

# Practical Methods for Geometric and Photometric Correction of Tiled Projector Displays on Curved Surfaces

Michael Harville, Bruce Culbertson, Irwin Sobel, Dan Gelb, Andrew Fitzhugh, Donald Tanguay  
Hewlett-Packard Laboratories  
Palo Alto, CA United States

## Abstract

We describe a novel, practical method to create large-scale, immersive displays by tiling multiple projectors on curved screens. Calibration is performed automatically with imagery from a single uncalibrated camera, without requiring knowledge of the 3D screen shape. Composition of 2D-mesh-based coordinate mappings, from screen-to-camera and from camera-to-projectors, allows image distortions imposed by the screen curvature and camera and projector lenses to be geometrically corrected together in a single non-parametric framework. For screens that are developable surfaces, we show that the screen-to-camera mapping can be determined without some of the complication of prior methods, resulting in a display on which imagery is undistorted, as if physically attached like wallpaper. We also develop a method of photometric calibration that unifies the geometric blending, brightness scaling, and black level offset maps of prior approaches. The functional form of the geometric blending is novel in itself. The resulting method is more tolerant of geometric correction imprecision, so that visual artifacts are significantly reduced at projector edges and overlap regions. Our efficient GPU-based implementation enables a single PC to render multiple high-resolution video streams simultaneously at frame rate to arbitrary screen locations, leaving the CPU largely free to do video decompression and other processing.

## 1. Introduction

High-resolution, large-scale, wide field-of-view displays create immersive visual experiences that greatly enhance applications such as home theater, gaming, remote collaboration, scientific visualization, and human-computer interaction. While such displays may be built by abutting multiple smaller display units, the resulting inter-unit seams strongly detract from the experience. Multi-projector display walls are currently the most promising method for building large, seamless displays in a cost-effective manner. We are particularly interested in curving such displays around the viewers, to better fill their field-of-view and draw them into the experience.

Construction of curved, tiled projector displays is beset by several significant challenges. Perhaps the most apparent is the need to correct for the image distortions imposed

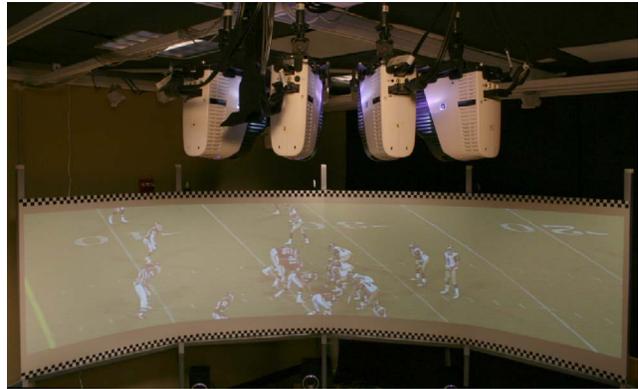


Figure 1. Projector array, curved screen, and example imagery, with lights on. A vertical physical seam exists at center of screen.

by projection onto a curved surface, and to align the image sub-sections projected by different projectors. Most prior work on multi-projector displays assumes approximately planar screens, for which automated geometric correction and alignment is simplified through use of homography coordinate transforms between the screen surface, the projector frame buffers, and the images of one or more calibration cameras observing the screen. Some researchers found that, by equating screen coordinates with those of a single camera, and by generalizing the projector-to-camera homographies to mesh-based mappings, imagery could be made to appear undistorted on arbitrary surfaces when viewed from the location of the camera [6, 9]. Through multi-camera construction of a 3D screen model, together with computer vision tracking of viewer location, such techniques were extended to adapt the preferred viewing direction to match the movements of a single viewer [7].

It would be highly preferable, however, to omit the complexity of person tracking and construct a curved display that looks undistorted to multiple viewers at different observation locations. Raskar et. al. have made recent breakthroughs in this area, by applying the techniques of conformal mapping and quadric transfer to minimize stretching of image pixels projected onto non-planar surfaces [8]. The resulting displayed imagery appears exactly like “wallpaper” on developable surfaces (discussed in Section 2) such as cylindrical sections, and approximately so for quadric surfaces such as domes, ellipsoids, and paraboloids.

A second important challenge in building multi-projector displays is photometric correction, which attempts to create smooth brightness levels and color gamut across the entire displayed image. After geometric alignment, the brightness of projected imagery will be higher in screen regions where projectors overlap. The complexity of these overlap regions can be high even for a simple 2x2 tiling arrangement, where 4 projectors may overlap in one place. Early multi-projector displays addressed this with “geometric blend maps”, derived entirely from geometric calibration data, that cross-fade the maximum brightnesses of projectors in overlap regions [1, 6, 7, 9]. Figure 5 shows example blend maps for a simple horizontal arrangement of three projectors; more complex arrangements require more complex maps.

Proper photometric correction, however, must also account for additional, important projector phenomena. First, projector light leakage causes a non-zero “black offset” to be projected even for black image input, and these offsets add up in projector overlap regions to produce visually disturbing artifacts. Further, the projector Tone Reproduction Functions (TRFs) relating output light color to input image color may be different for each projector, and tend to vary across pixels within a single projector. For example, maximum projector brightness often decreases toward the edge of the projector frustum. Even if geometric blend maps are applied, such TRF variations result in visible brightness and color discontinuities at projector seams. Recent work has compensated for such problems by measuring projector black level, maximum brightness, and TRF as a function of projector spatial coordinate, and then computing offset maps, attenuation masks, and TRF inversion functions that smooth the display appearance [4, 5].

Even if geometric and photometric correction are well solved, it is critically important for most practical display systems that installation and calibration may be done quickly and easily by relatively naive users. It is preferable that the system may be periodically re-calibrated in a rapid, fully automatic manner. Furthermore, the runtime implementation of the geometric and photometric correction should not be so complex as to prohibit display of imagery at video rates on commodity hardware.

Our work makes novel contributions to both geometric and photometric correction, while improving ease-of-calibration and run-time performance. For geometric correction, we build on the method of Surati [9], in which composition of two non-parametric 2D coordinate transforms, specifically 2D lookup tables from projector-to-camera and camera-to-screen, enables inversion of lens distortion and screen curvature effects in a single step. We obtain greater flexibility by employing coordinate transforms based on triangle meshes, and we improve significantly upon the techniques for estimation of these transforms. In addition, we extend the theory around these methods to show that the

wallpaper-like display produced by Raskar et. al. in [8] can be created for the same class of surfaces using a single uncalibrated camera, with reduced computational complexity and no estimation of 3D geometry. The advantages of our method are obtained at the cost of a new requirement to place a physical border around at least part of the screen area that will be used. We believe this is a practical, desirable tradeoff in many contexts, and that relative to prior work, our method provides a more easily implemented, fully automatic technique for geometric calibration of curved projection displays, without sacrifice in quality.

In the area of photometric correction, we believe the state-of-the-art may be advanced by combining the perceptually motivated smoothing of Majumder et. al. [4] with the geometric blend maps used, for example, in Raji et. al. [5]. We develop a mathematical framework to describe the method, and discuss its improved robustness to geometric correction error, which is often higher for curved displays than planar ones. To further ensure this reduced geometric sensitivity, we generate geometric blend maps in a new way that avoids internal discontinuities for any projector overlap configuration. Our methods are applicable to both planar and curved displays.

We have integrated these novel geometric and photometric methods into a real system that drives four, and soon eight, projectors with a single PC. By leveraging the PC graphics card to do most of the correction operations, the system is able to MPEG2-decompress and display multiple high-resolution videos simultaneously, at frame rate.

In Section 2, we describe our geometric correction techniques, including calibration measurement methods, building of mesh-based coordinate transforms, and construction of rendering meshes. Then, in Section 3, we develop mathematics for photometric correction, and describe the necessary computational and rendering steps. We discuss the architecture and performance of our current prototype system in Section 4, and conclude in Section 5.

## 2. Geometric Correction

We would like our curved display to have a physically consistent appearance across a large zone of viewing locations in 3D space. One way to achieve this is to imagine that the image to be displayed has been printed on a large, planar piece of paper that we conform, like wallpaper, to our display surface. Such a display would provide an intuitive, natural experience to one or more viewers, whether stationary or moving, without requiring adaptation of the projection parameters over time.

Physical wallpaper can be applied without distortion (stretching) only to the set of developable surfaces, which are the set of shapes that can be made by transforming a plane through folding, bending, cutting, and gluing. This is an important class of shapes for us, since curved screens are most readily made by manipulating flexible, planar sheets

of screen material, such as posterboard or (painted) sheet metal, in exactly these ways. The set of developable surfaces includes flat walls, piecewise-planar shapes (such as two walls meeting in a corner), cylindrical and conical sections, and sections of generalized cylinders (i.e. cylinders for which the cross-section may be any smooth curve).

We show that, to project imagery without distortion onto a developable surface, it is not necessary to know its 3D shape. This can be understood by considering an alternative, intuitive definition of a developable surface as a shape created by moving a straight line segment  $\lambda$  through 3D space. The right portion of Figure 2 provides illustration. If we let  $E_1(t_1)$  and  $E_2(t_2)$  be the “endpoint curves” traced by the movement of the endpoints of  $\lambda$ , and if we have the correspondence between the one-dimensional parameterizations  $t_1$  and  $t_2$  of these curves, then it is possible to construct a 2D coordinate system for a rectangular section of the plane  $\pi$  from which the curved screen surface was developed. Specifically,  $E_1(t)$  and  $E_2(t)$  are parallel in  $\pi$ , with the connecting line segment  $\lambda$  lying in the orthogonal direction at each  $t$ . We incorporate these ideas into a practical geometric correction method that results in a wallpaper-like, curved projection display.

### 2.1. Mesh-based coordinate transforms

Our geometric correction method employs three types of 2D coordinate systems:

- $\mathcal{P}_i$ : The frame buffer of the  $i$ th projector
- $\mathcal{C}_j$ : An image from the  $j$ th camera observing the screen
- $\mathcal{S}$ : The plane made by flattening the screen

A fourth coordinate system of interest is  $\mathcal{I}$ , namely that of the input image to be rendered by our multi-projector display. However, because the goal of our method is to wrap this image like wallpaper on the screen surface, it is helpful to consider  $\mathcal{I}$  as equivalent to  $\mathcal{S}$ , up to a scale factor in each of its two dimensions. If we normalize both coordinate systems to the range  $[0, 1]$  in each dimension, then  $\mathcal{I}$  is identical to  $\mathcal{S}$ . This has a critical implication: if we can determine a mapping between  $\mathcal{S}$  and each projector coordinate system  $\mathcal{P}_i$ , then we will also know the mappings from projector frame buffer locations to the input image pixel coordinates  $\mathcal{I}$  needed to produce the input image on the screen.

Let  $P_i(\vec{s})$  be a continuous-valued function that maps 2D screen coordinates  $\vec{s} = (s_x, s_y)$  in  $\mathcal{S}$  to coordinates  $\vec{p}_i = (p_{x,i}, p_{y,i})$  of the  $i$ th projector’s frame buffer. As shown in Figure 2, we construct each  $P_i$  as a composition of two coordinate mappings:

$$\vec{p}_i = P_i(\vec{s}) = C_{i,j}(S_j(\vec{s})) \quad (1)$$

where  $S_j(\vec{s})$  is a 2D mapping from the screen to the image pixels of the  $j$ th observing camera, and  $C_{i,j}(\vec{c}_j)$  is a 2D mapping from image pixel locations  $\vec{c}_j = (c_{x,j}, c_{y,j})$  in the  $j$ th observing camera to the  $i$ th projector’s frame buffer. We assume that all  $S_j$  and  $C_{i,j}$  are invertible mappings, so that

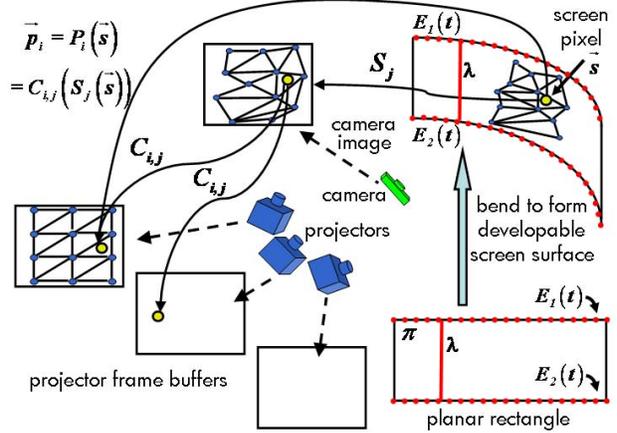


Figure 2. Developable surface concepts, and composition of mesh-based transforms from screen-to-camera and camera-to-projector. the mappings from projector frame buffers to the flattened screen are constructed similarly:

$$\vec{s} = P_i^{-1}(\vec{p}_i) = S_j^{-1}(C_{i,j}^{-1}(\vec{p}_i)) \quad (2)$$

Hence, all coordinate transforms required by our geometric correction can be derived from the  $S_j$  and  $C_{i,j}$  mappings.

To handle a broad set of screen shapes, we construct generalized, non-parametric forms of these coordinate mappings. Specifically, for each mapping, we use a mesh-based coordinate transform derived from point correspondences between the coordinate systems of interest. This technique has been used in prior multi-projector displays for representing the camera-to-projector mappings [1, 6, 9], but not for relating a camera image to locations on a curved surface.

Given a set of point correspondences between two 2D spaces  $\mathcal{A}$  and  $\mathcal{B}$ , we map a point location  $\vec{a}$  in  $\mathcal{A}$  to a coordinate  $\vec{b}$  in  $\mathcal{B}$  as follows. First, Delaunay triangulation is applied to the points in  $\mathcal{A}$ , followed by construction of the corresponding triangle mesh (according to the point correspondences) in  $\mathcal{B}$ . Next, we find the triangle in  $\mathcal{A}$  that contains  $\vec{a}$ , or whose centroid is closest to it, and compute the barycentric coordinates of  $\vec{a}$  with respect to that triangle. Finally, we select the corresponding triangle in  $\mathcal{B}$ , and compute  $\vec{b}$  as the point having these same barycentric coordinates with respect to this triangle.

This process is essentially the same affine texture mapping implemented by typical computer graphics rendering hardware. We perform our operations in software because our coordinate transforms are not being used directly to do texture mapping. Since Delaunay triangulation works for points in arbitrary arrangement, we have great flexibility in our methods for finding point correspondences. Construction of the triangle meshes needs to be done only once after a new set of point correspondences is measured from calibration methods such as those described in Section 2.2.

To determine the correct projector frame buffer contents needed to render the input image like wallpaper on the screen, we can first apply equation (2) to determine the screen location  $\vec{s}$  that each projector pixel  $\vec{p}_i$  lights up. If

we normalize  $\vec{s}$  to  $[0, 1]$  in both dimensions, then this is also the coordinate for the input image pixel whose color should be placed in  $\vec{p}_i$ , since wallpapering the screen effectively equates 2D flattened screen coordinates with image coordinates. Given the large number of projector pixels in our display, it is computationally intensive to set the color of each individually in this manner. Instead, for each projector, we use equation (2) to compute the image coordinates corresponding to each location on a sparsely sampled rectangular grid (e.g. 20x20) in the projector coordinate space. We then rely on graphics hardware to fill the projector frame buffer via texture mapping image interpolation. Hence, the final output of our geometric correction is one triangle mesh, computed on the rectangular grid, per projector.

It is important to note that because the method described above includes a dense mapping to the physical screen coordinate system, it corrects for image distortion caused not only by screen curvature, but also due to the projector lenses. Furthermore, the lens distortion of the observing cameras does not need to be calibrated and corrected. In fact, the method allows use of cameras with extremely wide angle lenses, without any need for camera image undistortion. Because of this, we have focused on calibrating our multi-projector display with a single, wide-angle camera. This approach can even be used to calibrate full 360° displays, by placing a conical mirror in front of the camera lens to obtain a panoramic field-of-view.

## 2.2. Point correspondence measurements

As discussed in Section 2.1, our mesh-based continuous coordinate transforms are all constructed from measured point correspondence data. Because the methods of Section 2.1 apply to points in any general 2D arrangement, many of the correspondence measurement methods described in the literature can be used. In this section, we discuss our current techniques, but these are not required by our geometric correction, and we continue to explore other methods.

We obtain camera-projector point correspondences via a structured light approach akin to that of Chang [2]. This method projects a sequence of bar images of increasingly fine spatial frequency to temporally encode the projector coordinates corresponding to various camera pixels. An advantage of this method is that it directly solves for camera-projector point correspondences without image search or computation of 3D data. We typically use a sequence of 8 to 12 images per projector, with the spatial frequency limit depending on the camera image resolution.

We next discuss the screen-to-camera point correspondences. As described at the start of Section 2, a developable surface can be made by folding, bending, and otherwise manipulating a planar sheet of material, and also corresponds to a surface swept out by moving a straight line segment through 3D space. A 2D coordinate system in a rectangular portion of the original planar sheet can be recovered from

the endpoint curves  $E_1(t_1)$  and  $E_2(t_2)$  traced by the moving line segment. To identify the correspondence between these curves, we apply a “fiducial border” around the portion of the screen onto which we wish to project. Our fiducial border contains features with regular physical spacing, thus defining a known sampling in the 2D manifold of the screen surface. For geometric correction with zero distortion on a non-planar developable surface, the border should be applied to the surface so that two opposite segments lie along straight lines in 3D space. The other two connecting portions of the border will then correspond to the endpoint curves  $E_1(t_1)$  and  $E_2(t_2)$ . This is sufficient to guarantee that, if the surface were flattened into the plane from which it was developed, the fiducial border would form a rectangle in that plane. This rectangle corresponds to the coordinate system of the image we wish to project.

Figure 1 shows a cylindrical screen section with checkerboard fiducial border strips applied only to the endpoint curves. These strips consist of adhesive tape onto which a regular pattern was printed. It should be noted that Surati’s method [9] requires temporary attachment of a 2D fiducial pattern across the entire curved display surface. We use the OpenCV corner detector within a predictive matching framework to find all of the border corner features from a single image of the screen. Because the physical spacing of the corners is known, we can convert the indices of the corners along any given strip into coordinates along the edge of the 2D screen manifold to be used by the tiled display. These screen coordinates are associated with the camera image locations at which the corner features were detected, to create the screen-to-camera correspondences.

While use of a fiducial border might at first seem undesirable, several points should be considered. Because a fiducial border allows geometric correction of large curved screens to be done with a single camera without 3D shape estimation, the complexities of multi-camera calibration and/or structured light measurement are avoided. Further, the border not only aids in geometric correction, but also defines the “window” in which the multi-projector display should create an image. To obtain a window with the desired aspect ratio and orientation, all prior systems have either also employed screen fiducials, have required user guidance, or rely on special sensors built into the screen or projectors. Because our border does not interfere with the projected imagery, it may be left up at all times to allow re-calibration, but it also may be removed after calibration has completed. Although our current fiducial tape employs checker features, more aesthetically pleasing features may be substituted to create an attractive screen border, provided that robust detectors are created to match.

## 3. Photometric correction

Our photometric correction method builds upon prior work, in particular that of Raj et. al. [5] and Majumder

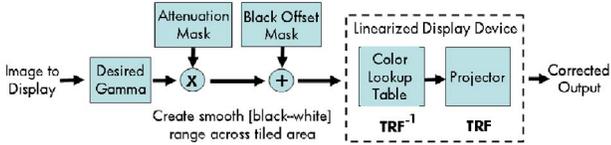


Figure 3. LSO photometric correction strategy

and Stevens [4]. We combine the best features of these approaches into a new formulation of photometric correction, and also introduce a new means of computing geometric blending maps that avoids internal discontinuities. A primary benefit of our method is reduced sensitivity to geometric correction error. We describe our overall photometric correction approach in Section 3.1, and we focus on the geometric blend maps in particular in Section 3.2.

### 3.1. Photometric correction theory

The goal of our photometric correction is to compensate for all of the effects discussed in Section 1 in order to create a tiled display that, given an input image of any single solid color, will not produce visually disturbing color variation on the screen. If this condition holds, any displayed image should appear reasonably faithful to its source content.

The basic strategy of our approach, which we refer to as “Linearize, Scale, and Offset” (LSO), is shown in Figure 3. It is motivated as follows. A black offset image can be added to each projector’s output in order to create a smooth black level across the display. Multiplicative attenuation maps can be applied to brightnesses above this smooth black level in order to smooth their spatial variation. With nonlinear projector TRFs, different attenuation factors are needed to smooth the screen appearance of different input colors. We can instead allow the same attenuation maps to be used for all inputs if we linearize the projector TRFs. This is effectively achieved by applying the inverse TRF to the imagery just prior to sending it to the projector, so that the combination of this inversion and the physical projector may be treated together as a linear projector. The only remaining problem is that the displayed imagery will appear saturated, because the standard nonlinear “gamma” exponential function typically applied to images has been replaced with a linear pass-through. To correct this, we can apply an exponential enhancement to the input image prior to other corrections. At a high-level, this LSO strategy is identical to that of [4], while only the initial “desired gamma” operation is omitted in [5].

Part of our photometric correction is based on geometric calibration data alone. Specifically, we compute geometric blending maps  $G(p_i)$  for each projector, so that brightness for this projector is attenuated in regions for which it overlaps other projectors. These maps spatially “cross-fade” the projectors’ brightness levels through overlap zones between two or more projectors, so that the maximum brightness contributed by any one projector approaches zero as one approaches the edge of that projector’s coverage area. Our

method for computing these maps is described in Section 3.2, but for now it is sufficient to understand their purpose and typical appearance, as in Figure 5.

We use only a computer-controlled Canon digital camera to collect the measurements needed for photometric correction. Specifically, for each projector in turn, we project a series of gray levels from black to white, and capture one camera image of each. All images are captured in three-channel color. While gray levels for one projector are being captured, all overlapping projectors are turned on but are projecting black. Measurements for non-overlapping projectors may be obtained simultaneously, if desired. Most digital camera brightness response curves have a nonlinear shape, but we can put the Canon camera into a special “linear” output mode, such that the resulting image values are proportional to the light intensity at the imaging chip. For cameras that do not have such a mode, the brightness response curve may be measured by any of several standard methods [3], and inverted in software to produce linear image data. Also, because typical cameras, including ours, do not have sufficient dynamic range to capture both the darkest and brightest projector outputs, we actually capture two sets of imagery at different camera settings: one that is properly exposed for the darkest levels, and a second that captures the brightest levels without saturation.

We use the image data for three computations:

**1. Inverse TRF estimation:** For each projector  $i$  and color channel  $l$ , the linearized camera measurements are plotted against the projector input gray levels at each of several projector coordinate locations. Each such curve is normalized to have a  $[0, 1]$  domain and range, then inverted, and finally fit with an exponential function. The resulting exponential “gamma” factors are averaged across all curves to obtain a single parameterized form of the inverse TRF for that projector and color channel, which we will denote as  $h^{-1}(I_{i,l})$ . This averaging is justified because we have found experimentally, as in [4], that the TRF is similar for different spatial locations within a given projector.

**2. Measured and smooth black level:** We use the geometric mapping functions and our images of the darkest projector output to construct a black level “measurement map” in the flattened screen space  $S$  of the full-tiled display. This is what should be measured by the camera when all projectors output black. We denote this map as  $B(\vec{s})$ . We apply to this a constraint-based smoothing algorithm analogous to that of Majumder and Stevens [4] to create a black offset “target map”  $B_t(\vec{s})$  that is spatially smooth across the entire tiled display. The target map is what should be measured by the camera after black level correction is applied.

Camera images captured using the longer exposure setting are used to compute these black maps. Note that the measurement map will also incorporate any screen texture and ambient lighting, and hence the target black map will

be a smoothed version of this. We therefore turn off all room lights at capture time.

**3. Measured and smooth white level:** The measured map of maximum screen brightness in screen coordinate space  $S$ , which we denote as  $W(\vec{s})$ , is computed at each screen location  $\vec{s}$  as a weighted sum of the white-level contributions from all projectors illuminating  $\vec{s}$ , where the weights are the projectors’ geometric blend map values  $G(P_i(\vec{s}))$  at this screen location. We include the measured black level  $B(\vec{s})$  only once during the summation, and apply the blend weight only to the portion of the white measurement in excess of  $B(\vec{s})$ , since the black level cannot be attenuated. Constraint-based smoothing, as in [4], is then applied to  $W(\vec{s})$  to create a white “target map” denoted  $W_t(\vec{s})$ .

We use only the camera images with shorter exposure to compute the white maps, under the assumption that the black level is negligible in comparison. If this assumption does not hold across the entire screen, then high-dynamic range imaging should be used to relate the brightness scales of the two sets of camera images, so that the black level can be accurately subtracted from the white.

We now derive appropriate attenuation and black offset maps to use in the rendering scheme of Figure 3. Let  $I(\vec{s})$  be the three-channel input image color at screen location  $\vec{s}$  and, by equation (1), also at corresponding projector coordinates  $P_i(\vec{s})$ . If we assume that the  $i$ th projector’s TRF has been linearized by application of  $h^{-1}(I_{i,l})$ , then the projector output color  $L(\vec{p}_i)$  at pixel location  $\vec{p}_i$  should be:

$$L(\vec{p}_i) = [G(\vec{p}_i)(W(\vec{p}_i) - B(\vec{p}_i))] * I(P_i(\vec{s})) + B(\vec{p}_i) \quad (3)$$

This is the equation of a line that, over the domain of  $I = [0, 1]$ , has a minimum value at  $I = 0$  equal to the measured black offset  $B(\vec{p}_i)$ , and a maximum value at  $I = 1$  equal to the measured white offset after attenuation by geometric blend factor  $G(\vec{p}_i)$ .

To compensate for the linearity of the projector response, we enhance the input image color  $I$  with an exponential function  $H$  (shown at left side of Figure 3), with exponent typically around 2.3. Because of the projector linearity,  $H$  becomes the effective “gamma” of our entire tiled-projector display, and is controllable in software. This enhancement is applied prior to other corrections, and may be expressed through a simple substitution in the above equation:

$$L(\vec{p}_i) = [G(\vec{p}_i)(W(\vec{p}_i) - B(\vec{p}_i))] * H(I) + B(\vec{p}_i) \quad (4)$$

For  $M$  projectors overlapping at screen location  $\vec{s}$ , the expected output color on the screen is obtained by summing equation (4) across all projectors:

$$L(\vec{s}) = H(I) * \sum_{i=1}^M [G(\vec{p}_i)(W(\vec{p}_i) - B(\vec{s}))] + B(\vec{s}) \quad (5)$$

For  $I = 0$  and  $I = 1$ ,  $L(\vec{s})$  equates to black and white measurement map values  $B(\vec{s})$  and  $W(\vec{s})$ , respectively.

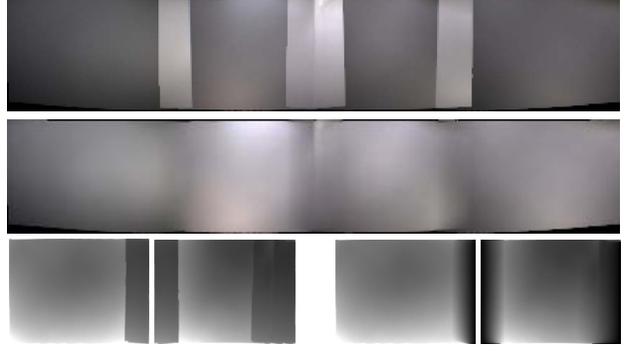


Figure 4. Top: Measured white map using method of [4]. Middle: Measured white map of proposed method, with geometric blending. Bottom left: Attenuation maps for two left-most projectors from method of [4], with internal discontinuities. Bottom right: Maps for same projectors using proposed method are smooth.

The desired projector response at  $\vec{s}$ , defined by the target black and white maps computed as described above, is also a line, but with different slope and intercept:

$$L(\vec{s}) = H(I) * [W_t(\vec{s}) - B_t(\vec{s})] + B_t(\vec{s}) \quad (6)$$

We begin to bring equations (5) and (6) into agreement by inserting into equation (5) a scale factor  $\alpha(\vec{p}_i)$  and offset factor  $\beta(\vec{p}_i)$  that are the same for all of the projectors:

$$L(\vec{s}) = H(I) * \sum_{i=1}^M [\alpha(\vec{p}_i) G(\vec{p}_i)(W(\vec{p}_i) - B(\vec{s}))] + (\beta(\vec{p}_i) + B(\vec{s})) \quad (7)$$

It can be shown that the following formulas cause equations (6) and (7) to be equal:

$$\begin{aligned} \alpha(\vec{p}_i) &= \frac{W_t(\vec{s}) - B_t(\vec{s})}{\sum_{i=1}^M G(\vec{p}_i)(W(\vec{p}_i) - B(\vec{s}))} \\ \beta(\vec{p}_i) &= \frac{B_t(\vec{s}) - B(\vec{s})}{M} \end{aligned} \quad (8)$$

Intuitively, the value of  $\alpha(\vec{p}_i)$  at a given screen location is the ratio of the target display dynamic range here (from smoothed white map  $W_t$  down to smoothed black map  $B_t$ ) to the original measured dynamic range of the tiled display *after geometric blending has been applied*.  $\beta(\vec{p}_i)$  distributes the difference between target smooth black map  $B_t$  and measured map  $B$  equally among all projectors overlapping at  $\vec{p}_i$ . The black offset maps used by the renderer of Figure 3 are the  $\beta(\vec{p}_i)$ , while the final attenuation maps are  $\alpha(\vec{p}_i) * G(\vec{p}_i)$ . Because  $B$ ,  $B_t$ ,  $W$ , and  $W_t$  are all in three-channel color, the above method can produce separate results for each color channel.

Application of geometric blending prior to creation of the target white measurement map  $W_t(\vec{s})$  is a critical difference between this method and that of [4], resulting in a photometric calibration that is more tolerant of geometric calibration error. As shown in Figure 4, a white measurement map created without geometric blending contains

sharp brightness discontinuities at projector overlap region boundaries. In contrast, the proposed method blends projector contributions in overlap regions to produce a relatively smooth  $W(\vec{s})$  whose differences from uniformity reflect only the intrinsic brightness variations of the projectors, rather than spatial overlap geometry. Because the attenuation map values of method [4] are proportional to the differences between  $W(\vec{s})$  and the smoothed target white map  $W_t(\vec{s})$ , the discontinuities evident in  $W(\vec{s})$  propagate to the attenuation maps. Examples of such maps are shown in Figure 4. Unless these discontinuities are geometrically aligned across projectors with sub-pixel precision (which is especially difficult to achieve on curved surfaces), salient high-frequency artifacts are produced. Elimination of white measurement map discontinuities through geometric blending, on the other hand, yields smoother attenuation maps, shown for comparison in Figure 4, and greater tolerance of geometric calibration imprecision.

### 3.2. Smooth geometric blend maps

Geometric blend maps, created from the geometry of projector overlap regions alone, have been used in many projector tiling methods. Geometric blending applies multiplicative attenuation factors to projector outputs in overlap regions, with these factors summing across all projectors to 1 at each spatial location. Consider the case of two partially overlapping projectors. A simple approach is to divide the attenuation equally across both projectors within the overlap region. However, this creates a step edge discontinuity, from a value of 1 to 0.5, inside the attenuation map for each projector at the location of the overlap region boundary. Unless the discontinuities in the two maps are precisely aligned with each other on the screen by the system’s geometric correction, a disturbing visual artifact is produced.

For this reason, most multi-projector displays use geometric blend maps with smooth “roll-off” functions from one boundary of the overlap region to another. This creates a spatial “cross-fade” between overlapping projectors. The roll-off function may be a simple linear ramp, or more complex shapes such as cosines or sigmoids. However, simple functions such as these are defined for only one dimension, so that it is not always clear what to do for overlap regions that are not aligned along a coordinate axis of both projectors. Furthermore, such functions do not offer a solution where three or more projectors overlap.

To better solve this problem for arbitrary projector overlap arrangements, Raskar et. al. [7] introduced a scheme that computes the distances  $d_j$  of a screen point to the nearest edge of each  $j$ th projector that overlaps it. The blend map value for the  $i$ th projector at this location is then given by  $G(\vec{p}_i) = \frac{d_i}{\sum_j d_j}$ . However, as shown in the middle of Figure 5, this continues to produce edge artifacts inside the blend maps. These artifacts occur along the diagonals

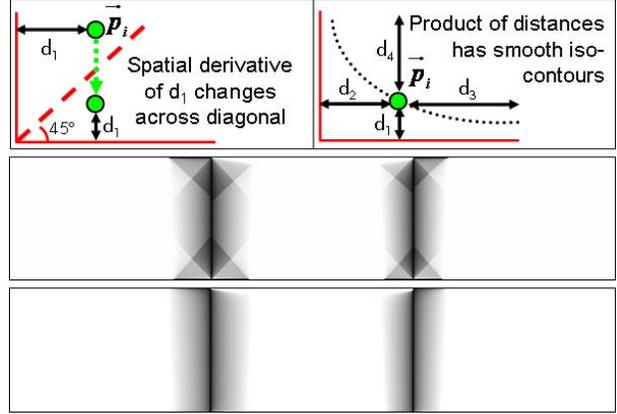


Figure 5. Top left: Spatial derivative discontinuity for method [7]. Top right: Proposed method. Middle: Geometric blend maps laid side-by-side for 3-projector horizontal arrangement, computed by method [7]. Bottom: Blend maps using proposed method.

of each projector’s coordinate system, where the choice of nearest projector edge changes, as indicated in the top left of Figure 5. Discontinuities in the first-order spatial derivative of  $d_i$  result here, and propagate to form similar discontinuities in  $G(\vec{p}_i)$ . The discontinuity inside one projector’s blend map is complemented by similar ones within all blend maps of overlapping projectors, and these must be very precisely aligned by geometric correction to avoid creating edge artifacts, to which the human visual system is highly attuned, in the final tiled display.

We therefore seek a new method for creating blend maps that avoids such internal discontinuities. We begin by creating a scalar-valued function  $\xi(\vec{p})$  over projector coordinates, having the following properties:

1.  $\xi_i(\vec{p}_i)$  goes to zero as  $\vec{p}_i$  approaches any projector edge
2.  $\xi_i(\vec{p}_i)$  and its spatial derivative are not discontinuous anywhere inside the coordinate bounds of the projector

We can then construct  $G(\vec{p}_i)$  as in [7], replacing  $d_i$  with  $\xi_i$ .

Our solution is illustrated at the top-right of Figure 5. Let us denote the distances of  $p_i$  to the four projector edges as  $d_{i,k}$ , where  $k \in [1 \dots 4]$ . With the distances sorted in ascending order, we consider the product of the two smallest distances, namely  $\xi_i = d_{i,1} * d_{i,2}$ . The isocontours of  $\xi_i$  form smooth hyperbolae within the projector coordinate system, approaching zero at the projector boundaries. This function is suitable for generating smooth geometric blend maps. However, the product of all four distances produces similar isocontours, and is faster to compute because no conditionals or distance sorting is needed. Hence, we compute geometric blend maps according to:

$$G(\vec{p}_i) = \frac{\xi_i}{\sum_j \xi_j}; \quad \xi_i = \prod_{k=1}^4 d_{i,k} \quad (9)$$

Figure 5 shows blend maps computed in this manner. The method is applicable to planar as well as curved screens, and reduces the effects of geometric alignment errors.



Figure 6. Image on 4-projector display, before and after correction.

#### 4. System construction and performance

The largest display we have built so far uses four HP mp3130 projectors, each having 1024x768 resolution, arranged in a ceiling-mounted 4x1 array. This is shown in Figure 1. They are driven by a single HP xw9300 workstation, which includes two AMD Opteron 2.19 GHz CPUs and two Nvidia Quadro FX3400 GPUs, each of which can drive two projectors. We will soon move to a newer PC that supports 4 graphics cards, and hence 8 projectors. Additional projectors can be added by running on multiple machines with synchronized graphics cards.

Our current screen is made from two 8x4 foot sheets of counter-top laminate material (available from Home Depot), bent into a 120° cylindrical arc. This arrangement comfortably seats 6 people across for immersive viewing, and many more may watch from behind them. The screen will soon be extended to 180°, to entirely fill the observers' field-of-view. The exact resolution of the resulting display varies with the amount of overlap of the projector images; we typically obtain about 3300x700 pixels after calibration.

Our calibration camera is a computer-controllable, 8MP Canon EOS Digital Rebel. With a standard 18-55mm lens, the camera can view and calibrate the 120° screen. We are currently experimenting with a panoramic mirror camera attachment for calibration of screens with greater angular extent. The screen material provides relatively Lambertian light dispersion, although not perfectly so. Hence, our photometric calibration performs best for screen regions where the camera viewing angle is nearer the surface normal. This leads us to prefer a wide angle or panoramic calibration camera placed near the center of our cylindrical arc.

Our geometric and photometric calibration procedure is fully automated, after application of the “fiducial tape” to the borders of the screen. We have not yet attempted to optimize our image capture and correction computations, but expect to obtain a complete calibration cycle time of a few minutes, or less, when we soon devote ourselves to this.

The majority of the runtime computation is performed by the graphics card's GPU rather than by the CPU. Geometric correction is applied using texture mapping of the input image onto one triangle mesh per projector. Photometric

correction is applied via fragment shader programs. Figure 6 shows an image displayed by four projectors on our screen, first before geometric and photometric correction is performed, and then after correction.

By performing most runtime computation on the GPU, we leave the computer CPU free to receive and decompress multiple video and audio streams. We have also designed our renderer to display multiple images or videos at once, at different screen locations and sizes. Dynamic repositioning and scaling of the videos, allowing one or more to be brought to emphasis at a keystroke, is also implemented. The system can show six movies simultaneously at their native 24fps, or six standard definition video streams at 30fps.

#### 5. Conclusion and Future Work

The method described herein is among the few to attempt high-quality, fully automatic correction of both geometric and photometric aspects of multi-projector displays on curved surfaces. A key advantage is its use of a single calibration camera, enabled by reliance on a fiducial border defining the window of projection on the screen. As a result, the complexities of multi-camera registration and color mismatch are completely avoided, yielding robust results and easier implementation. We also have improved on prior photometric correction methods, applicable to planar as well as curved screens, to reduce their sensitivity to geometric correction error. We have integrated these concepts into a working prototype that allows display and interaction with multiple video streams at frame rate. We plan to improve both the quality and speed of the correction methods, and expand our system onto larger curved surfaces.

#### References

- [1] M. Brown, W. B. Seales. “A practical and flexible tiled display system.” In *Pacific Conf. on Comp. Graphics and Apps.*, 2002.
- [2] N. Chang. “Efficient dense correspondences using temporally-encoded light patterns.” In *PROCAMS'03*.
- [3] *Colour Engineering* Ed. by P. Green & L. MacDonald. 2002.
- [4] A. Majumder, R. Stevens. “Perceptual photometric seamlessness in projection-based tiled displays.” In *ACM Transactions on Graphics*, Vol. 24, No. 1, pp. 118-139, 2005.
- [5] A. Raji, G. Gill, A. Majumder, H. Towles, H. Fuchs. “PixelFlex2: a comprehensive, automatic, causally-aligned multi-projector display”. In *PROCAMS'03*.
- [6] R. Raskar, G. Welch, H. Fuchs. “Seamless projection overlaps using image warping and intensity blending.” In *Intl. Conf. on Virtual Systems and Multimedia*, Gifu Japan, 1998.
- [7] R. Raskar, M. Brown, R. Yang, et. al. “Multiprojector displays using camera-based registration.” In *IEEE Visualization*, pp. 161-168, 1999.
- [8] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, C. Forlines. “iLamps: Geometrically Aware and Self-Configuring Projectors”. In *ACM Transactions on Graphics*, Vol. 22, No. 3, pp. 809-818, 2003.
- [9] R. Surati. *Scalable self-calibrating display technology for seamless large-scale displays*. Ph.D. thesis, Dept. of Computer Science, MIT, 1998.