A Projection System with Radiometric Compensation for Screen Imperfections*

Shree K. Nayar, Harish Peri, Michael D. Grossberg and Peter N. Belhumeur

Department of Computer Science, Columbia University
New York, New York 10027
E-mail: {nayar, mdog, hsp7, belhumeur}@cs.columbia.edu

Abstract

A major limitation of existing projection display systems is that they rely on a high quality screen for projecting images. We believe that relaxing this restriction will make projectors more useful and widely applicable. The fundamental problem with using an arbitrary surface for a screen is that the surface is bound to have its own colors and textures (bricks of a wall, painting on a wall, tiles of a ceiling, grain of a wooden door, etc.) or surface markings (paint imperfections, scratches, nails, etc.). As a result, when an image is projected onto the surface, the appearance of the image is modulated by the spatially varying reflectance properties of the surface. Humans are very sensitive to such modulations.

In this paper, we present a method that enables a projector to display images onto an arbitrary surface such that the quality of the images is preserved and the effects of the surface imperfections are minimized. Our method is based on an efficient off-line radiometric calibration that uses a camera to obtain measurements from the surface corresponding to a set of projected images. The calibration results are then used on-line to compensate each display image prior to projection. Several experimental results are shown that demonstrate the advantages of using our compensation method.

1 Introduction

In the last decade, projection display technology has undergone a revolution. Projectors are able to display images with very high spatial resolution and dynamic range. At the same time, they have become highly efficient in terms of their power consumption as well as their physical packaging. In addition, their designs have gone through significant innovations that have reduced their prices to levels that make them a consumer product. The end result of all of these advances is that the projector has become ubiquitous; it is now an integral part of our everyday lives.

In recent years, several display systems have been developed that use projectors as the basic building blocks. For instance, a set of projectors have been used to develop a large seamless high resolution display [9, 5, 3, 8], to create immersive environments [1, 17], to produce high quality images that have several component images [12], and to eliminate shadows cast on the screen [21]. Projectors have also been used to change the appearance of a real object [18]. For instance, a Lambertian white object can be made to appear like one with texture. If the observer's position is known, the appearance of the object can even include specular and transparency effects [4].

The above applications rely on some prior information about the projectors and the surface they project onto. In most cases, the geometric mapping between the projector(s) and the screen must be known [2, 6, 10]. When displaying multiple overlapping images, the photometric properties of the projectors must be calibrated and accounted for [11, 19, 20]. A convenient way to solve these geometric and photometric calibration problems is by incorporating one or more cameras into the system [16, 22, 13].

The above efforts have significantly enhanced the capabilities of projectors. However, there remains an open problem that severely limits the use of projectors. Any projection system requires a high quality screen to ensure that its output is pleasing to the observer. We believe that relaxing this requirement will make the projector a more useful and more widely used device. Clearly, the need for a screen is inherent to the notion of a projection system. However, if a projector can project its images onto virtually any surface (walls, doors, drapes, ceilings, etc.) without sacrificing the photometric quality of its output, it immediately becomes a more versatile and powerful device.

The fundamental problem with using an arbitrary surface for a screen is that the surface cannot be assumed to be a white, matte one. It is bound to have its own colors and textures (bricks of a wall, painting or poster on a wall, tiles of a ceiling, grain of a wooden door, etc.) or at least surface markings (paint imperfections, scratches, nails, etc.). As a result, when an image is projected onto an arbitrary surface, the appearance of the image is modulated by the spatially varying reflectance

*This work was conducted at the Computer Vision Laboratory at Columbia University. It was supported by an NSF ITR Award (No. IIS-00-85864).

properties of the surface. Humans are very sensitive to such modulations. One may assume that the problem can be more or less remedied if the projector is powerful (bright) enough. However, the modulation due to the surface is a multiplicative effect and hence increasing the brightness does not change the proportion of the modulation.

In this paper, we present a projector-camera system that displays images onto an arbitrary surface such that the quality of the images is preserved and the effects of the surface imperfections are minimized (ideally, made invisible to the observer). The development of such a system requires us to build a detailed radiometric model that takes us all the way from a display image to the image captured by the camera. Based on this model, we develop a simple and yet effective calibration method that obtains the required data without knowing the radiometric parameters of the individual components of the system. The calibration scheme only involves the projection and capture of a set of images. The calibration results are used on-line to compensate each display image prior to projection. The compensation step is very simple and hence can be done at frame-rate. The end result is that the imperfections of the arbitrary projection surface are made to vanish and the surface behaves like a high quality screen. We show several experimental results that demonstrate the advantages of using our radiometric compensation algorithm.

Finally, we describe a compensation algorithm that does not use the off-line radiometric calibration but instead a continuous (on-line) feedback approach. This algorithm is simple but has the disadvantage that, for any given display image, a few iterations (display and capture) are needed to achieve compensation. However, the feedback algorithm can be very effective when used with radiometric compensation; feedback is used to only correct for errors (residues) in the radiometric compensation.

2 Radiometric Model of a Projector-Camera System

The projector-camera system we have used in our work is shown in Figure 1. The projector is a Sony SVGA VPL-CS5 model. It has a native resolution of 800×600 pixels. We have constructed a large variety of textured screens (one of which is shown in the figure) for testing our algorithms. These screens are in the form of posters that can be easily changed during experimentation. The camera we have used is a Sony DXC 950 Power HAD model with a resolution of 640×480 pixels. Our algorithms are run on a Dell Precision 330 computer with a Pentium P4 (1.8 GHz) processor and 1 Gb of memory. Images are sent out to the projector via a ATI Radeon VE display card and images from the camera are captured using a Matrox Meteor II frame-grabber.

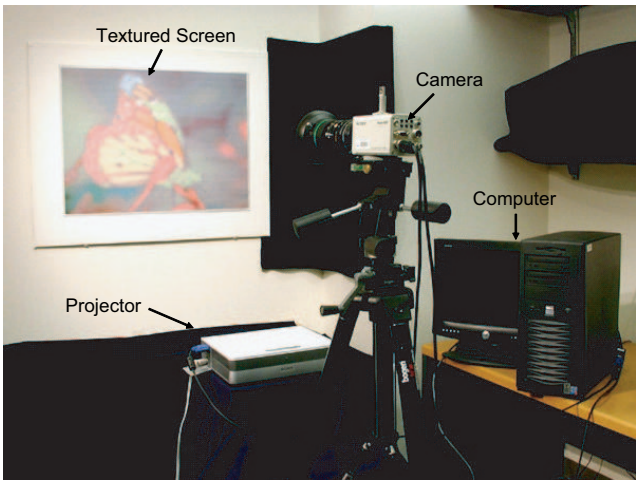


Figure 1: The projector-camera system we have used in our experiments. Images are displayed using a Sony SVGA VPL-CS5 projector and captured using a Sony DXC 950 Power HAD camera. The calibration and compensation algorithms are run on a Dell Precision 330 computer that uses an ATI Radeon VE card to output images and a Matrox Meteor II frame-grabber to capture images.

Figure 2 shows the complete dataflow pipeline for the projector-camera system. An image I chosen by a user is processed by the display device before it is received by the projector. The projected image is reflected by the screen and captured by the camera. The output of the camera is then digitized by the capture device to obtain the final measured image M. Although our focus is on the radiometric properties of this system, we rely on knowing the correspondence between points in the display image I and the measured image M. We briefly describe the calibration we use to determine the geometric mapping between points in the two images[1] (see [15] for details).

We model the mapping between the two coordinate frames using piecewise second-order polynomials. Let a point $\mathbf{x}_i = [x_i, y_i]^T$ in the display image map to the point $\mathbf{x}_m = [x_m, y_m]^T$ in the measured image, as shown in Figure 2. The polynomial model can be concisely

[1]Note that a projector-camera system can be designed such that the mapping between the displayed and acquired images is fixed and is unaffected by the location or the shape of the screen. This is achieved by making the optics of the projection and the imaging systems coaxial. For instance, the same lens can be used by the projector and the camera by means of a beam-splitter placed behind the lens. Alternatively, two different lenses can be used for the projector and the camera with a beam-splitter placed in front of the two lenses. In these cases, there is no need for geometric calibration. The use of coaxial optics has the added benefit that all points that are visible to the projector are also visible to the camera; there is no possibility of occlusion.

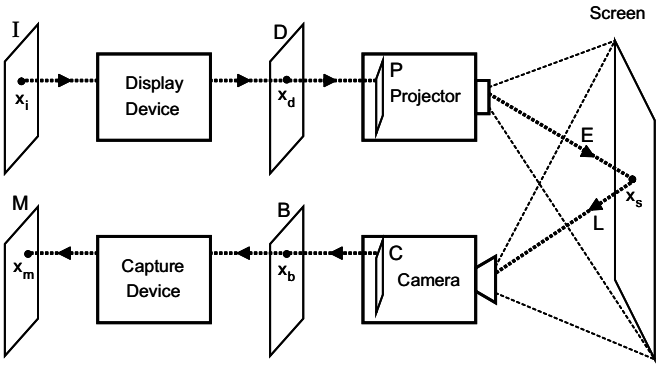


Figure 2: The complete geometric and radiometric dataflow pipeline for a projector-camera system.

written as:

$$\begin{aligned} \mathbf{x}_i &= A\,\hat{\mathbf{x}}_m\,, \\ \mathbf{x}_m &= B\,\hat{\mathbf{x}}_i\,, \end{aligned} \tag{1}$$

where:

$$\begin{aligned} \hat{\mathbf{x}}_m &= [{x_m}^2 \;\; {y_m}^2 \;\; x_m y_m \;\; x_m \;\; y_m \;\; 1]^T\,, \\ \hat{\mathbf{x}}_i &= [{x_i}^2 \;\; {y_i}^2 \;\; x_i y_i \;\; x_i \;\; y_i \;\; 1]^T\,. \end{aligned} \tag{2}$$

A and B are 2×6 matrices that include the unknown coefficients of the mappings. Given a set of corresponding points in the two images, the coefficient matrices A and B are easily computed using the least squares method. The corresponding points are obtained by projecting 1024 square patches uniformly spaced in the display domain. The patches are efficiently scanned by using binary coding; the correspondences for all 1024 patches are obtained using just 10 projected images [15].

Instead of using a single polynomial model for the entire projection area, we have divided the camera domain into $4 \times 4 = 16$ blocks and computed a separate model for each block. This piecewise approach produces very accurate results as it can accommodate for geometric distortions that may be caused by the optics of the projector and the camera as well as the surface being smoothly curved rather than planar. The final geometric mappings (both ways) between the projector and the camera are stored as look-up tables; each point in one domain is used as an index to obtain the corresponding point in the other. Our piecewise polynomial model results in very high mapping accuracy. The maximum and RMS errors produced by the mapping were less that 0.6 display pixels and 0.19 display pixels, respectively [15].

With the geometric mapping in place, we can now focus our attention on the radiometric model. Once again, consider Figure 2. We will develop the model for a single point on the screen, noting that the same model can be used (with possibly different parameters) for any other point on the screen. Note that each of the devices in our system will have its own unknown, non-linear radiometric response. Since the process of radiometric compensation requires us to invert these responses, we will assume that the individual responses are monotonic. This is a reasonable assumption as all the devices are expected to increase in output with input.

The projector and camera may have multiple spectral (color) channels. For now, let us assume that the projector has only a single channel denoted by K. The pixel value I_K in the display image is transformed to D_K by the radiometric response d_K of the display device:

$$D_K = d_K(I_K)\,. \tag{3}$$

The output of the display device is mapped by the radiometric response of the electronics of the projector to a projector brightness value as

$$P_K = p_K(D_K)\,. \tag{4}$$

This projector brightness is modulated by the spectral response $w_K(\lambda)$ of the projector channel to produce the screen irradiance

$$E_K(\lambda) = P_K\, w_K(\lambda)\,. \tag{5}$$

Let the spectral reflectance of the irradiated screen point be $s(\lambda)$ in the viewing direction of the camera. Then, the radiance of the screen point in the direction of the camera can be written as

$$L_K(\lambda) = P_K\, w_K(\lambda)\, s(\lambda)\,. \tag{6}$$

Now, let us assume that radiance of the screen point is being measured by a camera with a single spectral channel L with quantum efficiency $q_L(\lambda)$. Then, the irradiance detected by the camera detector is:

$$C_L = P_K \int w_K(\lambda)\, s(\lambda)\, q_L(\lambda)\, d\lambda\,. \tag{7}$$

This irradiance is processed by the electronics of the camera to produce the final camera output

$$B_L = b_L(C_L)\,, \tag{8}$$

where b_L is the radiometric response of the camera. Finally, the camera output is mapped to the final measured brightness by the capture device (frame-grabber):

$$M_L = m_L(B_L)\,. \tag{9}$$

The above expressions, together, give us the relation between brightnesses in the display image and the final captured image. With this model in place, we can explore what happens in the case of multiple color channels. It is important to note that the spectral responses

of the projector and camera channels can be arbitrary and are unknown to us. Let us assume that the projector and the camera each have three color channels (R, G, B)[2]. Then, we can extend the above radiometric model and write it compactly using vectors and matrices as

$$\mathbf{C} \;=\; \mathbf{V}\,\mathbf{P}\,, \tag{10}$$

where:

$$\mathbf{C} = \begin{bmatrix} C_R \\ C_G \\ C_B \end{bmatrix}, \mathbf{V} = \begin{bmatrix} V_{RR} & V_{RG} & V_{RB} \\ V_{GR} & V_{GG} & V_{GB} \\ V_{BR} & V_{BG} & V_{BB} \end{bmatrix}, \mathbf{P} = \begin{bmatrix} P_R \\ P_G \\ P_B \end{bmatrix},$$

$$V_{KL} = \int w_K(\lambda)\, s(\lambda)\, q_L(\lambda)\, d\lambda\,,$$
$$P_K = p_K(\, d_K(\, I_K\,)\,)\,,$$
$$C_L = {b_L}^{-1}(\, {m_L}^{-1}(\, M_L\,)\,)\,.$$

The matrix $\mathbf{V}$ is referred to as the color mixing matrix. A few observations about the above model are worth making. Note that the spectral responses of the color channels of the projector and the camera can overlap with each other in an arbitrary way. The couplings between the projector and camera channels and their interactions with the spectral reflectance of the screen point are all captured by the matrix $\mathbf{V}$. Most importantly, $\mathbf{V}$ does not include the non-linear response functions of the individual components of the projector-camera system. Hence, the model nicely decouples brightness non-linearities of the system from the spectral characteristics of the system[3]. This fact will come in handy when we develop our compensation algorithms.

3 Compensation Algorithms

With the radiometric model in place, we are in a position to develop techniques for screen compensation. We will begin with the simple case where the projector and the camera have a single gray channel. Then, we will discuss how color can be handled. As with the radiometric model, we will describe our compensation algorithms for a single pixel, bearing in mind that all pixels are treated the same way.

[2]Our model is in no way restricted to three channels. For instance, the projector may have 3 channels while the camera has 5 channels (multi-spectral). The expressions given here directly generalize to such cases.

[3]In our experiments, we have used a DLP projector. It is known that a DLP projector adds a "white" component that is a function (possibly non-linear) of the R, G, B color values (see [20] for details). For this case, the color mixing is more complex. However, our experimental results indicate that the above model works as a reasonable approximation.

3.1 Gray World

Consider the special case where the projector only outputs gray-scale images (equal values in the R, G, and B channels), the camera is a black and white one with a broad spectral response, and the screen is gray with possibly varying albedo. We will refer to this as the "gray world." In this case, the projector image can be represented by a single brightness P_{BW} and the projector's spectral response is given by

$$w_{BW}(\lambda) = w_R(\lambda) + w_G(\lambda) + w_B(\lambda)\,. \tag{11}$$

Since the screen is gray, its spectral response is given by $s(\lambda) = \rho$, where ρ is the albedo of the screen point. The black and white camera has a quantum efficiency $q_{BW}(\lambda)$ and produces a single brightness value C_{BW}. Therefore, the model in equation 10 can be written as

$$C_{BW} = V_{BW}\, P_{BW}\,, \tag{12}$$

where:

$$V_{BW} = \sum_{K=R,G,B} \sum_{L=R,G,B} V_{KL}\,.$$

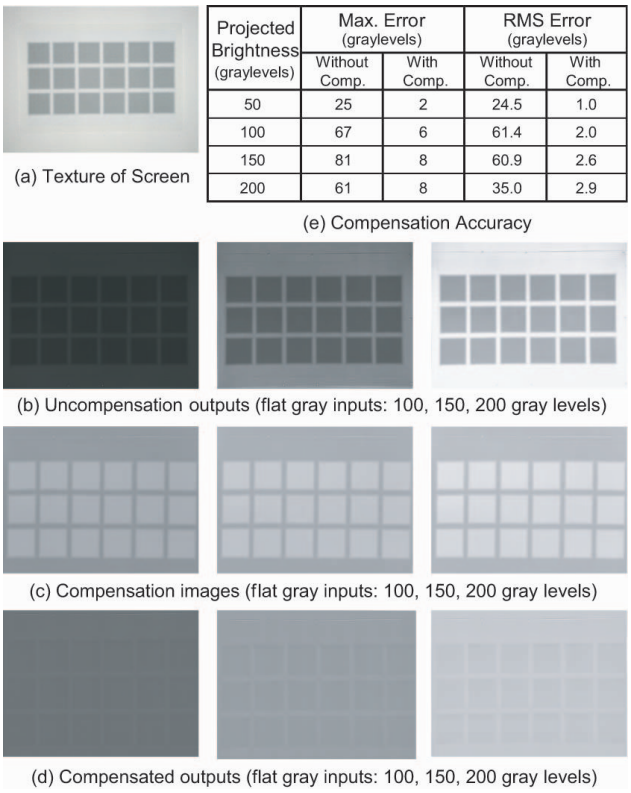
In this special case, we see that the radiometric model for the complete system can be represented using a single non-linear monotonic response function:

$$M_{BW} = h_{BW}(I_{BW})\,, \tag{13}$$

where h_{BW} includes the non-linear effects of all the individual components of the system; $h = m_{BW} \circ b_{BW} \circ V_{BW} \circ p_{BW} \circ d_{BW}$, where $\circ$ denotes composition.

Note that if we can determine the response h_{BW} we can compute the display image brightness I_{BW} needed to produce any desired measured image brightness M_{BW}. We use a simple and fast calibration procedure to obtain the response h_{BW}. A set of 255 display images (in the case of an 8-bit per channel projector) are displayed in quick succession and their corresponding camera images are recorded[4]. For each pixel (in the camera or the projector domain) we have a densely sampled response function. This discrete function is then inverted to obtain discrete (possibly non-uniform) samples of the inverse response ${h_{BW}}^{-1}$. We preprocess the inverse response samples to ensure the function is monotonic; for each sample that makes it non-monotonic, the closest monotonic value is used. Then, we interpolate the new samples to obtain a continuous inverse response function. This function is uniformly sampled and stored as a 1D lookup table.

[4]Note that we are being conservative here. A smaller number of images can be projected as the response function is always smooth and can be interpolated from more widely spaced measurements without sacrificing accuracy.



(a) Texture of Screen

Projected Brightness (graylevels)	Max. Error (graylevels)		RMS Error (graylevels)	
	Without Comp.	With Comp.	Without Comp.	With Comp.
50	25	2	24.5	1.0
100	67	6	61.4	2.0
150	81	8	60.9	2.6
200	61	8	35.0	2.9

(e) Compensation Accuracy

(b) Uncompensation outputs (flat gray inputs: 100, 150, 200 gray levels)

(c) Compensation images (flat gray inputs: 100, 150, 200 gray levels)

(d) Compensated outputs (flat gray inputs: 100, 150, 200 gray levels)

Figure 3: Gray world experimental results for a screen with dark squares. (a) An image of the textured screen taken under ambient illumination. (b) Uncompensated outputs measured by the camera for flat-gray input display images. (c) The compensation images computed by the gray world algorithm. (d) The compensated output images measured by the camera. (e) Comparison between results obtained without and with compensation.

Now, consider a novel (original) display image $\mathbf{I}$[5]. If this image is passed on to the projector, the system would measure the "uncompensated output" $\mathbf{M}$. To apply compensation, we assume that the image we would like the system to measure exactly equals the original display image $\mathbf{I}$. The calibrated inverse response ${h_{BW}}^{-1}$ is used to compute the "compensation image" $\tilde{\mathbf{I}}$. When we apply the compensation image, we get the final "compensated output" $\tilde{\mathbf{M}}$.

Figure 3 shows results obtained using the above algorithm for a screen with dark gray squares. Figure 3(b) shows the uncompensated output images measured for three flat-gray original display images. As expected, the squares on the screen are clearly visible. Figure 3(c) shows the compensation patterns computed by the algorithm and Figure 3(d) shows the corresponding compensated outputs. Note how the squares on the screen have more or less vanished in these output images. The compensation accuracy is summarized in the table in Figure 3(e), where maximum and RMS errors (in gray levels) are given for the uncompensated and the compensated outputs for four different flat-gray input images. The RMS errors are less than 3 gray levels in the compensated case, while they are above 60 gray levels in some of the uncompensated cases. The errors for the input image with brightness 200 are lower than in some of the other cases because the uncompensated image in this case is saturated (clipped) in many regions.

[5]Even though these are gray-scale images, we use vectors and treat them like color images to stay consistent with the notation used in the subsequent sections.

(a) Original (desired) image (I) (b) Uncompensated output (M)

(c) Compensation image ($\tilde{I}$) (d) Compensated output ($\tilde{M}$)

Figure 4: Results for a face image displayed on the screen with squares shown in Figure 3(a). There is no perceptible difference between the original image and the compensated output.

Even though the errors in the compensated images are very small, one can see faint squares in the compensated images in Figure 3(d). This is because the human observer is very sensitive to sharp discontinuities in the images [7]. For this reason, in all our experiments, we have avoided using surface marking with very sharp edges. Such edges cannot be compensated for in our current system because the camera used is of lower resolution than the projector.

It is worth noting that the use of flat-gray test images is the most stringent test of the system. This is because the test image itself does not include any features that divert the observer's attention from imperfections in the compensated image. In short, for an arbitrary image the results always look better than in the flat-gray case. Figure 4 shows results for a face image projected on the screen in Figure 3. Note that the compensated output looks indistinguishable from the original one.

3.2 Handling Color

We now address the general and more complex case of color. From equation 10 we see that if we had corresponding pairs of projector colors $\mathbf{P}$ and camera colors

$\mathbf{C}$, we can compute the matrix $\mathbf{V}$ using the least squares method. However, such pairs of values are impossible to obtain without prior calibration of the camera and the projector. It is exactly this type of full calibration that we need to avoid to ensure that our calibration method remains simple and efficient. It turns out that we can compute the matrix $\mathbf{V}$ without any prior knowledge regarding the projector, if the response function of the imaging system (camera plus capture device) is known. This response needs to be determined only once and can be done in many ways; a calibration chart may be used or multiple exposures of an unknown scene (perhaps, the screen) may be used [14]. Once this is done, we can map any measured color $\mathbf{M}$ to the corresponding color $\mathbf{C}$ detected by the camera.

Our calibration for the color case has two stages. First, we compute the matrix $\mathbf{V}$ for each pixel. Then, we compute the non-linear response of the projector for each pixel. These two calibrations are all we need to compensate for the screen. To compute matrix $\mathbf{V}$ we will constrain its diagonal elements to be equal to unity, i.e. $V_{KK} = 1$. This does not restrict us in any way as fixing the diagonal elements can be viewed as introducing unknown scale factors associated with each of the three rows of the matrix. These scales can be absorbed by the unknown radiometric responses on the projection side of the system. Now consider applying two different display colors at a pixel where the two colors only differ in one of the three channels, say the red channel:

$$\mathbf{I}^{(1)} = \begin{bmatrix} {I_R}^{(1)} \\ {I_G}^{(1)} \\ {I_B}^{(1)} \end{bmatrix}, \ \mathbf{I}^{(2)} = \begin{bmatrix} {I_R}^{(2)} \\ {I_G}^{(1)} \\ {I_B}^{(1)} \end{bmatrix}. \qquad (14)$$

From equation 10, we have:

$$\begin{bmatrix} {C_R}^{(1)} \\ {C_G}^{(1)} \\ {C_B}^{(1)} \end{bmatrix} = \mathbf{V} \begin{bmatrix} {P_R}^{(1)} \\ {P_G}^{(1)} \\ {P_B}^{(1)} \end{bmatrix}, \begin{bmatrix} {C_R}^{(2)} \\ {C_G}^{(2)} \\ {C_B}^{(2)} \end{bmatrix} = \mathbf{V} \begin{bmatrix} {P_R}^{(2)} \\ {P_G}^{(1)} \\ {P_B}^{(1)} \end{bmatrix}. \qquad (15)$$

Since we have changed only the red channel of the input, the corresponding changes in the three channels are simply:

$$\begin{aligned} \Delta C_R &= V_{RR} \Delta P_R \,, \\ \Delta C_G &= V_{RG} \Delta P_R \,, \\ \Delta C_B &= V_{RB} \Delta P_R \,. \end{aligned} \qquad (16)$$

Since $V_{RR} = 1$, we have $\Delta P_R = \Delta C_R$. Hence,

$$V_{RG} = \frac{\Delta C_G}{\Delta C_R} \text{ and } V_{RB} = \frac{\Delta C_B}{\Delta C_R}. \qquad (17)$$

Similarly, the unknown elements of $\mathbf{V}$ corresponding to the green and blue channels are obtained by using two display images for each of those channels. A few points are worth making about the above procedure. (a) The $\mathbf{V}$ matrices for all pixels are computed by simply projecting 6 images (two per channel)[6]. (b) The exact values used in these display images are not important as they are never used in the computation of $\mathbf{V}$. (c) If the matrix needs to be computed with very high accuracy, more that 6 display images can be used and the expressions in equation 17 can be used to estimate the matrix elements using the least squares method.

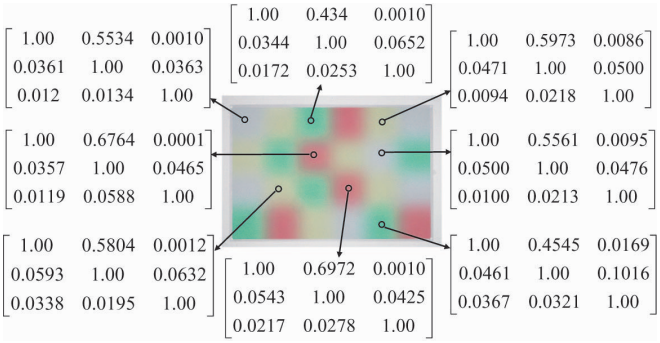


Figure 5: A screen with strong colors. The color calibration algorithm was used in this case. The $\mathbf{V}$ matrices for several of the screen regions are shown. For all regions V_{RG} is high, indicating that color mixing must be taken into account to achieve good compensation.

Once the matrix $\mathbf{V}$ has been computed, we can take any measured color $\mathbf{M}$, map it to the camera color $\mathbf{C}$ (using the camera calibration results), and then multiply the result with $\mathbf{V}^{-1}$ to obtain the projector color $\mathbf{P}$. The projector color is related to the display image color as:

$$\begin{aligned} P_R &= g_R(I_R) \,, \\ P_G &= g_G(I_G) \,, \\ P_B &= g_B(I_B) \,, \end{aligned} \qquad (18)$$

where $g_K = p_K \circ d_K$ are the composite non-linear radiometric responses of the channels of the projection system. Note that these responses and their inverses can now be independently computed using exactly the same calibration procedure used in the gray world case in section 3.1.

Figure 5 shows a screen with highly saturated colors. One can appreciate that this is a much more difficult screen to deal with than the previous gray-scale one. We applied the above calibration algorithm and the $\mathbf{V}$ matrices computed for several of the colors on the screen are shown in the figure. Note that the matrices for squares with the same color tend to be very similar. More importantly, V_{RG} is high for all the squares, indicating that the red channel of the projector overlaps significantly with the green channel of the camera.

[6]Actually, 4 images suffice as the initial image for all three channels can be the same.

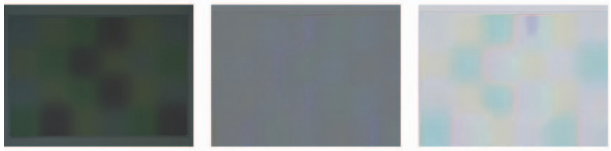
(a) Texture of Screen

Projected Brightness (R, G, B)	Max. Error (R, G, B)		RMS Error (R, G, B)	
	Without Comp.	With Comp.	Without Comp.	With Comp.
50,50,50	25,26,26	2,3,7	24.1,24.2,24.5	0.6, 0.8, 1.7
100,100,100	66,70,71	6,8,14	58.9,57.2,64.9	1.6, 1.9, 5.4
150,150,150	75,98,95	19,15,21	49.0,44.5,69.1	4.8, 3.1, 5.8
200,200,200	50,99,90	39,29,25	27.7,47.5,42.7	12.7,4.0,10.7

(e) Compensation Accuracy

(b) Uncompensation outputs (flat gray inputs: 100, 150, 200 gray levels)

(c) Compensation images (flat gray inputs: 100, 150, 200 gray levels)



(d) Compensated outputs (flat gray inputs: 100, 150, 200 gray levels)

Figure 6: Experimental results for a screen with strong colors. The compensation algorithm does well in producing near-gray images except in the 200 gray-level case where the projector output saturates in some regions of the image as the projector does not have enough power to fully compensate for some of the colors.

Figure 6 shows results for flat-gray display images projected onto the same screen. The compensated outputs are quite consistent with the desired output except in the 200 gray-level case where the projector output saturates in many regions as it simply does not have the power to generate the required colors. In Figure 7 we see results of projecting a face image onto the same screen. Overall, the quality of the compensated output is very good except for the top left corner of the image for which the projector output saturates.

More experimental results are shown in Figures 8 and 9. In these case, the screen not only has strong colors but also stronger edges. As we discussed earlier, humans are very sensitive to compensation errors around strong edges. This effect can be observed in the compensated output images in both Figure 8 as well as Figure 9.

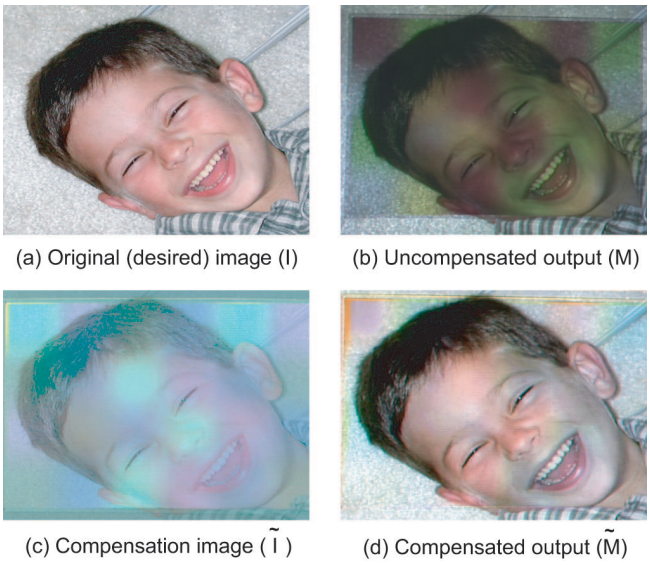
3.3 Refinement by Continuous Feedback

We conclude by describing a simple closed-loop compensation algorithm where the appearance of the screen is continually measured by the camera and these measurements are used on-line to compensate the display images. Let $\mathbf{I}(t)$ be the original display image to be shown at time t and let $\mathbf{M}(t)$ be the corresponding measured image. The compensated display image for time $t+1$ can be computed as:

$$\tilde{\mathbf{I}}(t+1) = \tilde{\mathbf{I}}(t) + \alpha(\mathbf{I}(t) - \mathbf{M}(t))\,, \qquad (19)$$

where $\tilde{\mathbf{I}}(0) = \mathbf{I}(0)$. α is the feedback gain and it lies between 0 and 1.

Several experimental results for the feedback algorithm are presented in [15]. This algorithm is very simple and works well but has the disadvantage that it takes several iterations (frames) to converge. During this convergence process, the input display image may change. This could result in large compensation errors. However, the feedback algorithm can be very effective when used with radiometric compensation; feedback is used to only correct for errors (residues) in the radiometric compensation. Since these errors are expected to be small, the convergence time required by the feedback algorithm becomes less critical.



(a) Original (desired) image (I) (b) Uncompensated output (M)

(c) Compensation image ($\tilde{I}$) (d) Compensated output ($\tilde{M}$)

Figure 7: Results for a face image projected onto the screen in Figure 6.

References

[1] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proc. of SIGGRAPH*, pages 135–142, Anaheim, CA, 1993.

[2] R. Holmes. Common projector and display modules for aircraft simulator visual systems. In *IMAGE V Conference, The Image Society*, pages 81–88, Phoenix, AZ, Jun. 1990.

[3] G. Humphreys and P. Hanrahan. A distributed graphics system for large tiled displays. In *Proc. IEEE Visualization*, pages 215–223, 1999.

[4] M. Inami, N. Kawakami, D. Sekiguchi, Y. Yanagida, T. Maeda, and S. Tachi. Visuo-haptic display using head-mounted projector. In *Proc. of the IEEE VR*, pages 233–240, 2000.

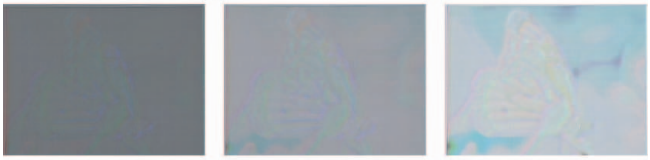
(a) Texture of Screen

Projected Brightness (R, G, B)	Max. Error (R, G, B)		RMS Error (R, G, B)	
	Without Comp.	With Comp.	Without Comp.	With Comp.
50,50,50	25, 26, 26	6,6,9	24.7,25.8,24.8	0.9,1.5,1.3
100,100,100	70, 71, 72	21,20,26	65.4,65.3,67.0	1.8,2.7,3.2
150,150,150	100,109,108	30,15,29	76.8,73.6.79.9	5.2,4.1,5.0
200,200,200	109,119,126	64,21,64	62.1,52.2,61.5	13.4,5.3,6.7

(e) Compensation Accuracy

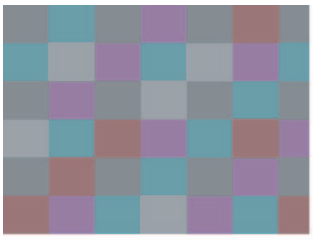
(b) Uncompensation outputs (flat gray inputs: 100, 150, 200 gray levels)

(c) Compensation images (flat gray inputs: 100, 150, 200 gray levels)

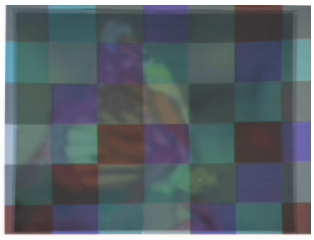


(d) Compensated outputs (flat gray inputs: 100, 150, 200 gray levels)

Figure 8: Experimental results for a screen with the poster of a butterfly. This screen not only includes strong colors but also sharp edges. As a result, the errors in the compensation are more visible to the observer, even though the actual errors are no greater than for the screen in Figure 6.

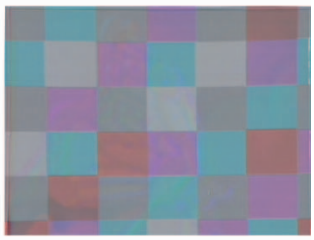


(a) Original (desired) image (I)



(b) Uncompensated output (M)

(c) Compensation image ($\tilde{I}$)



(d) Compensated output ($\tilde{M}$)

Figure 9: Results for a colored checkerboard image projected onto the butterfly screen in Figure 8. While the uncompensated output is of very poor quality, the compensated output is quite close to the desired one.

[5] P. Inova. Seamless video display, #4,974,073. US Patent, 1988.

[6] K. Jarvis. Real-time 60hz Distortion Correction on a Silicon Graphics IG. *Real-time Graphics*, 5(7):6–7, Feb. 1997.

[7] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of Optical Society of America*, 61(1):1–11, Jan. 1971.

[8] K. Li and Y. Chen. Optical blending for multi-projector display wall system. In *IEEE Proc. 12th Lasers and Electro-Optics Society*, volume 1, pages 281–282, 1999.

[9] P. Lyon. Edge-blending multiple projection displays on a dome surface to form continuous wide angle fields-of-view. In *Proc. of 7th I/ITEC*, pages 203–209, 1985.

[10] A. Majumder. Intensity seamlessness in multiprojector multisurface displays. Technical report, University of North Carolina, Chapel Hill, 1999.

[11] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proc. IEEE Visualization*, pages 117–124, 2000.

[12] A. Majumder and G. Welch. Computer graphics optique: Optical superposition of projected computer graphics. In *Fifth Immersive Projection Technology Workshop*, May 2001.

[13] A. Majumder and R. Stevens. Lam : Luminance attenuation map for photometric uniformity across a projection based displays. In *ACM Virtual Reality and Software Technology*, pages 147–154, 2002.

[14] T. Mitsunaga and S. K. Nayar. Radiometric Self Calibration. In *Proc. of Computer Vision and Pattern Recognition '99*, volume 1, pages 374–380, June 1999.

[15] S. K. Nayar, H. Peri, M. Grossberg, and P. Belhumeur. A projector with automatic radiometric screen compensation. Technical report, Department of Computer Science, Columbia University, New York, 2003.

[16] R. Raskar, M.S. Brown, R. Yang, W. Chen, H. Towles, B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In *Proc. IEEE Visualization*, pages 161–168, Oct. 1999.

[17] 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. *Proc. of SIGGRAPH*, pages 179–188, 1998.

[18] R. Raskar, G. Welch, K-L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image based illumination. In *Eurographics Workshop on Rendering*, pages 89–102, 2001.

[19] M. Stone. Color and brightness appearance issues in tiled displays. *Comp. Graph. and Appl.*, 21(6):58–67, Sept. 2001.

[20] M. Stone. Color balancing experimental projection displays. In *Proc. of the 9th Color Imaging Conference, Scottsdale, AZ*, number 5-9, pages 342–347, Nov. 2001.

[21] R. Sukthankar, T.J. Cham, and G. Sukthankar. Dynamic shadow elimination for multi-projector displays. In *IEEE Comp. Vis. and Patt. Recog.*, pages II:151–157, 2001.

[22] R. Yang, D. Gotz, J. Hensley, H. V, and M. Brown. Pixelflex: A reconfigurable multi-projector display system. In *Proc. IEEE Visualization*, pages 167–174, Oct. 2001.