form the multi-projector system. Apriori color calibration for each and every projector makes the method unscalable. Blending approaches avoid this color calibration completely and focus on controlling only the contribution from the projectors. Therefore, they are scalable, but not very accurate. Gamut matching and gamut mapping methods depend heavily on good prior color calibration of the devices and sometimes are content-dependent. Therefore, though relatively accurate, they cannot scale. Our method does not need any prior calibration of the devices making it highly scalable. By morphing the color gamut, we also achieve high perceptual fidelity and therefore, a high-quality display. Finally, our spherical space transformation step allows distributed calculation of gamut morphing across the projectors in a fast, efficient and scalable manner, especially germane for scalable solutions for large number of projectors.

**Improved Results with Geometry-Aware Gamut Morphing**: Our method can be easily made geometry-aware by using 3D distances, instead of 2D distances, when computing the weights for the Laplacian equations. We explored the impact of considering geometry in such a manner (similar to [27]) in our gamut morphing method. This geometry aware morphing provides perceptibly better results, but not significantly better results, in the systems we test (Figure 13). However, we anticipate this improvement to be amplified when the system scales to a large number of projectors and surfaces with large dramatic variations where color variations can be lot more pronounced. Therefore, we recommend using the the geometrically aware morphing since it does not impact any other aspects of the Laplacian and only improves the final display quality.

## 2 Notations

In this section, we define and explain geometric and color notations used in this paper.

**Geometric Registration Representation:** Our multi-projector display consists of $M$ projectors projecting on an arbitrary shaped display surface with some overlap of the projections at their periphery observed by a single high quality color camera. We denote the 2D pixel coordinate of projector $i$, $1 \leq i \leq M$, by $(p_i, q_i)$. We assume that 3D coordinate of the display surface, $(U, V, W)$ and the camera coordinate $(u, v)$ corresponding to $(p_i, q_i)$ is estimated apriori using any geometric calibration method. In our case, we use the method described in [26], [29]. Thus, the 3D coordinate of each pixel is mapped to a projector coordinate $(p_i, q_i)$ by $F_i : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ mapping as

$$(U, V, W) = F_i(p_i, q_i) \tag{1}$$

Further, each pixel $(u, v)$ in camera space is related to the pixel $(p_i, q_i)$ of projector $i$ by $H_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ as

$$(u, v) = H_i(p_i, q_i) \tag{2}$$

**Color Representation:** To define color, we use tristimulus values $X$, $Y$ and $Z$ of CIE 1931 color space. The $Y$ coordinate denotes the *luminance* or perceived brightness. The total energy of the color is given by the total tristimulus value or $ttv$ of $I$, which provides a measure of the brightness and is given by

$$I = X + Y + Z \tag{3}$$

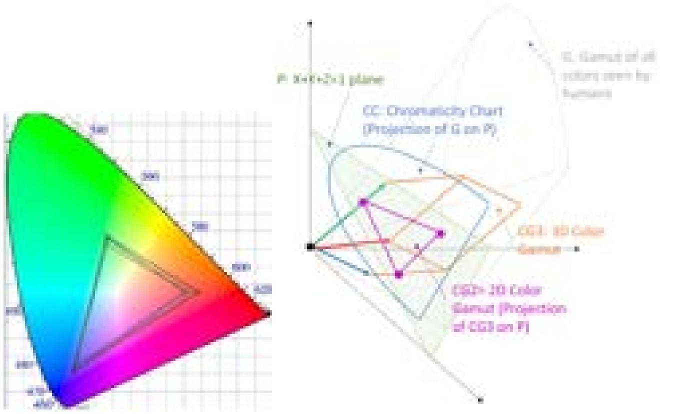
The chromaticity coordinates of the color $(x, y)$ measures the chroma and is given by the proportion of $X$ and $Y$ in $I$, i.e.

$$x = \frac{X}{I} = \frac{X}{X+Y+Z}, \;\; y = \frac{Y}{I} = \frac{Y}{X+Y+Z}. \tag{4}$$

If a point on the display surface is lit by a set of $K$ overlapping projectors, where each projector projects a different color $(X_i, Y_i, Z_i)$. The final color of that point on the display surface is given by the addition of the tristimulus values of the colors from the constituting projector given by $(\sum_{i \in K} X_i, \sum_{i \in K} Y_i, \sum_{i \in K} Z_i)$. The $ttv$ of the resulting color provides a measure of its brightness (in terms of total energy and not perceptible brightness or luminance) and is equal to the sum of the $ttv$ of all the overlapping projectors i.e. $I_R = \sum_{i \in K} I_i$. The chromaticity coordinate of the resulting color is obtained by weighted sum of their chromaticity coordinates, weighted by the proportion of their $ttv$ in the resulting added $ttv$.

$$(x, y) = \sum_{i \in K} \frac{I_i}{I_R}(x_i, y_i) \tag{5}$$

**Color Gamut:** We assume that each projector has three independent color channels i.e. RGB. The tristimulus value of a channel $c$ when it projects its maximum input while others are off is $(X^c, Y^c, Z^c)$. This tristimulus value represents a vector from origin in XYZ color space shown by the red, green and blue vectors in Fig 3b. The color created by normalized channel input $a$, $0 \leq a \leq 1$, when the other two channels are mute are given by the tristimulus


P: X+Y+Z=1 plane
G: Gamut of all colors seen by humans
CC: Chromaticity Chart (Projection of G on P)
CG1: 3D Color Gamut
CG2: 2D Color Gamut (Projection of CG3 on P)

(a)

(b)

Fig. 3. **(a)** The color gamuts of four projectors on the chromaticity chart. **(b)** Illustration of a 3D color gamut of a device.

values $(aX^c, aY^c, aZ^c)$ where $0 \leq a \leq 1$. However, $a$ is usually related to the input by a non-linear function $h_c$, commonly called the *channel transfer function* or *gamma function*. The parallelepiped spanned in XYZ color space by the linear combination of the three vectors $(X^c, Y^c, Z^c)$, $c \in \{R, G, B\}$ includes all the colors that can be produced by the projector [7], [14]. This is called the *3D color gamut* of the device and is shown by the orange volume in Fig 3b. Therefore, the XYZ color of a pixel with input RGB values $(n^r, n^g, n^b)$, $0 \leq n^c \leq 1$, is:

$$(X, Y, Z) = \sum_{c \in \{r,g,b\}} n^c (X^c, Y^c, Z^c) \tag{6}$$

The chrominance of each channel, denoted by $(x^c, y^c)$, is obtained by projecting each of the $(X^c, Y^c, Z^c)$, $c \in \{r, b, b\}$ on the $X + Y + Z = 1$ plane, shown by the pink dots in Fig 3b. Therefore, it creates a triangle in the chromaticity chart known as the *2D color gamut* of the projector shown by the pink triangle in Fig 3b.

The white generated by projector $i$ is given by

$$(X_i^w, Y_i^w, Z_i^w) = \sum_{c \in \{r,g,b\}} (X_i^c, Y_i^c, Z_i^c) \tag{7}$$

The chrominance of white $(x_i^w, y_i^w)$, called the *white point*, plays an important role in the color variation and is calculated as:

$$(x_i^w, y_i^w) = \sum_{c \in \{r,g,b\}} \frac{I_i^c}{I_i^w} (x_i^c, y_i^c) \tag{8}$$

where $I^c = X^c + Y^c + Z^c$ is the brightness of color channel $c$ and $I^w = I^r + I^g + I^b$ is the brightness of white.The white point of a projector is decided by the proportion of the red, green and blue in its white and is dependent of the relative length of the red, green and blue vectors with respect to each other.

Equations 5 and 8 provide simple linear equations to combine chrominances, white point and chrominance gamuts in CIE XYZ space using $ttv$ which can be directly linked to channel intensities. Such a simple equation is not available when comparing other non-linear perceptually uniform color spaces like CIELab or CIELuv. This motivates us to work in the simpler domain of CIE XYZ space. We strive towards seamlessness at all viewing conditions and therefore don't use RLab space which tunes results based on viewing conditions.

**Intra-projector spatial color variation:** The intra-projector color variation in a multi-projector display arises from the following variations. First, $(X^c, Y^c, Z^c)$ can vary across the projected area, especially in its magnitude, even when projecting on a flat surface from a single device. The $ttv$ of each channel (and therefore of resulting white) show a fall-off from center to periphery, often called *vignetting* or *hot-spot* effect. Mathematically, the intensity at a pixel $(p_i, q_i)$ of projector $i$ is

$$I_i^c(p_i, q_i) = V_i(p_i, q_i) \bar{I}_i^c(p_i, q_i), \quad c \in \{r, g, b\} \tag{9}$$

where $\bar{I}_i^c$ is the original intensity, $V(p, q) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the vignetting function. Due to this spatial variation in the magnitude of $(X^c, Y^c, Z^c)$, the 3D color gamut of a projector is not the same at every pixel. However, the direction of $(X^c, Y^c, Z^c)$ is constant within a projector which leads to a spatially constant 2D color gamut within a projector. Finally, the channel transfer function is usually constant within a projector. We assume that the vignetting affects only the brightness and not the chrominance. This assumption works reasonably well for most consumer grade projectors.

**Inter-projector spatial color variation:** Across projectors, both the direction and magnitude of $(X^c, Y^c, Z^c)$ varies. Therefore, both the 3D and 2D color gamuts can vary significantly, even across projectors of same make and model. Figure-3a shows the 2D color gamuts of 4 different projectors. These variations create the significant perceptible spatial color variation across a multi-projector display, especially when projecting flat colors.

**Overlap Variation:** In areas where $|K| > 1$ projectors overlap, $I_c$ scales almost by $|K|$ leading to a highly bright region. In addition, the chrominance is combined based on the relative proportion of channel $ttv$ coming from the overlapping projectors. Therefore, the 2D and 3D gamut both vary spatially in the overlap region.

**Variation due to Geometry:** All the aforementioned spatial color variation can be further accentuated by non-linear surfaces. The same $ttv$ coming from a pixel can be stretched over different surface areas due to complex surface geometry. It also depends on the distance between the projector and the area illuminated by a pixel, which in turn, depends on the surface geometry and the position of the projector relative to it. However, note that geometry related variation can only impact the brightness and does not introduce a chrominance variation, unless it is in the overlap region. When exploring geometry aware color gamut morphing in subsequent sections, we use the 3D distance between neighboring 3D points.

# 3 Algorithm

Figure-4 shows a typical example of color variation in a multi-projector display on a complex geometry made by the sand pit. Notice the differing white points and the brighter overlaps with color variations. Our starting point is the gamut morphing method of [25] that requires rectangular overlaps. We present an improved gamut morphing method

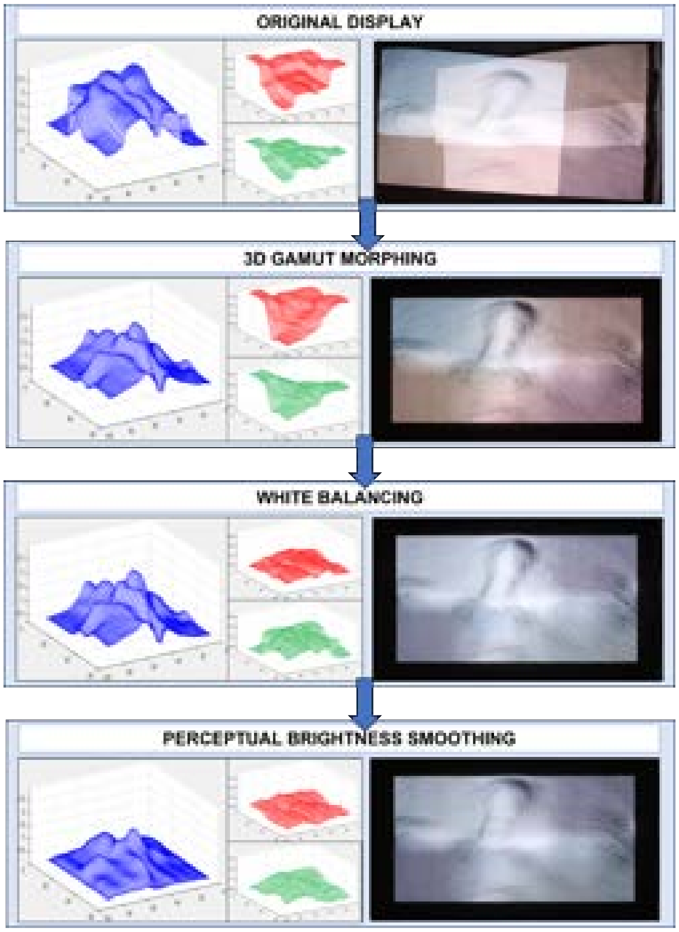


Fig. 4. The complete flowchart of our algorithm. Each projector is projecting white. We show the effect of each step on the spatial variation of the brightness (blue), the chromaticity coordinate x (red) and y (green) across the display. Note the 3D gamut morphing smooths the shape changes in the chromaticity coordinates in the overlap region, but since the difference in the whites are still significant, the seams are visible in the overlap region. The white balancing reduces the variation in the white across the projectors making the seams much less perceptible by the brightness variation is still evident which is finally smoothed by the constrained brightness smoothing.

that lifts the restrictions of rectangular overlap regions in the next section. Note that our method assumes a smooth 3D surface shape without sharp discontinuities and does not take the surface normals into account.

### 3.1 3D Gamut Morphing

The reason single projector displays don't show severe variation on flat surfaces is due to the fact that there is no intra-projector variation of 2D gamut and only brightness varies. The brightness variation may become worse for non-flat surfaces, but 3D gamut still remains constant within the projector. The variation of 3D gamut impacts the overlap adversely. Therefore, our goal is to morph the 3D color gamut smoothly in the overlap area of the projectors to transition from the spatially constant 3D gamut of one projector to another. Following this 3D gamut morphing, the remaining variation is due to brightness only. Therefore, applying a constrained brightness smoothing achieves the total 3D gamut morphing. The proposed method morphs the color gamut of projectors across the 3D geometry of the display surface, followed by white balancing and constrained brightness smoothing (Section-3.2) of entire display. Figure 2b illustrates 3D gamut morphing in the overlap area of the projectors. Figure 4 shows the steps of our algorithm.

Let $Q$ be a point on the display surface which lit by a set of $K$ overlapping projectors, $|K| \leq M$. Further, let $(x_i^c, y_i^c)$, $c \in \{r, g, b\}$ denote the spatially invariant 2D chrominance coordinates of projector $i \in K$, $1 \leq i \leq |K|$. Since the multi-projector setup has been geometrically calibrated, all projectors project the same RGB value on their corresponding pixels $(p_i, q_i)$ that project on 3D point $Q$, where $Q = F_i(p_i, q_i), i \in K$. The chrominance at point $Q$ lit by the set of projectors $K$ can be calculated as:

$$(x^c, y^c) = \sum_{i \in K} \alpha_i^c(p_i, q_i) \cdot (x_i^c, y_i^c), \tag{10}$$

where $\alpha_i^c(p_i, q_i)$ is the ratio of the brightness of projector $i$ to the sum of the brightness of all projectors at point $Q$ given by

$$\alpha_i^c(p_i, q_i) = \frac{I_i^c(p_i, q_i)}{\sum_{j \in K} I_j^c(p_j, q_j)}. \tag{11}$$

Note that $\sum_{i \in K} \alpha_i^c(p_i, q_i) = 1$. Further, since the chrominances of the different channels define the 3D color gamut, the 3D color gamut at point $Q$ is a linear combination of the 3D gamuts of overlapping projectors at that point.

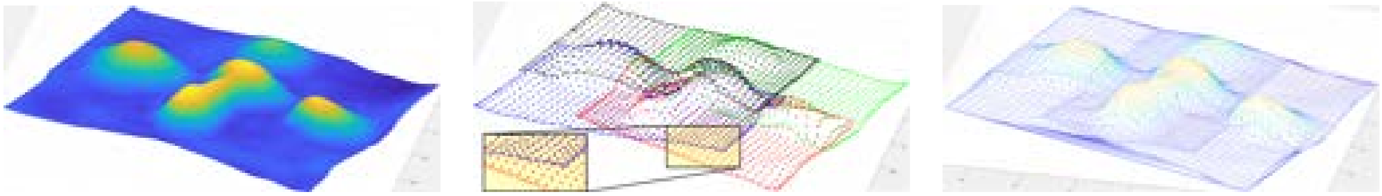
We smoothly morph the 3D color gamuts of non-overlapping region of one projector to another through the overlap areas by precisely controlling the brightness from each channel of each projector at every pixel in the overlap. This is achieved if $\alpha_i^c$ changes smoothly across the projector overlapping regions. In order to achieve this, we introduce a per-pixel attenuation factor $\gamma_i^c(p_i, q_i)$ to attenuate the brightness of each channel at every pixel of projector $i$ in the overlapping region such that:

$$\alpha_i^c(p_i, q_i) = \frac{\gamma_i^c(p_i, q_i) \cdot I_i(p_i, q_i)}{\sum_{j=1}^{M} \gamma_j^c(p_j, q_j) \cdot I_j(p_j, q_j)}, \tag{12}$$

Uniform distribution of pixels in the projector space may not lead to uniform distribution of pixels on the surface geometry which depends on the shape of the surface geometry and the angle of projection. We account for this non-uniform sampling of the the surface geometry by the projector pixels in three ways. When determining $\gamma_i^c(p_i, q_i)$, we apply a constrained graph construction, followed by spherical space transformation, followed by the computing the per-pixel scale factor for each projector. The graph based representation helps us address arbitrarily shaped overlap regions.

#### 3.1.1 *Constrained graph construction*

After geometric calibration of the system, using the mapping $F_i(p_i, q_i)$, we can find the corresponding 3D point for every pixel of projector $i$ to find a dense 3D point cloud of the part of the surface geometry illuminated by each projector $i$. Let us denote this dense point cloud by $H_i^d$. Our goal is to find a factor $\gamma_i^c(p_i, q_i)$ for every point in $H_i^d$. In order to achieve this, we first sample a sparser set of the 3D points in $H_i^d$, denoted by $H_i^s$ and calculate the attenuation factor for this sparser set of 3D points. Next we find the attenuation factor from $H_i^s$ at every point of $H_i^d$ by

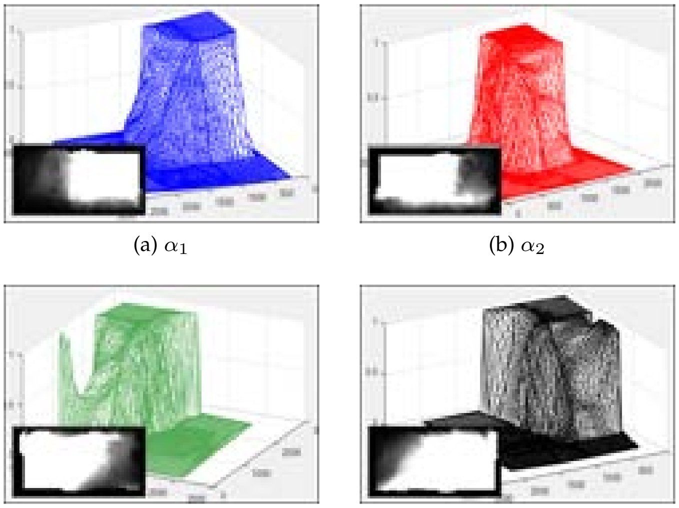


(a) Surface geometry $H^d$

(b) Sampled surface $H^s$

(c) Triangulated mesh $G$

Fig. 5. Constrained graph construction. **(a)** The dense reconstructed surface $H^d$ is sampled to generate **(b)** the sampled surface $H^s$ for each projector.Notice that the sampling density is higher at the boundaries than in the middle (see the zoomed in view in the rectangle). This sampled surface is used to introduce edge constraints to generate **(c)** the triangulated mesh $G$.



(a) $\alpha_1$

(b) $\alpha_2$

(c) $\alpha_3$

(d) $\alpha_4$

Fig. 6. The computed $\alpha$'s for each projector by transforming to spherical space and solving the Laplacian (see Section-3.1.2). The weights used in solving the Laplacian depend on the 3D geometry. The insets show the corresponding projector masks.

interpolation. To create $H_i^s$, we sub-sample points from $H_i^d$ (in the projector image space) at a higher frequency at the boundary than in the interior of the projector. Therefore, in $H_i^s$, the boundary is more densely sampled than the interior. Figure-5(a-b) provides a visualization for $H^d$ and the denser sampling at the boundaries for each $H_i^s$.

We then gather all the points of $H_i^s$ of all projectors to create a 3D point cloud $H^s$ for the entire projection surface. Note that due to denser sampling at projection boundaries $H^s$ does not have uniform density of points. Next, we apply constrained Delaunay triangulation on $H^s$ to create a mesh $G$ (see Figure-5c). Using the constraints, we force the 3D points on a projector boundary to connect to the closest point on boundary of the same projector. This is to ensure no edges in $G$ cut across the boundary of a projector i.e. a 3D point *inside* the projection area of one projector is not connected to a 3D point *outside* the projection area of the same projector.

Recall that the 2D color gamut of each 3D point in $H^s$ should be a linear combination of the 2D gamuts of the overlapping projectors at that point. We assign a $M$-dimensional vector $V = (\alpha_1, \alpha_2, \ldots, \alpha_M)$ to each 3D point in $G$ where $\sum_{i=1}^{M} \alpha_i = 1$. For non-overlapping area of the projection surface covered *only* by projector $i$, all entries of $V$ is 0 except $\alpha_i = 1$. Therefore, the vector $V$ is known for at the 3D points illuminated by only one projector. However, if a point is illuminated by $K$ overlapping projectors $\alpha_i$ for such a point is

$$\alpha_i = \begin{cases} 0 & i \notin K \\ \textit{unknown} & i \in K \end{cases} \tag{13}$$

Our goal is to estimate unknown parameters of the vector $V$ for all the 3D points in $G$ such that $\alpha_i$ changes smoothly across $G$ respecting the 3D geometry of the display surface.

In order to achieve this, we consider $G$ as a graph and the 3D points its nodes. We assign a weight $w_{RT}$ to each edge connecting two 3D points, $R$ and $T$, that is equal to the Euclidean distance between the 3D coordinates of $R$ and $T$ i.e.$w_{RT} = ||R - T||$. Now, we construct and solve a weighted Laplacian equation solving for the $\alpha$s over $G$ to smoothly change $\alpha$s across the graph. Our boundary conditions are the known values of $V$ for the nodes in the non-overlapping regions of the projectors.

However, formulating the problem as a weighted Laplacian poses a few challenges. First, the solution will not guarantee $\sum_{i=1}^{M} \alpha_i = 1$ for $V$ at each node. Second, the boundary conditions create dependencies across the $\alpha$s at different nodes of $G$. This prevents us from calculating them in a distributed manner. Third, the time complexity of solving the Laplacian equation increases with increase in the number of projectors. Finally, when adding or removing a projector to the system, we have to recalculate all $\alpha_i$s, $1 \leq i \leq M$ again even if projector $i$ did not have any overlap with the added or removed projector. We address this problem using posing this problem after a spherical space transformation as explained in the following section.

### 3.1.2 Spherical-space Transformation

To introduce the constraint $\sum_{i=1}^{M} \alpha_i = 1$ to satisfy Equation-11, instead of the vector $V$ we use vector $V' = (\sqrt{\alpha_1}, \sqrt{\alpha_2}, \ldots, \sqrt{\alpha_M})$. Therefore, using a new constraint of $||V'|| = 1$, we can ensure that the sum of $\alpha_i$s will be equal to 1 at every node of $G$ since

$$||V'|| = \sqrt{(\sum_{i=1}^{M} \alpha_i)} = 1 \tag{14}$$

Next, we transform $V'$ from Cartesian coordinate system to $M$-dimensional spherical coordinate space comprising of a radial coordinate $r$ and $(M-1)$ angular coordinates $(\phi_1, \phi_2, \ldots, \phi_{M-1})$ as follows:

$$\begin{aligned}\sqrt{\alpha_i} &= r\cos\phi_i \prod_{k=1}^{i-1}\sin\phi_k, \quad 1 \le i \le M-1 \\ \sqrt{\alpha_M} &= r\prod_{k=1}^{M-1}\sin\phi_k\end{aligned} \tag{15}$$

Note that using above equation, $r = \sqrt{(\sum_{i=1}^{M}\alpha_i)}$. Imposing a constraint of $||V'|| = 1$ can be achieved by making $r = 1$ at all nodes of $G$.

Therefore, we have converted the vector $V$ at each node $t$ of $G$ into a $(M-1)$-dimensional vector $\phi^t = \{\phi_1^t, \phi_2^t, \ldots, \phi_{M-1}^t\}$ in spherical space. For each node $t$, $\phi_i^t$ is known in the non-overlapping regions (either 0° or 90°) and unknown in the overlapping regions. Therefore, our goal is to determine the value of each unknown $\phi_i^t$ such that it smoothly morphs over the other nodes of $G$ across the overlapping regions. We achieve this by solving a weighted Laplacian equation for each $\phi_i$:

$$\begin{gathered}\boldsymbol{b} = \boldsymbol{A}\Phi_i, \\ \boldsymbol{A}(t,t) = \begin{cases}1 & \phi_i^t \text{ is known} \\ -1 & \phi_i^t \text{ is unknown}\end{cases} \\ \boldsymbol{A}(t,r) = \begin{cases}\Upsilon(t,r) & r \in N(t), r \ne t \\ 0 & \text{otherwise}\end{cases} \\ \boldsymbol{b}(t) = \begin{cases}\phi_i^t & \phi_i^t \text{ is known} \\ 0 & \phi_i^t \text{ is unknown}\end{cases} \\ 1 \le i \le M-1 \\ 1 \le t, r \le |G|\end{gathered} \tag{16}$$

where $\phi_i^t$ is the value of $\phi_i$ at node $t$, $\boldsymbol{A} \in \mathbb{R}^{|G|\times|G|}$ and $\boldsymbol{b} \in \mathbb{R}^{|G|\times 1}$. $\Phi_i \in \mathbb{R}^{|G|\times 1}$ contains the final computed values of $\phi$ for each node. $N(t)$ is the set of all neighbors of node $t$. The function $\Upsilon(t,r)$ is the weighted average of the 3D distance between node $t$ and $r$ to the sum of the 3D distances between node $t$ and all its neighbors. It is defined as:

$$\Upsilon(t,r) = \frac{||T-R||}{\sum_{s\in N(t)}||T-S||} \tag{17}$$

where $R, S, T$ are the 3D coordinates at nodes $r, s, t$ respectively, and $\sum_{s\in N(t)}\Upsilon(t,s) = 1$. We solve Equation-16 for $\Phi_i$ and transform the result from spherical-space back to Cartesian-space to find the value of each $\alpha_i$ in the vector $V$.

### 3.1.3 *Finding per-pixel scale factor*

In the previous step we calculated the value of vector $V = (\alpha_1, \alpha_2, \ldots, \alpha_M)$ for nodes in $G$ and therefore all 3D points in $H^s$. As shown in Equation 11, the value of $\alpha_i$ is equal to ratio of brightness of projector $i$ to the sum of the brightness of all projectors at each point for each color channel. However, scaling each color channel individually as in Equation-15 will change the color of white across the projector as well. This will result in a spatially varying white balance that can be detected very easily by humans.

Therefore, like [25], we find one common scale factor for all channels by scaling the brightness of white. This still assures a smooth morphing of color gamuts across the display surface. We need to find the per-pixel scale factor $\gamma_i$ to scale the brightness of each projector pixel such that we achieve the desired $\alpha_i$ at each point:

$$\alpha_i(p_i,q_i) = \frac{\gamma_i(p_i,q_i)I_i^w(p_i,q_i)}{\sum_{j=1}^{M}\gamma_j(p_j,q_j)I_j^w(p_j,q_j)} \tag{18}$$

The above system of equations is an under-constrained system for computing $\gamma_i, 1 \le i \le M$. For this reason, we fix the value $\gamma_i(p_i,q_i) = 1$ for one of the projectors and then compute $\gamma_j(p_j,q_j), j \ne i$ for all other projectors. If any $\gamma_j > 1$, we fix the $\gamma_i$ value to 1 for another projector and repeat the calculations. Once we have the scale factors for all the sparse point cloud $H^s$, we perform interpolation to find the attenuation factors $\gamma_i(p_i,q_i)$ for all 3D points in the dense point cloud $H^d$. Finally, using $F(p_i,q_i)$ we find the $\gamma$s at every pixels in projector $i$, $1 \le i \le M$.

## 3.2 White Balancing and Brightness Smoothing

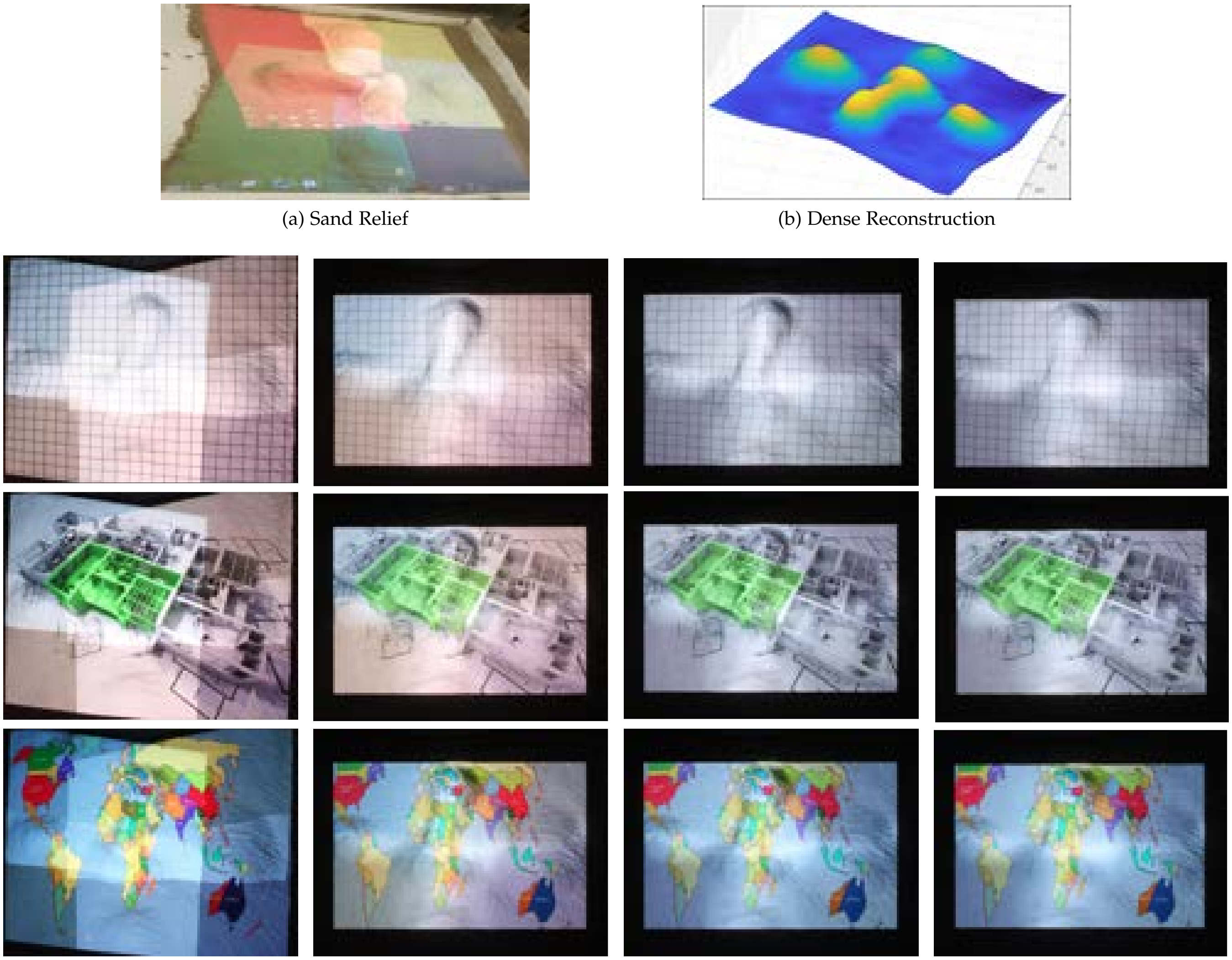
The previous steps ensure smooth 3D gamut morphing on the arbitrary shaped display based on the surface geometry. Figure 4: *3D Gamut Morphing* shows the result after gamut morphing using the 3D geometry. Note that the variation of the white color and brightness is still noticeable over the display surface after the 3D color gamut morphing. In order to reduce this variation, we perform projector white balancing [25] which gets all the projectors to have similar whites (Figure-4: *White Balancing*). Finally, we perform perceptual brightness smoothing (Figure-4: *Brightness Smoothing*) to achieve a completely seamless display.

### 3.2.1 *White Balancing*

Since human visual perception is more sensitive to white, a viewer can notice the difference of white points of projectors across the surface very easily [6], [25]. Hence, we want to ensure a constant white color over the entire display surface. For this purpose, we used the method proposed in [25] to find a per-channel scale factor $\psi_i^c,\ c \in \{r,g,b\}, 1 \le i \le M$, for each projector to achieve the same white point across all projectors. This is called *white balancing*. As shown in Equation-8, white balancing changes the chrominance of white by scaling the brightness of each color channel, even if the projectors have different color gamuts. Therefore, this step does not undo the effect of the gamut morphing. Figure-4: *White Balancing* shows the display after white balancing.

### 3.2.2 *Brightness Smoothing*

Although the 2D color gamut changes smoothly over the display surface, the brightness variation is still noticeable (Figure-4: *3D Gamut Morphing* and *White Balancing*). In this step, we apply a perceptually constrained brightness smoothing presented in [17] followed by Bezier surface fitting to achieve $C_1$ continuity in the spatial brightness variation. The variation is also controlled in such a manner that it is imperceptible to the human eye and is determined by the brightness smoothing parameter $\lambda$. The value of $\lambda$ depends on the angle subtended by the pixels on the human eye and therefore, depends both on the pixel density of the display and the distance of the viewer from the display. A



(a) Sand Relief

(b) Dense Reconstruction

(c) Uncorrected

(d) 3D gamut morphing

(e) White balancing

(f) Brightness smoothing

Fig. 7. Results at each step of the algorithm when projecting different content on **(a)**: a sand relief with four projectors having non-rectangular overlaps. **(b)**: The dense reconstruction of the surface. **(c)**-**(f)**: (*top-row*) white grid, (*middle-row*) building plan and (*bottom-row*) world map. Notice that even with colored content, the difference in the color gamuts of each projector is noticeable, especially in the low-frequency regions of the content (e.g. the white background in the building plan). When 2D gamut morphing is performed, the transition of color is smoothed, but is still easily perceptible in low-frequency areas of the projected content, e.g. the blue ocean in the world map image. After white balancing, the colors of the four projectors are a lot more consistent and with perceptual brightness smoothing, the entire projection is seamless.

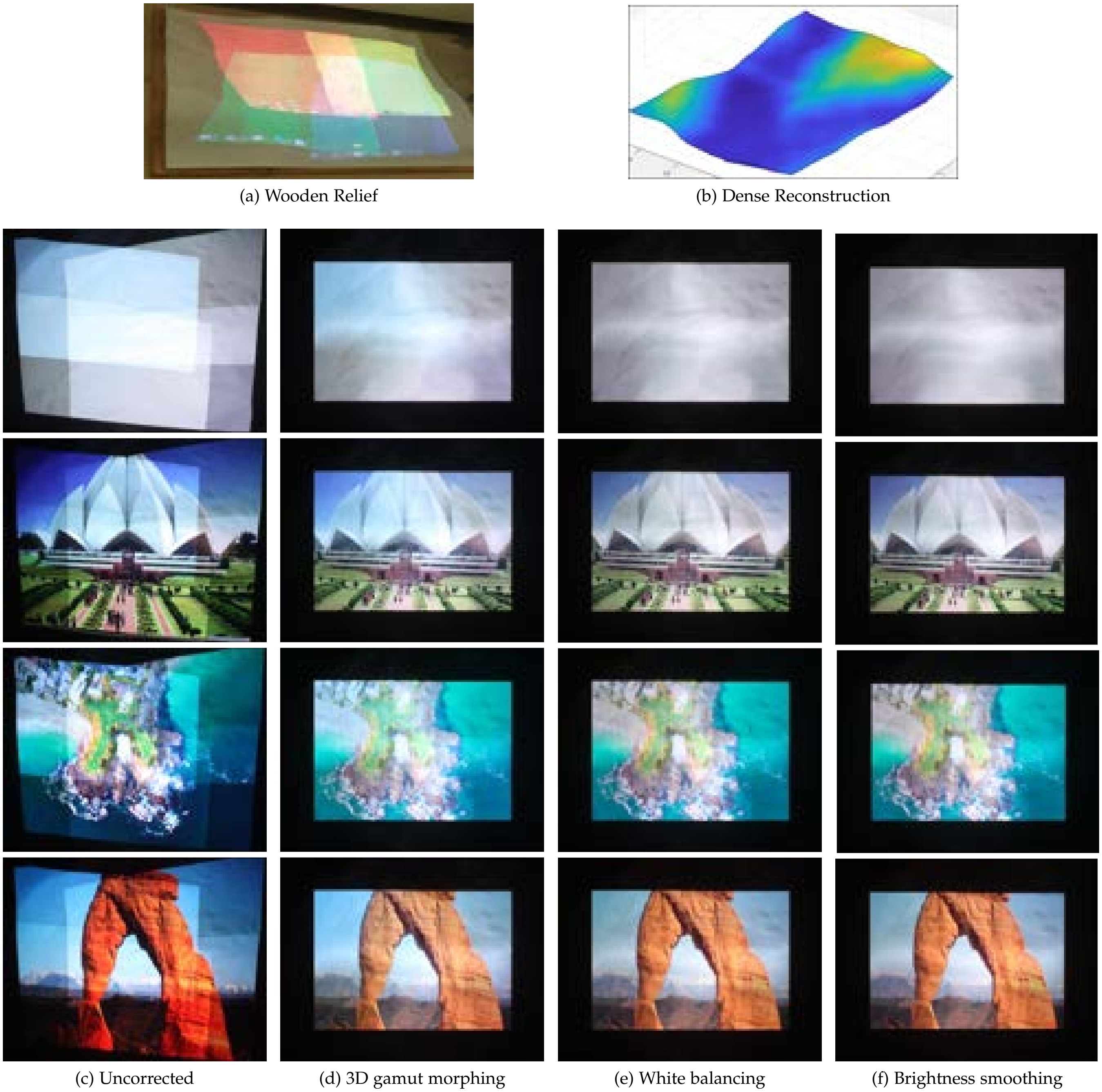
higher value of $\lambda$ ($\sim$100-200) results in perceptibly smoother display and is more suitable when the display is viewed from a larger distance. This step gives us an attenuation map $\Omega$ for each 3D point on the display surface. Using the inverse of the function $F_i(p_i, q_i)$, we transform $\Omega$ to each projector's coordinate space to get the attenuation map $\omega_i(p_i, q_i)$ for each projector. Since this attenuation is the same across channels, it does not remove the effect of the prior steps.

### 3.3 Seamless Imagery

Finally, to achieve seamless imagery, we have to change the input image of each projector based on the attenuation maps generated from the previous steps. Each projector has two per-pixel attenuation maps: $\gamma_i(p_i, q_i)$ generated by the gamut morphing step, and $\omega_i(p_i, q_i)$ generated by brightness smoothing. There is also a per-channel scale factor $\psi_i^c$ from the white balancing step. We need to scale the value of each input pixel using these attenuation maps. Assume $A_i^c(p_i, q_i)$ is the value of the channel $c,\ c \in \{r, g, b\}$, of pixel $(p_i, q_i)$ for projector $i$ after geometric registration. Therefore, the value of this pixel after photometric registration, denoted by $B_i^c(p_i, q_i)$ is:

$$B_i^c(p_i, q_i) = h_i^{-1}(\psi_i^c \cdot \omega_i(p_i, q_i) \cdot \gamma_i(p_i, q_i) \cdot A_i^c(p_i, q_i)) \quad (19)$$

where $h_i^{-1}$ is the inverse transfer function of the projector $i$. Since the projector is not a linear device, it is necessary to recover its transfer function so that all the attenuations are applied after the projector is linearized. We recover the transfer function using the method proposed in [17] that entails projecting different shades of color for each projector and capturing them with the camera followed by an image processing step to recover the transfer function.



(a) Wooden Relief

(b) Dense Reconstruction

(c) Uncorrected

(d) 3D gamut morphing

(e) White balancing

(f) Brightness smoothing

Fig. 8. Results at each step of the algorithm when projecting on **(a)**: a white wooden relief with four projectors having non-rectangular overlaps. **(b)**: The dense reconstruction of the surface. **(c)**-**(f)**: From top to bottom: white image, temple, laguna and canyon.

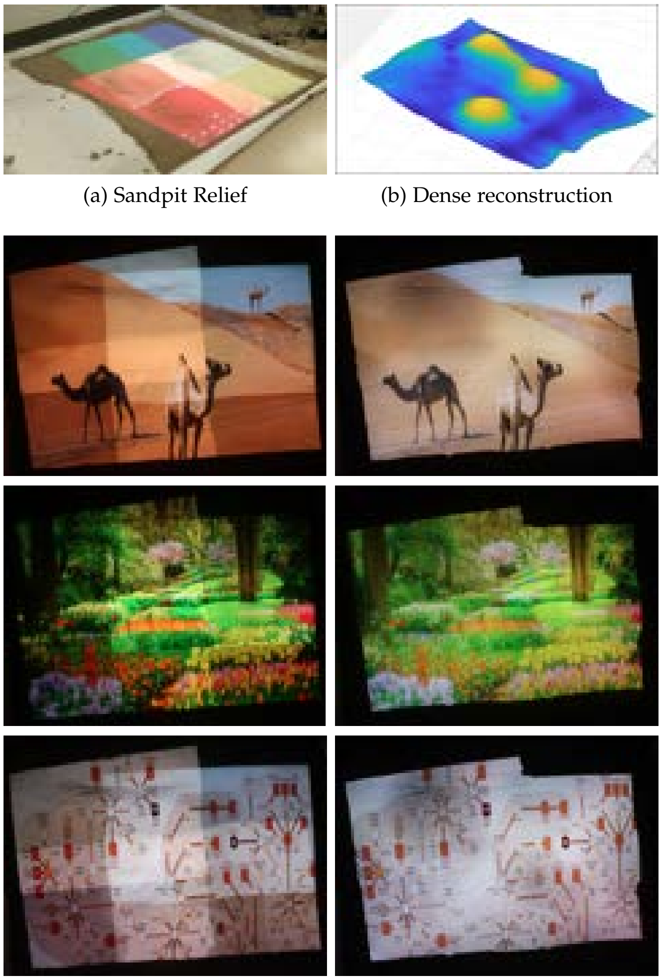
## 4 Implementation and Results

Our system comprises of four projectors and four cameras mounted on a steel rig (see Figure-1). We tested our system by projecting onto three kinds of 3D reliefs: two made of sand (Figures-7 and 9) that can be modified by hand to create different reliefs and one carved out of wood (Figure-8), which is a relief of a part of the Colorado river basin near Las Vegas. Each relief is around 4′x3′ is size. The two sand reliefs are brown in color, have different 3D geometries and projector overlaps: one has non-rectangular overlaps (Figure-7) whereas the other has rectangular overlaps (Figure-9). The wooden relief is white in color and also has non-rectangular projector overlaps (Figure-8).

We used Vivetek Qumi Q5 projectors along with Logitech C920 HD Pro cameras. The projector resolution is 720P, while the cameras captures at 1080P resolution. Each projector-camera pair is connected to a separate machine, though our algorithm does not require a distributed arrangement. The cameras are arranged such that each point on the projection surface is visible to at least two cameras. This is a requirement of geometric registration technique in [29], which we used to geometrically register projectors in our system. For capturing the color images, we mounted a Canon DSLR camera on the rig such that it sees the entire

TABLE 2
Evaluation of effect of the proposed method on the display quality. We measured the loss in dynamic range (DR) and the variation of the display brightness and chrominance by measuring the standard deviation (STD).

| Algorithm Steps | Mean DR Loss | Brightness STD | Chrominance STD |
|---|---|---|---|
| Before Correction | 0% | 0.719 | 0.374 |
| 3D Gamut Morphing | 16.82% | 0.456 | 0.271 |
| White Balancing | 25.65% | 0.428 | 0.255 |
| Brightness and Bezier-based Smoothing | 26.46% | 0.405 | 0.235 |


(a) Sandpit Relief

(b) Dense reconstruction

(c) Uncorrected

(d) Final Result

Fig. 9. The projection display before and after correction using our algorithm. **(a)**: A sand relief with four projectors having rectangular overlaps. **(b)**: The dense reconstruction of the surface. **(c)**-**(d)**: From top to bottom row: desert, garden and subway.

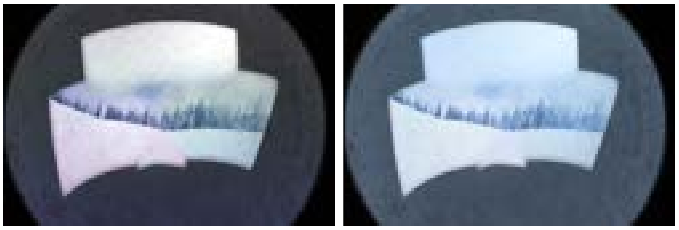


Fig. 10. Result of a three projector system on a half dome. *Left*: 3D gamut morphing. *Right*: Final result.

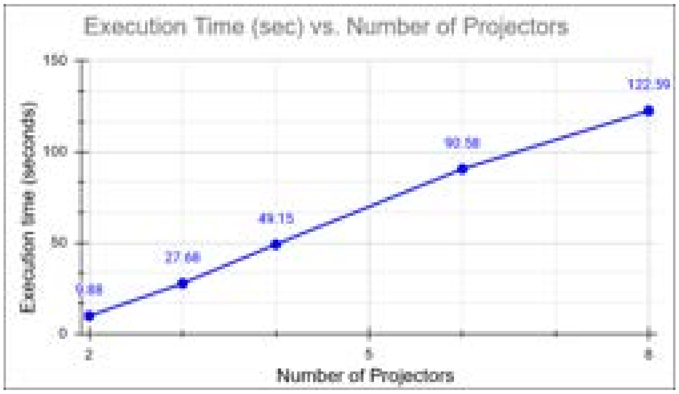


Fig. 11. Execution time (in seconds) of our algorithm against the number of projectors.

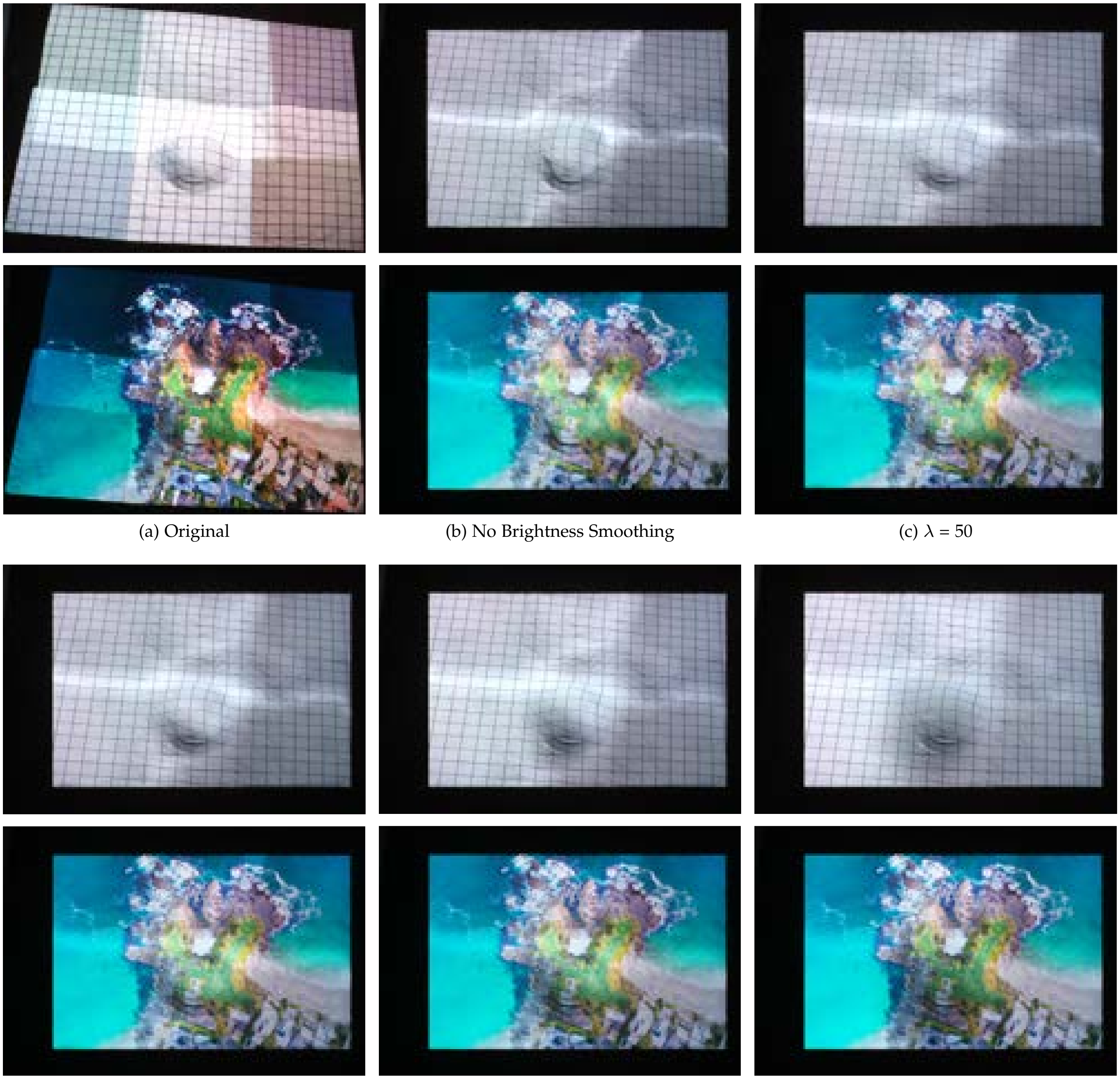
projection area. All results presented in this work have also been captured by the same camera.

Figures-7 and 8 show the surface geometries (**(a)**-**(b)**) and the projection display at each step of our algorithm (**(c)**-**(f)**). In Figure-7 **(c)**-**(f)** *(top-row)*, all projectors are projecting a white grid which clearly shows the differences between the color gamuts and white points of each projector. Similarly, Figure-8 **(c)**-**(f)** *(top-row)* shows all projectors projecting a white image on the wooden relief. Our algorithm is successfully able to morph the color gamuts and perform white balancing to produce a seamless display for both surfaces.

Figures-7 and 8 also show the same projectors projecting different imagery and the projection display at each step of our algorithm. Notice the images in Figure-7 **(c)**-**(f)** show clearly perceptible variations in color and brightness, especially in the low frequency regions of the projected content, for e.g. the background of the building plan and the blue ocean in the world map. Our algorithm is able to smoothly morph the color gamuts, perform white balancing and create a seamless display even though the projector overlaps are not rectangular. Similarly, our algorithm is able to create seamless displays on the wooden relief (Figure-8) with different types of content (temple, laguna and canyon).

Figure-9 shows the original and final results for different types of content on the sandpit with rectangular projection overlaps. Notice that while variations in color and brightness are visible in the low frequency regions of the projected content such as the sand in the desert image and the background in the subway image, they are not as perceptible in the garden image due to the high frequency content. Irrespective of the projected content, the proposed work is able to create color seamlessness across the entire display.

The proposed algorithm is scalable and works with any



(a) Original

(b) No Brightness Smoothing

(c) $\lambda$ = 50

(d) $\lambda$ = 100

(e) $\lambda$ = 200

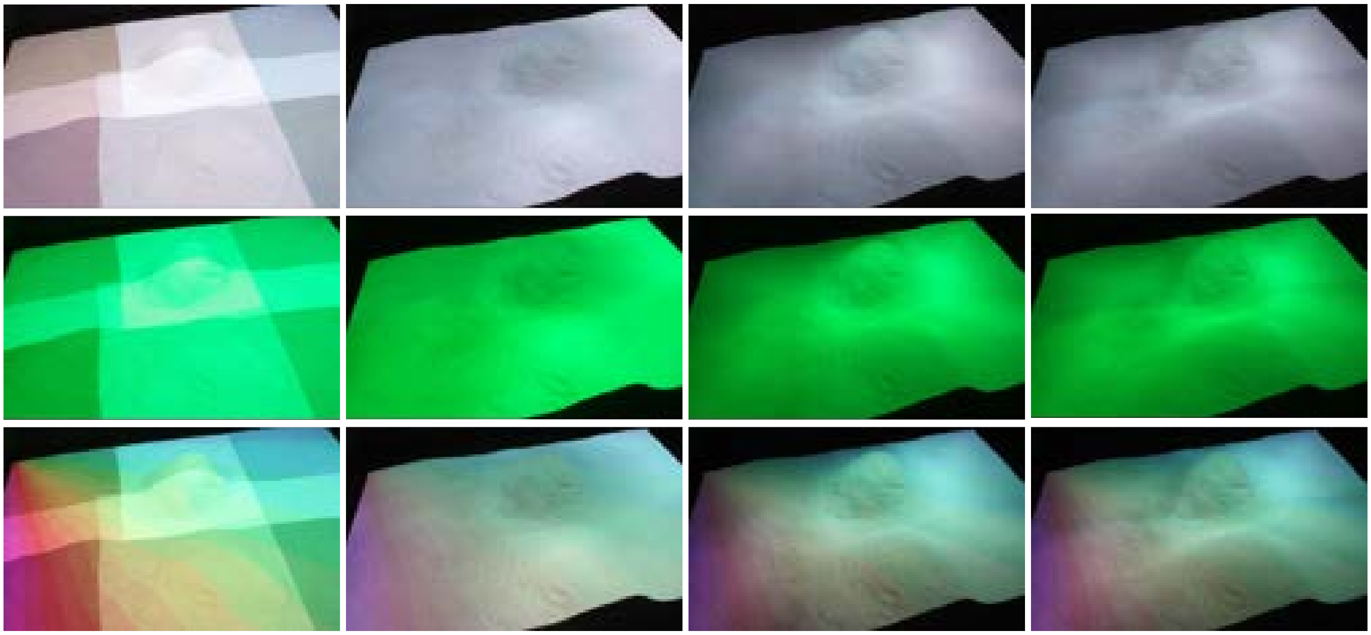
(f) $\lambda$ = 400

Fig. 12. Effect of the brightness smoothing parameter $\lambda$ on the final display when projecting a white grid and the Laguna image. Note that the value of $\lambda$ depends on the distance of the user from the display. A small image in a paper simulates an environment where the user is very far from the display and therefore $\lambda = 400$ creates the most seamless results. However, when the user is at the usual 4'-6' distance from the display, $\lambda$ of 50, 100 or 200 can provide a perceptively seamless display.

number of projectors and on any smooth shape. Figure-10 shows the 3D gamut morphing and final result of a three projector system on a half dome. Figure-11 shows how the execution time of the proposed algorithm scales linearly against the number of projectors. Note that the timing results were generated from unoptimized code implemented in MATLAB. Also, our experimental system consists of 4 projectors. The numbers for 6 and 8 projectors were generated using simulations.

### 4.1 Effect of Smoothing Parameter

Though [17], [25] has studied the effect of the brightness smoothing parameter in details before, we studied the effect of varying the brightness smoothing parameter $\lambda$ on the final display in a similar way when using the gamut morphing method (see Figure-12). This assures that our method does not contradict the prior explorations on the effects of $\lambda$. Most results are generated with $\lambda$ of 50, 100 or 200 (a distance of 4′ - 6′ from the display). However, when the same 4′x3′ sandpit display is presented as a small image on a paper, it simulates the case of a viewer much farther away



(a) Original (b) Gamut morphing with 3D surface distances (c) Gamut morphing with 2D parameter-space distances (d) Intensity Blending

Fig. 13. Comparison of the proposed gamut morphing using **(b)** 3D distances, **(c)** 2D distances and **(d)** standard intensity blending when projecting **(Top row)**: a flat white image, **(Middle row)**: a flat green image and **(Bottom row)**: a rainbow image. Notice how gamut morphing using 3D distances is more seamless compared to 2D-distance gamut morphing and intensity blending. The differences are particularly apparent on the side of the hill.

from the display and therefore some remaining seams are noticed. When viewed in person from the correct distance of 4′-6′, these seams will disappear. Figure-12, shows the effect of $\lambda$ on the results. Note that as $\lambda$ increases, the brightness variations across the display become less perceptible since the spatial variation become smoother. Also, note that an image with more high frequency content (the Laguna image in Figure-12) becomes seamless at a lower $\lambda$ than a flat colored image. This is because the remaining seams at lower $\lambda$ are hidden by the busier image content.

### 4.2 Effect of Geometry

In our work, we use the distance between 3D points on the surface to compute the weights for solving the Laplacian (see Equation-17). However, we additionally studied the effect of ignoring the 3D geometry of the surface when computing the weights. We compute a function $\rho(\cdot)$ that parameterizes the 3D surface i.e. $\bar{t} = \rho(T)$, where $T \in \mathbb{R}^3, \bar{t} \in \mathbb{R}^2$. We fit a plane to the 3D surface, project the 3D points on the plane and normalize them to compute the coordinates in parameter-space. Then, we compute the weights as in Equation-17, but using the distance between the points in parameter-space:

$$\bar{\Upsilon}(t,r) = \frac{||\rho(T) - \rho(R)||}{\sum_{s \in N(t)} ||\rho(T) - \rho(S)||} \tag{20}$$

Finally, we solve the Laplacian in Equation-16, but using $\bar{\Upsilon}(t,r)$ instead of $\Upsilon(\bar{t},r)$ and determined $\alpha_i$ for each projector as in Section-3.1.2. and compared the results of 3D gamut morphing with and without the surface geometry.

Figure-13 shows the comparison between using 3D surface distances versus 2D parameter-space distances to solve the weighted Laplacian. Notice how the brightness variations are more apparent when using 2D distances compared to 3D distances, especially on the sides of the hills. This is because 2D-parameter space cannot account for the hills on the surface. Therefore, the weights computed to solve the Laplacian are different due to the different distances in 2D and 3D respectively. Thus, we see a more seamless display when using 3D distances.

It is important to note that gamut morphing is not the same as intensity blending, which is why we also compare our results with standard intensity blending in Figure-13. For each 3D point that is illuminated by multiple projectors, we compute the 3D distance to the closest edge in each overlapping projector and scale the contribution of that projector pixel accordingly i.e. a pixel that is farther away from the edge as a higher contribution. Note that intensity blending results in very obvious brightness seams, especially on the hills, compared to both 3D-distance and 2D-distance gamut morphing. The fundamental assumption of intensity blending is that the projectors are very similar in color and the blend region is so wide that by blending the contributions from different projector in a complementary and smooth manner, we can create an imperceptible transition between the projectors. However, in almost any real system that uses projectors of different brands or lamp ages will show marked variation in color. In the rare case of projectors indeed providing very similar color on flat surfaces, the variation gets pronounced when projecting on complex 3D shapes. Finally, the blending region may not be adequately wide to create impreceptible transitions between differently colored projectors. Due to all these factors, blending in a complexly arranged multi-projector system will likely not yield conducive results, and has also been confirmed in earlier works [17], [25].

Although using 2D parameter-space distances still yields a reasonable display, we recommend using 3D distances to solve the Laplacian as it does not rely on the surface parameterization. Although surface parameterization techniques are quite advanced, incorrect parameterization could lead to incorrect weights for the Laplacian, which in turn can degrade the final quality of the display.

### 4.3 Effect on Display Quality

Performing color correction of a multi-projector system will impose constraints that reduce the overall brightness and dynamic range of the final display. Table-2 evaluates the effect of each step on the display quality. We measured the loss of dynamic range of the display, as well as the variation of brightness and chrominance across the display. For measuring the dynamic range, we calculated the average of the ratio of brightness of the display at each pixel when it projects white to when it projects black. To measure the variation of color across the display, we calculated the standard deviation of the Euclidean distance of the chrominance at each pixel from the mean chrominance of the display. The ideal color calibration method would have the least amount of loss in the dynamic range while reducing the variation in brightness and chrominance.

From Table-2, we can see that each step of the proposed method results in a loss of the dynamic range of the display. This is expected and can be visually confirmed in the results as well. However, note that the variation in both brightness and chrominance also reduces with each step, which results in a seamless display.

## 5 Conclusion

In conclusion, extending the gamut morphing approch from [25], we have presented a comprehensive solution to address spatial color variation in a geometrically aware manner for any arbitraily shaped 3D surface illuminated by a multi-projector system. Our method accounts for the presence of non-rectangular overlaps creating truly seamless multi-projector systems where the projector boundaries are invisible in the display. This is achieved by a Laplacian method based morphing of 3D gamuts that can be parallelized using a novel spherical space transformation resulting in an efficient solution. Using perceptual parameters, we make the spatial variation to be within human tolerance instead of matching color reproduction at every pixel. This provides us the leverage to maximize the color gamut and dynamic range of the display, yielding a display with much higher quality overall. This enables us to correct for flat colors on arbitrarily shaped geometry, a very challenging scenario for any prior method. Therefore, we deliver similar seamless quality as a prohibitively expensive high-resolution single projector using consumer grade multi-projector displays.

Our method is designed for smooth 3D shapes and does not take into account the surface normals. Our focus was to handle non-rectangular overlaps for which 3D distance based consideration of the 3D shape works well since drastic change of surface normals are not present in a smooth 3D surface. However, in the future we would like to address sharp variations in surface normals by using a metric that will consider surface normals in addition to 3D distances on the 3D shape.

## References

[1] D. G. Aliaga. Digital inspection: An interactive stage for viewing surface details. In *Proceedings of the ACM Symposium on I3D*, 2008.

[2] D. G. Aliaga and Y. Xu. Photogeometric structured light: A self-calibrating and multi-viewpoint framework for accurate 3d modeling. In *Proceedings of the IEEE CVPR*, 2008.

[3] D. G. Aliaga, Y. H. Yeung, A. J. Law, B. Sajadi, and A. Majumder. Fast high-resolution appearance editing using superimposed projection. In *ACM Transactions on Graphics*, 2012.

[4] M. Bern and D. Eppstein. Optimized color gamuts for tiled displays. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, p. 274–281, 2003.

[5] E. S. Bhasker, P. Sinha, and A. Majumder. Asynchronous distributed calibration for scalable and reconfigurable multi-projector displays. In *IEEE Transactions on Visualization and Computer Graphics (Visualization)*, 2006.

[6] E. J. Giorgianni and T. E. M. Madden. *Digital Color Management: Encoding Solutions*. Wiley, 2nd ed., 2009.

[7] B. E. Goldstein and J. Brockmole. *Sensation and Perception*. Cengage Learning, 10th ed., 2016.

[8] M. Hereld, I. Judson, , and R. Stevens. Introduction to building projection-based tiled display systems. In *IEEE Computer Graphics and Applications*, 2000.

[9] M. T. Ibrahim, M. Gopi, and A. Majumder. Dynamic projection mapping of deformable stretchable materials. In *26th ACM Symposium on Virtual Reality Software and Technology*, 2020. doi: 10.1145/3385956.3418970

[10] M. T. Ibrahim, A. Majumder, and M. Gopi. Dynamic projection mapping on deformable stretchable materials using boundary tracking. In *Computers and Graphics*, vol. 103, 2022.

[11] P. Kurth, V. Lange, M. Stamminger, and F. Bauer. Real-time adaptive color correction in dynamic projection mapping. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 174–184, 2020. doi: 10.1109/ISMAR50242.2020.00039

[12] A. Law, D. Aliaga, B. Sajadi, A. Majumder, and Z. Pizlo. Perceptually-based appearance modification for compliant appearance editing. In *Computer Graphics Forum*, 2011.

[13] A. Majumder. Properties of color variation across multiprojector displays. In *Proceedings of SID Eurodisplay*, 2002.

[14] A. Majumder and M. Gopi. *Introduction to Visual Computing: Core Concepts in Computer Vision, Graphics, and Image Processing*. CRC Press, 1st ed., 2018.

[15] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. In *Proceedings of IEEE Vis*, 2000.

[16] A. Majumder and R. Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. In *IEEE Transactions on Vis and Computer Graphics 10, 2*, March-April 2004.

[17] A. Majumder and R. Stevens. Perceptual photometric seamlessness in tiled projection-based displays. In *ACM TOG*, 2005.

[18] T. Nomoto, W. Li, H.-L. Peng, and Y. Watanabe. Dynamic multi-projection mapping based on parallel intensity control. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2125–2134, 2022. doi: 10.1109/TVCG.2022.3150488

[19] B. Pailthorpe, N. Bordes, W. P. Bleha, S. Reinsch, and J. Moreland. High-resolution display with uniform illumination. In *Proceedings Asia Display IDW*, pp. 1295–1298, 2001.

[20] P. Pjanic, S. Willi, D. Iwai, and A. Grundhöfer. Seamless multi-projection revisited. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2963–2973, 2018. doi: 10.1109/TVCG.2018.2868597

[21] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs. Pixelflex 2: A comprehensive automatic casually aligned multi-projector display. In *IEEE PROCAMS*, 2003.

[22] A. Raij and M. Polleyfeys. Auto-calibration of multi-projector display walls. In *Proceedings of ICPR*, 2004.

[23] R. Raskar, M. S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs. Multi projector displays using camera based registration. In *Proceedings of IEEE Vis*, 1999.

[24] 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 displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98)*, 1998.

[25] B. Sajadi, M. Lazarov, A. Majumder, and M. Gopi. Color seamlessness in multi-projector displays using constrained gamut morphing. In *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.

[26] B. Sajadi, M. A. Tehrani, M. Rehimzadeh, and A. Majumder. High-resolution lighting of 3d relief maps using a network of projectors and cameras. In *3D-TV conference on Immersive and Interactive 3D Media Experience Over Networks*, 2015.

[27] C. Siegl, M. Colaianni, L. Thies, J. Thies, M. Zollhofer, S. Izadi, M. Stamminger, and F. Bauer. Real-time pixel luminance optimization for dynamic multi-projection mapping. In *ACM Trans. Graph.*, October 2015.

[28] M. C. Stone. Color balancing experimental projection displays. In *9th IST/SID Color Imaging Conference*, 2001a.

[29] M. A. Tehrani, M. Gopi, and A. Majumder. Automated geometric registration for multi-projector displays on arbitrary 3d shapes using uncalibrated devices. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, pp. 2265–2279, 2021.

[30] G. Wallace, H. Chen, and K. Li. Color gamut matching for tiled display walls. In *Immersive Projection Technology Workshop*, 2003.

[31] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. S. Brown. Pixelflex: A reconfigurable multiprojector display system. In *Proceedings of IEEE Vis*, 2001.

**Mahdi Abbaspour Tehrani** Mahdi Abbaspour Tehrani received the BSc degree in computer engineering from the Sharif University of Technology and the PhD degree in computer graphics from the Department of Computer Science, University of California, Irvine, in 2018. His research interests include unified and automatic calibration of spatially augmented reality systems.

**Muhammad Twaha Ibrahim** received his Bachelor's degree from National University of Sciences and Technology, Pakistan and his Master's degree from Cornell University. He is currently working towards his PhD in CS at the University of California, Irvine. His research interests lie in the overlap of computer vision, graphics and visualization. His current research is in the domain of real-time multi-projector display systems that adapt to changing surface geometry and their application to projection-based augmented reality systems for creating interactive 3D experiences.

**Aditi Majumder** is a Professor at the Department of Computer Science in University of California, Irvine. She received her PhD in CS from University of North Carolina at Chapel Hill, USA in 2003 and Bachelors in CSE from Jadavpur University, Kolkata, India. Her research resides at the junction of computer graphics, vision, visualization, VR/AR and human-computer interaction. She is the co-author of two books - Practical Multi-Projector Display Design and Introduction to Visual Computing: Core Concepts in Computer Vision, Graphics and Image Processing. She has authored 70+ technical publications in premier journal and conferences, 5+ best paper awards, 10+ international keynotes. She has served as the program or general chair and program committee is several top venues including IEEE Virtual Reality (VR), ACM Virtual Reality Software and Technology (VRST), Eurographics and IEEE Workshop on Projector Systems. She has served as Associate Editor in Computer and Graphics and IEEE Computer Graphics and Applications. She has played a key role in developing the first curved screen multi-projector display that won the best display award in CES 2009 and was marketed by NEC/Alienware. She was an advisor at Disney Imagineering for advances in their projection based theme park rides. She is the recipient of the NSF CAREER award in 2009 for Ubiquitous Displays Via a Distributed Framework. She was a Givens Associate and a student fellow at Argonne National Labs from 2001-2003, Link Foundation Fellow from 2002-2003, and is currently a Senior Member of IEEE. She has recently launched her entrepreneurial venture via the startup Summit Technology Laboratory.

**M. Gopi** is a Professor of Computer Science in the Department of Computer Science, and Associate Dean at the Bren School of Information and Computer Sciences at University of California, Irvine. He received his BE from Thiagarajar College of Engineering, Madurai, MS from Indian Institute of Science, Bangalore, and PhD from University of North Carolina at Chapel Hill. His research interests include geometry and topology in computer graphics, massive geometry data management for interactive rendering, medical visual data processing, and visualization. His work on representation of manifolds using single triangle strip, hierarchyless simplification of triangulated manifolds, use of redundant representation for big data for interactive rendering, and biomedical image processing have received critical acclaim including best paper awards in two Eurographics conferences and in ICVGIP. He is a gold medalist for academic excellence at Thiagarajar College of Engineering, a recipient of the Excellence in Teaching Award at UCI, a Link Foundation Fellow, and Senior Member of IEEE. He served as the program co-chair and papers co-chair of ACM Interactive 3D Graphics conference in 2012 and 2013 respectively, area chair for ICVGIP in 2010 and 2012, program co-chair for International Symposium on Visual Computing 2006, an associate editor of the Journal of Graphical Models, a guest editor of IEEE Transactions on Visualization and Computer Graphics and in the steering committee of ACM Interactive 3D Graphics.