Avoiding Seams On High-Resolution Multi-Projector Displays Using An Un-calibrated Camera

Yuqun Chen, Douglas W. Clark, Adam Finkelstein, Timothy Housel, and Kai Li
Department of Computer Science, Princeton University, Princeton, NJ 08544
{yuqun, doug, af, li}@cs.princeton.edu
THousel@aol.com

Abstract

A scalable, high-resolution display may be constructed by tiling many projected images over a single display surface. One fundamental challenge for such a display is to avoid visible seams due to both misalignment and variations in color balance among the projectors. Traditional methods for avoiding seams involve sophisticated mechanical devices and expensive CRT projectors, coupled with extensive human effort for fine-tuning the projectors. This paper describes three methods that facilitate seamless imaging on a multi-projector display: automatic alignment, automatic color balance, and optical blending. These methods rely on an inexpensive, uncalibrated camera to measure the relative mismatches between neighboring projectors, and then correct the projected imagery to avoid seams without significant human effort.

1 Introduction

Visualization of large data has become available and indispensable due to rapid advances in computer technology. While the cost-performance ratio for many key enabling technologies has been improving at or beyond the rate predicted by Moore's Law, display resolution – a key aspect of an effective information system – is lagging far behind. Monitors are still the dominant display technology through which people visualize information. Their resolutions have been increasing at a mere 5% annual rate for the last two decades.

One strategy for overcoming limited resolution is to tile multiple projectors over a single display surface. Several commercial vendors and research labs [9] have employed this approach to build small-scale multi-projector displays using 3 or 4 projectors. However, most previous systems are not scalable and cannot employ many projectors to achieve tens of millions of pixels. The two dominant limitations are: (1) the use of expensive high-end computer systems to drive the projectors, and (2) manual alignment and color balancing among multiple projectors. The Scalable DisplayWall Project at Princeton University investigates research issues in building a scalable Display Wall systems out of commodity projectors and clusters of PCs [1]. Our clustering approach can in principle scale to drive a multi-projector Display Wallwith scores of projectors and up to hundreds of millions of pixels.

The two key challenging problems in building such a scalable display wall are achieving pixel-level image alignment across overlapped regions and color uniformity across projectors.

A multi-projector display might in principle be perfectly aligned by manual means just once, but in practice physical realities (vibration, lamp-changing, and so on) mean that re-alignment is frequently needed. In addition, manually fine-tuning projector mounts to achieve good alignment is a time-consuming task that requires both skill and experience. It also requires the use of either sophisticated mechanical positioning devices or projection geometry adjustment found only on expensive CRT projectors. The high cost of the manual approach can hinder wide deployment of high-resolution multi-projector displays.

Similarly, due to small variations in lamp ages, optical paths, and imaging devices (LCD and DMD), the color temperatures vary, often noticeably, across projectors. In the past color-balancing multiple projectors was also done by hand, using sophisticated color tuning mechanism such as found on high-end CRT projectors and requiring trained human experts. Color-matching two or three projectors could be manageable, though already time-consuming. Doing so on a large projection array with tens of projectors is a daunting, if not impossible task.

We have developed *automatic alignment and color-balancing* method that use an uncalibrated camera to determine the amount of misalignment and color imbalance amongst the projectors. Our algorithm then computes appropriate *digital compensation* for each projector in both geometry and color spaces to counter the effect of actual physical misalignment and color imbalance. The digital compensation, in the form of pre-distortion and color-adjustment, is applied to each image tile before it is sent to the projector so that when the final image displayed by all projectors is perfectly aligned and color consistent.

Our algorithm introduces a novel technique of taking only relative measurements such as point and line relationships between adjacent projectors, as opposed to absolute measurements such as measurement against a physical grid, or a well-calibrated camera, or an expensive colorimeter.

The benefits are threefold. First, our algorithm obviates completely the need for human involvement in setting up the absolute measurement environment; all that is required is to place the camera(s) in front the Display Wall. Second, our algorithm allows the camera to zoom arbitrarily close to the display area of interest to obtain the measurements with finest accuracy, thus overcoming the huge resolution gap between cameras and a large-scale Display Wall. Third, the measurements in our algorithm only require the camera field of view (FOV) to be linear in some cases. This requirement can be easily met by zooming the camera close and using only its center FOV. Due to these three advantages, our algorithm can align and color-balance multi-projector high-resolution displays with little or no human involvement and at a low cost. We only require the coarsest physical alignment of the projectors and placement the camera(s) in front of the display wall.

In addition to these computational methods, we also developed an optical edge-blending technique that can create a smooth luminance transition between adjacent projectors. The traditional method to edge-blend images is to digitally alter the image source. This approach has problems with commodity projectors which leak substantial amount of light even when they are supposed to display a pitch black image. The result is a distinct gray stripe in the overlapped region when the projectors project black. Our optical edge-blending method operates directly on the light emitted by the projector. It does not suffer from the same problem that a digital blending method does.

The rest of the paper describes the details and implementation results of our automatic alignment and color-balancing algorithm as well as our optical edge-blending technique.

2 Related Work

Multi-projector displays have been around for at least a decade [9]. Previous systems typically employed expensive CRT projectors, because they offer sophisticated mechanisms to adjust the image distortion and color balance. These systems required manual adjustments on each CRT to align projectors. Recent interest in bringing high-resolution displays to offices and entertainment arenas have motivated several research projects to address the seamlessness issue [15, 5]. The algorithms developed employ a common two-stage process: camera calibration and geometric registration. The first stage calibrates one or more **fixed** cameras according to a fixed global coordinate system (either 2-dimensional or 3-dimensional). The calibrated cameras are used as precision measurement instruments to map pixels from each projector to the global coordinate system.

Surati and Knight developed a vision-based algorithm to register pixels from a multi-projector array in a pre-established global screen coordinate system [15]. Their method consists of two stages. During the first stage, a camera is calibrated against a precision grid affixed to the display surface. The grid is physically drawn by a high-precision plotter. It helps the vision software establish a mapping between pixels in the camera field and the physical points on the display surface. In the second stage, each projector projects a regular grid onto the display surface. A computer vision algorithm acurately locates each projected grid point in the camera's field of view. Using the camera-to-display-surface mapping established previously, the projected grid point (or a pixel in the projector) is mapped to a physical point on the display surface with high precision. This method works pretty well for a small-scale display wall. Its drawback is the reliance on calibrating the camera using an absolute measurement grid. For a large display wall, it is problematic whether one can generate a physical or project a virtual measurement grid that is large enough but still have fine precision.

Raskar *et al.* attempted to solve a general case in which the display surface can be arbitrarily complex, for example, the corner of a wall, or a curved screen [5]. This requires registration of the 3D surface geometry of the screen surface as well as registration of the projected pixels on the screen surface. Their algorithm uses known 3D objects such as painted boxes to calibrate the extrinsic and intrinsic parameters for a set of fixed cameras. Two calibrated cameras that overlap in their fields of view can observe the same mesh pattern displayed by a projector. The observations from both cameras are correlated using the stereo vision technique to derive the exact location of a projected pixel on the display surface. The location information is in the 3-dimensional space and therefore also reveals the surface contour of the screen. The drawback of this approach is again the requirement of camera calibration.

Rencently Majumder et al studied the problem of color-matching a multi-projector display [8]. Their approach is very similar to ours. The luminance of the projectors are first matched using a precision color-spectrometer. Then the luminance of each color channel is matched separately, again with a precision color-spectrometer. They demonstrated good color matching between two commodity projectors. The key component of their method is an expensive color-spectrometer, which gives very accurent measurements. Operating the spectrometer is also time-consuming, as it is generally not designed for large-scale automatic data gathering.

Previous systems of seamless multi-projector displays employed what is known as "edge blending" to reduce the perceptual seams between projectors. In this method, adjacent projectors are overlapped by some amount, say, 10 %. The image intensity in the overlapped region is brought down gradually to zero, so that the combined image intensity is a smooth curve that can bridge two projectors with different color characteristis without sharp transitions. Edge-blending is typically done digitally by altering the image source either at the video signal level or in the raw image themselves [9, 14, 15]. This approach cannot easily deal with light leakage problem found on commodity projectors without reducing the effective contrast ratio of the display wall [15].

Our research is different from previous work in two ways. First, we use uncalibrated cameras to minimize the amount of human involvement and equipment required in the projector calibration process. Reliance on camera calibration implies that the cameras themselves must be fixed and cannot pan and zoom during measurement, for otherwise camera parameters will have to calibrated continously. Camera calibration also requires either human involvement and non-trivial equipment such as fine-plotted grid and regular 3D objects. Second, we devised an optical edge-blending technique to bypass the light-leakage problem with commodity projectors. Optical edge blending yields reasonably good results. We are experimenting digital tuning method using the feedbacks from uncalibrated cameras to further fine-tune the result of optical edge blending. This would allow us to achieve near optimal edge-blending for projectors that have the light energy leaking problem when their video signals are at the lowest level.

3 The Automatic Alignment Algorithm

The basic idea of our algorithm is to use the camera to "observe" geometric relations between adjacent projectors in the *alignment measurement* process. Relations thus obtained serve as constraints in the *alignment computation* process, which employs a multi-dimensional global minimization technique to deduce a good set of correction functions, one for each projector, that satisfies these constraints. A schematic of our alignment system is depicted in Figure 1.

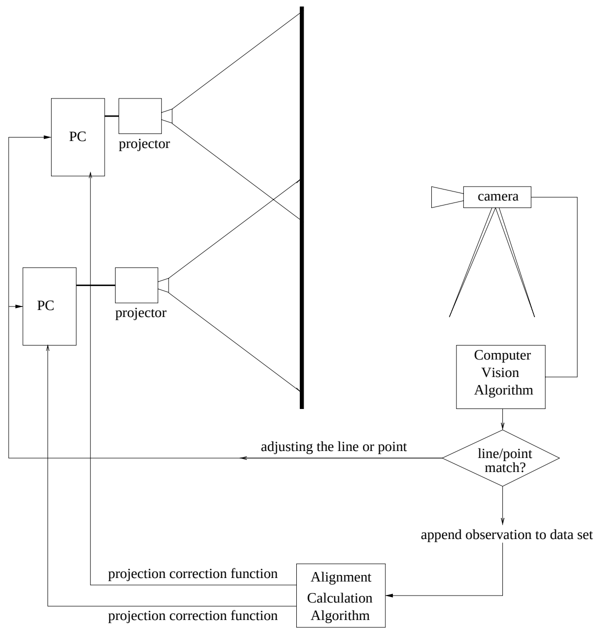

PC
projector
PC
projector
camera
Computer Vision Algorithm
line/point match?
adjusting the line or point
append observation to data set
projection correction function
projection correction function
Alignment Calculation Algorithm

Figure 1: Camera-based Alignment Data Collection

Our alignment algorithm uses two types of inter-projector relations: the **point match** and the **line match**. A point match simply states that a pixel (x, y) from one projector locates at the same spot on the display surface as another pixel (x', y') from an adjacent projector. Note here that we use the fact that adjacent projectors overlap by a small amount. The rationale for using point matches is simple: if there are a lot of point matches between any pair of adjacent projectors, the result of alignment computation will yield a set of projection functions that will maintain C^0 continuity across projectors. However, the point matches alone are not sufficient to constrain the system. This is particularly so when the projectors overlap only a small portion of their screens in a typical display wall system. The line match provides further shape constraints, especially C^1 continuity across the projectors. It basically means that a projected line from one projector is co-linear with another line from the neighboring projector.

The point and line matches act very much like the constraints in a rigid-body system. Line matching is like putting "struts" to keep components of a system from arbitrary deformation. We hope that by observing many such relations for a grid of projectors, there may just be just enough constraints so that we can figure out relative position and orientation for each projector.

3.1 Automatic measurement

Obtaining the line and point matches can be fully automated by using a camera controlled by a computer. Figure 2 contains a brief sketch of the measurement algorithm to obtain a point match between point P from projector A and a point Q from projector B.

```
1. display black on all projectors
2. pan the camera to roughly center its FOV on pixel P
3. display a cross centered at P on projector A
4. measure P's location L in the camera
5. take a guess of a pixel Q in projector B
6. loop
7.        display a cross center at Q on projector B
8.        measure Q's location L' in the camera
9.        if $\|L, L'\| > \epsilon$
10.          modify Q accordingly
11.       else
12.          return (P, Q)
```

Figure 2: pseudo-code for obtaining a point match

The line-matching algorithm works in similar fashion. As a simplification, we insert a point match so that the inner ends of the two line segments occupy the same spot on the display surface. To be more specific, the first line segment is displayed from pixel P_{1A} to pixel P_{2A} by projector A; the second line segment from P_{1B} to P_{2B} by projector B. The two line segments are said to be matched when (a) the have the same slope in the camera's field of view, and (b) (P_{2A}, P_{1B}) is a point match. The pseudo-code for obtaining a line match is sketched in Figure 3.

```
1. display black on all projectors
2. roughly center the camera's FOV on pixel $P_{2A}$
3. obtain a point match $(P_{2A}, P_{1B})$
4. measure line slope $k$ of $\overline{P_{1A}P_{2A}}$ in camera space
5. take a guess of a pixel $P_{2B}$ in projector B
6. loop
7.        measure line slope $k'$ of $\overline{P_{1B}P_{2B}}$
9.        if $\angle k, k' > \epsilon$
10.          modify $P_{2B}$ accordingly
11.       else
12.          return $(P_{1A}, P_{2A}, P_{1B}, P_{2B})$
```

Figure 3: pseudo-code for obtaining a line match

The benefit of relying only on line and point matches is that all the measurements taken are relative. Therefore the cameras need not be calibrated. Furthermore, when taking point matches, our algorithm can tolerate any kind of camera distortion, as long as the camera remains steady during a point match. For obtaining a line match, we only require that the the camera field of view be linear. This assumption can be easily met by using only the central area of a camera or, by centering the camera FOV on the adjoining ends of the line segment as we currently do. The latter approach can deal with a camera's radial distortion, because a straight line passing through the center of a camera FOV is not bent by radial distortion.

By using only relative measurements, we also circumvent the resolution limitation of an off-the-shelf camera. The problem lies in the fact that the resolution of a camera is much smaller than that of a high-resolution display wall. Simply using a single camera to capture the entire display surface at once won't yield enough measurement accuracy to guarantee pixel-level alignment. One can use multiple cameras, each observing a small portion of the display surface, or pan one camera across the display in great detail. Both approaches would require stitching together many pieces of local information to yield a globally consistent picture. Our algorithm can employ either approach. It has complete freedom to zoom the camera to obtain point and line matches with highest accuracy possible, and not worrying about change of camera parameters during the zoom and pan motions.

3.2 Alignment computation

We treat the problem of figuring out a set of projection functions as a global minimization problem with constraints derived from point and line matches. Note that we assume that projectors are already roughly aligned. It is straightforward to obtain an initial guess for each projector's position shifts, horizontal and vertical, based on the point match data. Based on our assumption of coarse alignment, the guess won't be too far from the actual projection functions. Therefore we already have a good starting point that is very close to the globally optimal solution for our problem.

3.2.1 A mathematical framework

As the first step, we formulate the projection from multiple projectors in mathematical terms. Projection can be thought of as a mapping between pixels in *projector space* (x, y) and the illuminated dots (u, v) on the global *display space*. This mapping, or the *projection function*, is normally accomplished using a lens system. [1] Figure 4 is a conceptual diagram of a typical lens system. The input signal from a computer or video source is converted to illuminated pixel on the imaging device, for example, LCD or DMD; light rays emanating through or reflected from the pixel travel through the lens system and converge on the illuminated dot on the screen.

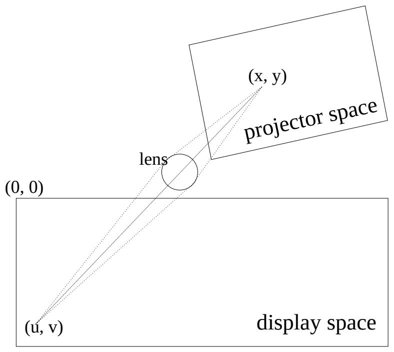


Figure 4: Conceptual diagram of a projection system

[1]The algorithm presented in this paper could in principle be applied to curved display surfaces as well. In that case, a 2D parametric coordinate system should be used for the display space.

A projection function can be generally decomposed into two parts, the projective transformation P and the non-linear distortion D [2]

$$(u, v) = (D_u(P_u(x, y), P_v(x, y)), D_v(P_u(x, y), P_v(x, y)))$$

Our current experiments deal with only the projective functions. But the method we employ should be extensible to deal with non-linear distortions as well.

The projective transformation can be expressed by a 3x3 matrix using 8 free parameters (m_{ij}), $i = 1..3$, $j = 1..3$ [2]:

$$\begin{aligned} P_u(x, y) &= \frac{m_{11} \cdot x + m_{12} \cdot y + m_{13}}{m_{31} \cdot x + m_{32} \cdot y + 1} \\ P_v(x, y) &= \frac{m_{21} \cdot x + m_{22} \cdot y + m_{23}}{m_{31} \cdot x + m_{32} \cdot y + 1} \end{aligned} \tag{1}$$

The radial distortion with respect to an optical center (c_x, c_y) and a distortion parameter ρ is given as

$$\begin{aligned} D_u(u, v) &= u + \rho \cdot (u - c_u) \cdot d^2 \\ D_v(u, v) &= v + \rho \cdot (v - c_v) \cdot d^2 \\ where, & \\ d &= \sqrt{(u - c_u)^2 + (v - c_v)^2} \\ (u, v) &= (P_u(x, y), P_v(x, y)) \\ (c_u, c_v) &= (P_u(c_x, c_y), P_v(c_x, c_y)) \end{aligned} \tag{2}$$

3.2.2 Match constraints

The constraints for the global minimization problems are produced as follows, assuming (P_u^i, P_v^i) is the projection function for projector i. Each point match between a pixel $p_1 = (x_1, y_1)$ from projector 1 and a pixel $p_2 = (x_2, y_2)$ from projector 2 results in a Euclidean distance error E^p on the display surface:

$$\begin{aligned} E^p(p_1, p_2) &= (u_1 - u_2)^2 + (v_1 - v_2)^2 \\ where\ (u_i, v_i) &= (P_u^i(x_i, y_i), P_v^i(x_i, y_i)) \end{aligned} \tag{3}$$

Each line match between two line segments $l_i = \overline{p_{11}, p_{12}}$ and $l_i = \overline{p_{21}, p_{22}}$ produces a point-match error and an error based on the angle between two line segments:

$$E^l(l_1, l_2) = \max(E^p(p_{12}, p_{21}), (\angle\overline{p_{11}p_{12}}, \overline{p_{21}p_{22}})^2)$$

Note that the error term $(\angle\overline{p_{11}p_{12}}, \overline{p_{21}p_{22}})^2)$ is also computed in the global display space. The goal is to minimize the maximum of the errors over all line and point matches:

$$E = \max(E^p_{max}, E^l_{max}) \tag{4}$$

3.2.3 Simulated annealing

We now have a well-defined multi-dimensional minimization problem. Global minimization over a large number of continuous variables has been studied for decades. Although scientists have not found a general, one-cure-all solution, several effective methods do exist. We chose **simulated annealing** [7, 11] as the minimization method to solve the alignment computation problem.

[2]This equation is only an approximation to an actual projection device. It ignores the several distortion effects such as color dispersion, while assuming that light rays coming from a single point on the image plane are focused on the same point on the screen.

Simulated Annealing is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems [10]. It mimics the manner in which metals recrystallize in the process of annealing: a melt, initially at high temperature and disordered, goes through a gradual cooling process. The system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at the temperature of zero. The initial temperature of the system must be reasonably high and the cooling process sufficiently gradual so that the system won't become quenched forming defects or freezing out in meta-stable states (i.e. trapped in a local minimum energy state).

The simulated annealing technique have seen successful uses in many scientific computations with hundreds and even thousands of continuous variables. Among several publicly available implementations, we chose the one provided by *Numerical Recipes in C* [11]. We wrote our own state evaluation function that calculates the energy for a given configuration of the projection functions. The energy is the error function that we just described in Equation 4. Figure 5 gives a sketch of the alignment computation using simulated annealing.

```
1. guess an initial set of projection functions
   based on the measurements
2. set initial temperature $T = T_0$
3. set total number of annealing steps at $N$
4. evaluate an energy $E$ for current configuration
5. $step = 0$
6. loop until $step = N$
7.       choose a random perturbation of
         the projection functions
8.       calculate the new energy $E'$
9.       $\Delta E = E' - E$
10.      accept the change with probability $e^{-\Delta E/kT}$
11.      $T = T_0 \cdot (1 - \frac{step}{N})^2$
12.      $step = step + 1$
13.done: return the projection functions
```

Figure 5: Alignment computation using simulated annealing

The simulated-annealing computation requires representing each projection functions using a vector of continuous variables. We discovered during our initial trials that an arbitrary projective transformation matrix may not correctly describe a realistic projection device. The reason is that a projective matrix allows shear deformation that an actual projector cannot produce. Therefore any attempt to calculate the projective matrices directly may yield projection mappings that do not correspond to reality. One way around this problem is to figure out a projector's extrinsic and intrinsic parameters which can then be used to derive the projective matrix. We model a projector with 9 parameters, X, Y, Z positions of the projector, its rotations along the three axises, its focal length, and its optical center (c_x, c_y). From these parameters, we can uniquely derive a projective function [2].

Given N projectors in the display wall system, alignment computation amounts of minimizing the error expression 4 over either $9N$ continuous variables (or $10N$ if radial distortion is also included). The total degree of freedom in this problem is quite reasonable. The results of automatic alignment will be presented in Section 6.2.1 and on Color Plate 1.

3.3 Computational re-alignment

Knowing the mapping function for a projector, we can apply it to "correct" the image displayed by that projector. This amounts to a re-sampling the image: given a projector's mapping function (P_u, P_v), and an image source $I_s(u, v) = (r, g, b)$, we obtain the intensity value I_p for a particular pixel (x, y) using the formula

$$I_p(x, y) = I_s(P_u(x, y), P_v(x, y)) \tag{5}$$

Care must be taken while sampling the source image. The image source is a discrete function, so is the projector. In practice, one pixel in the image source is rarely mapped exactly onto a pixel in the projector. If we simply pick one pixel in the image source for each projector's pixel, noticeable artifacts of a certain kind of "aliasing" will appear [6]. This is a well-known phenomenon that occurs re-sampling a discretized signal. There is, however, a simple and effective solution for *anti-aliasing* an image. A convolution kernel can be used to "smooth" out the blockiness by averaging neighboring pixels. In general, the bigger the convolution kernel, the better the result. In practice, a 3x3 kernel, or a *bi-cubic* sampling, is often sufficient. Sampling can be performed by the CPU. There also exist efficient ways to sample an image using a projective transform. Raskar et al described a method using the texture-mapping hardware found on many graphics accelerators [13, 12].

3.4 Discussion

Our algorithm offers several advantages over other algorithms reported recently. First, we obtain misalignment information using an **uncalibrated** camera that observe the point and line patterns on the projectors. No human involvement is required to take misalignment measurement other than placing the camera(s) in front of the display wall. Second, since all the measurements that we take are local and relative measurements, accurate measurements can be easily obtained by zooming in the camera close to the target. This allows us to always use the center field in the camera for measurements, obviating the need to correct for the camera's non-linear distortion. Our algorithm is free to do this because it does not depend on camera calibration. We are thus able to use a low-resolution camera to calibrate a very high-resolution display wall.

The drawback of our approach is that we rely on a global minimization technique that does not guarantee to give a satisfactory answer. Although this situation has not occurred in our experiments, we will look for a deterministic and more efficient method to calculate the projection functions.

4 The Color Balancing Algorithm

The color characteristics of a projector depends on many factors, the most significant of which is the color temperature of the lamp. As the lamp ages, its energy distribution over the spectrum slowly changes, and it is very difficult to bring the lamp temperatures among many projectors into agreement. Thus, w1hen we display a single image tiled across many projectors, the colors of different regions of the image don't match. To address this problem, we perform *digital color correction* on the image source before it is sent to the projectors for display.

We can break down the color balance problem into two sub-problems: color uniformity and color conformity. Color uniformity means that all projectors have nearly the same color characteristics. Color conformity means that the projector's color spectrum conforms to a standard color spectrum such as specified by the international standards $CIE-XYZ$. In this paper, we tackle the first sub-problem, color uniformity, because lack of color uniformity among the projectors has the most significant visual impact on the quality of the image.

Our strategy is to map an (r, g, b) triple in an image to another triple (r', g', b'), for each projector. In general, given 8 bits for each color channel, this requires mapping each of 2^{24} color values into another 24-bit value. However, we decouple the problem into the separate color channels, so we treat each channel as a one-dimensional (luminance) matching problem.

4.1 Luminance measurement

We use a single camera to measure the luminance curve on each projector. We fix the camera's shutter speed and exposure time so that the same luminance on different projectors induces the same reading in the camera. Figure 6 depicts the measurement process to obtain the luminance curve for each projector, at a set of samples. Next we apply quadratic interpolation to obtain the luminance response curve for the entire input range [0..255] while smoothing out sampling noise.

```
1.  display black on the projector
2.  focus camera FOV on the projector center
3.  for each pure color channel (R, G, B) :
4.      input $x = x_0$
5.      while $x <= 255$ :
6.              display a pure color block x
7.              measure luminance $y(x)$
8.              $x = x + delta$
10. return the observation data
```

Figure 6: pseudo-code for obtaining RGB luminance curves

4.2 Luminance matching

Given the measured luminance curves for a single color channel on all projectors, we can find a "common" curve that is within the luminance range on all projectors. More specifically, given luminance curves $y = L^i(x), i = 1..N$ for the N projectors, we calculate the common curve LL as follows:

$$\begin{aligned} LL(x) &= \frac{x}{255} \cdot \min(L^i(x), i = 1..N) \\ &\quad + \frac{255 - x}{255} \cdot \max(L^i(x), i = 1..N) \end{aligned} \tag{6}$$

The rationale behind the changing weight between the minimum and maximum luminance curve is that at the lowest luminance end, it is possible to adjust all projectors to match the brightest projector, and at the highest end, adjust all projectors to match the dimmest projector.

After deriving a common luminance, we establish a correction function for each color channel on each projector. The correction function is essentially a lookup table T with 256 entries:

$$\begin{aligned} T_r(x) &= L_r^{-1}(LL_r(x)) \\ T_g(x) &= L_g^{-1}(LL_g(x)) \\ T_b(x) &= L_b^{-1}(LL_b(x)) \end{aligned} \tag{7}$$

For a given (r, g, b) triple, we will get the same color on all projectors if they each display instead the translated triple $(T_r(r), T_r(g), T_r(b))$.

Digital color correction involves substituting RGB triples in an image with their corrected values from the lookup table. This lookup can be done efficiently if the projector hardware or graphics card supports a separate lookup table (or gamma correction table) for each color channel. We will show the results of our automatic color-balancing algorithm in Section 6.2.2.

5 Optical Edge Blending

In the regions where two (or four) projectors overlap, the screen emits light accumulated from both (all) projectors. Thus, even if the projectors were perfectly aligned, the image is too bright in these regions. A simple strategy for addressing this problem is to use *edge blending*, wherein the contribution of each projector gradually falls off across the overlap region such that their *total* light contribution is approximately correct for the given image. For example in Figure 7 the region AB would be bright if each projector contributed the full luminance; thus, our goal is for projector 1 to have full contribution at point B and zero contribution at point A (and *vice-versa* for projector 2) with smooth transitions in-between. The combined luminance resulting from edge-blending is a smooth curve that adjoins the two projectors together, even when they have slightly different luminance level.

The traditional method for edge-blending projection devices is to alter the video signal or the image intensity according to the blending curve. Unfortunately, it does not work for commodity LCD projectors, which have a fair amount of light leakage. When the projector is supposed to display RGB (0, 0, 0), some amount of light is leaked through the LCD chip to the screen, so that the overlapped region between two projectors is twice as "bright" as the non-overlapped region. It is not possible for the traditional edge-blending technique to remove this gray band, because the signal for RGB (0, 0, 0) cannot be reduced any further. This problem exists not only for a pitch black image, but also for RGB inputs in the low-luminance range.

We developed the optical edge blending that operates directly on the light output from the projectors, hence bypassing the light leakage problem. The method is based on aperture modulation. It is well-known that placing an opaque object between the lens and the screen can create a gradient shadow. Figure 7 illustrates the idea of optical-blending two projectors. A blend frame is placed in the free space between the lens and the screen. For a point A on the screen, all the light from the exit pupil of the projector 1 is blocked by the blend frame. Hence the luminance contribution from projector 1 is zero. At the point B, all the light from projector 1 passes unhindered to the screen.

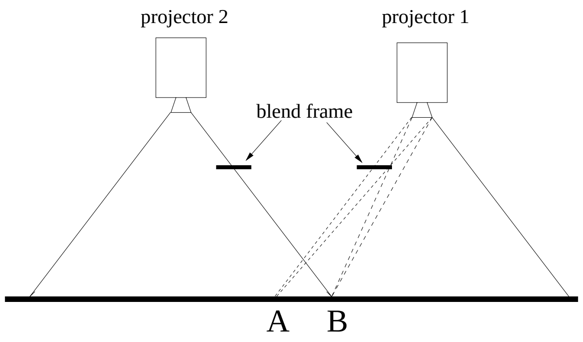


Figure 7: The conceptual diagram of optical blending

For points between A and B, the percentage of light from projector 1 that gets focused on the screen is a gradual curve from 0 to 100 %. Hence the name "aperture modulation." Assume $I(x,y)$ is the luminance distribution on the exit pupil for a particular pixel, the general equation for the blending curve is

$$\frac{\int\int_{visible\ pupil} I(x,y)dxdy}{\int\int_{entire\ pupil} I(x,y)dxdy} \tag{8}$$

This function can be calculated if the distribution $I(x,y)$ is known, or can be approximated by a constant or linear function. In reality, the light distribution on the exit pupil varies from pixel to pixel. The key point from Equation 8 is that the blending function is a smooth function. Furthermore, one can easily prove, by means of Euclidean geometry, that for image points aligned on the same vertical (or horizontal) line in the projector's imaging space, the unobstructed aperture is the same. This results in the same blending factor for object points that lie on a vertical (or horizontal) line.

6 Implementation and Results

We implemented our automatic alignment algorithm for a 2x4 multi-projector display wall. The following is a brief description of the hardware and software components in our setup.

6.1 Platform description

The display hardware includes Proxima 9200 LCD portable projectors in a 2x4 matrix. Each projector is capable of displaying true 1024x768 resolution. Adjacent projectors overlap between 40 and 70 pixels. The total resolution of the entire display surface is roughly 3800 pixels wide and 1500 pixels tall, spanning over an 18' by 8' black screen made by Jenmar, inc. Each projector is attached on a coarse-grain mechanical tables. The tables offer full pose adjustment with 6 degrees of freedom. They are normally used for time-consuming manual alignment of the projectors. But in our experiments, they provide us with an easy means to "mis-align" the projectors with arbitrary rotations and translations. Two projectors in a vertical column are placed in an off-the-shelf rack that costs roughly $100.

Each projector is driven by a 450 MHz Pentium-II PC with an Intergraph graphics accelerator. The PCs are interconnected by two networks: 100 Mb fast ethernet and Myrinet. User-level Virtual-Memory-Mapped Communication over the Myrinet system-area network [4, 3] offers very high-bandwidth and low-latency communication among the PCs.

In front of the display wall, we set up a Canon VC-3 video conferencing camera to measure the alignment patterns. The VC-3 camera has motorized pan and tilt, and motorized zoom and focus, all can be controlled through the serial port. The camera is connected to a stand-alone PC via the serial port for motion control. This PC also houses an Integral Video Grabber card which digitizes each video frame from the VC-3 camera into an array of RGB values in the PC's physical memory.

6.2 Emperical results

As the basis for comparison, the first color plate shows a picture of the displaywall without alignment and color balance.

6.2.1 Computational alignment

	number of annealing steps				
	1000	2000	5000	10000	50000
error (pixels)	1.32	1.24	1.15	1.23	1.09
time (mins)	1.2	2.4	4.0	8.7	43

Table 1: alignment computation time and accuracy (in pixels and minutes)

We ran the alignment data collection algorithm for all 8 projectors. There are total 10 overlapped region. For each overlapped region, we take 15 point match samples and 6 line match samples, resulting in a total of 150 point samples and 60 line samples. The data collection phase takes roughly 33 minutes. A lot of the time is spent in panning and tilting the camera to zoom onto a spot. The data collection time can be reduced further by employing multiple cameras, each responsible for a subarea of the display wall. Since our alignment algorithm only collects local and relative alignment information, there is no need to correlate data collected from different cameras. In other words, the data collection phase can be trivially parallelized by using many inexpensive cameras.

Obviously the accuracy of alignment computation depends critically on the accuracy of the point and line matches. Our alignment algorithm can in principle allow arbitrarily accurate data gathering, simply because it uses only relative geometric relationships among adjacent projectors. We can place or zoom a camera very close to the display surface to increase the measurement definition. In our experiments, we placed the Canon VC-3 camera at a distance of 10 feet to the DisplayWall. At the camera's highest zoom range, the image processing routine can easily distinguish two adjacent pixels from a projector. Our current point- and line-matching algorithm uses nearest-neighbor fit. This means that the measurement error is at worst half a pixel.

We then ran the simulated annealing algorithm on the alignment data to derive the projection functions. Table 6.2.1 shows the time and accuracy in the alignment computation, as the total number of annealing steps increases.

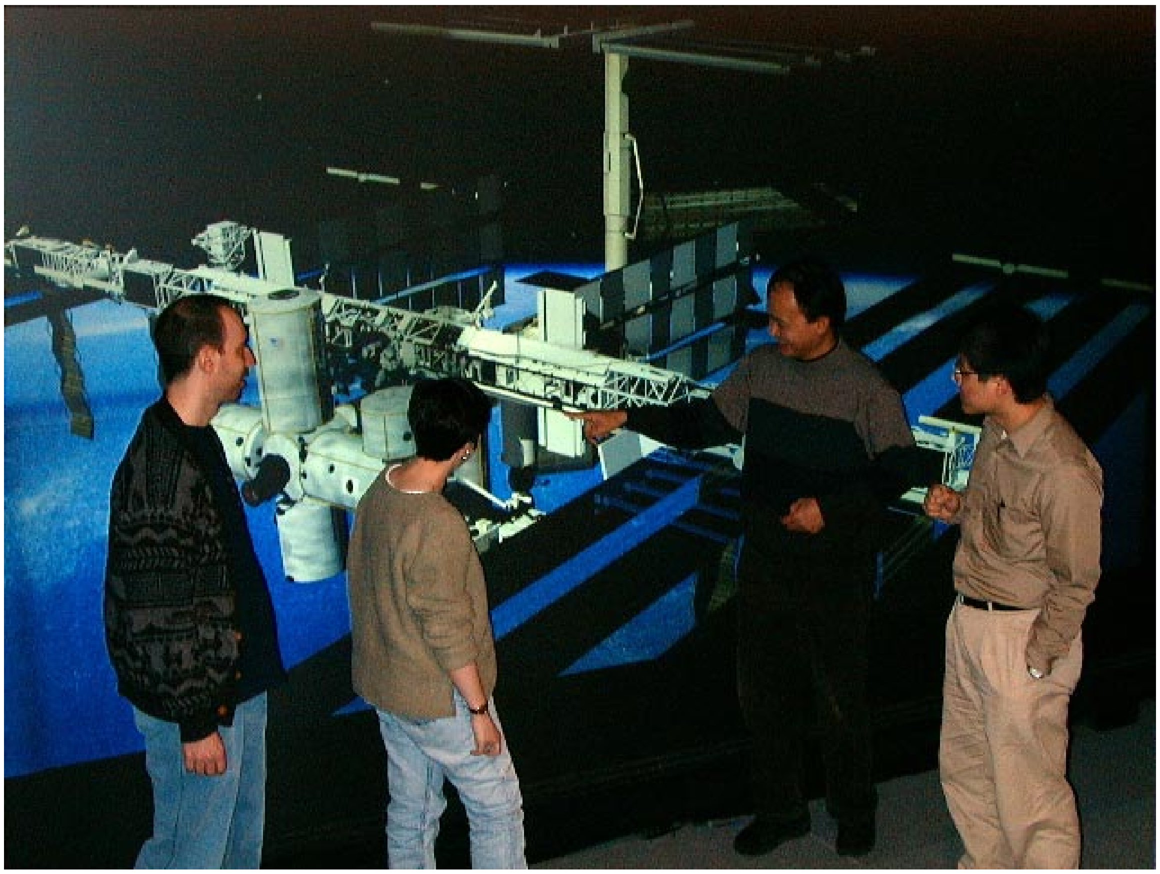


Figure 8: Optical blending

The accuracy is expressed in terms of maximum error between two points in a point match and the angle between two lines in a line match (in degrees). The pixel-grain error can be attributed to the error in the measurement. Due to the nearest-neighbor fit in the point match, the worse measurement error is half a pixel. In theory, this means our final alignment computation should yield an error of that magnitude. We are still investigating this problem. The actual alignment result, shown in Figure 9 is quite good.

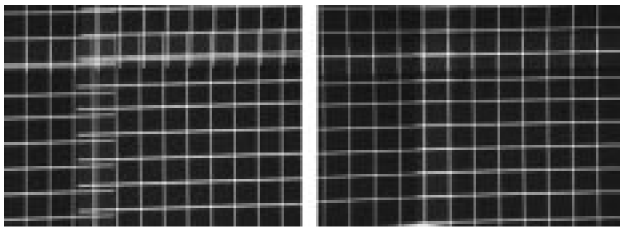


Figure 9: Aligning a grid: before and after pictures

The alignment computation time is spent mostly on computing the error functions in the simulated annealing algorithm. The time spent on each evaluation is proportional to the amount of observation data. The total computation time also depends on the number of steps in the annealing process. The more steps taken, the more gradual the annealing process is and usually the better the alignment result. In practice, 5000 steps or 10000 steps are generally sufficient.

Color plate 1 shows the before and after pictures for computationally aligning a map using 5000 steps in the annealing process.

6.2.2 Color balancing and optical blending

We used a single camera to collect color luminance data for each projector. Within the RGB range 0..256, we sample at the increment of 5. It took roughly 14 minutes to collect the data for all 8 projectors. The result of color balancing and optical blending a map is shown in Color plate 2. Figure 8 shows our smoothly blended display wall with the optical blending technique.

7 Conclusions

In this paper, we described our methods for aligning and color-balancing a multi-projector display using computational and optical means. Our algorithms use an uncalibrated camera to obtain misalignment and color imbalance information. A computation process uses the data collected to figure out good correction functions to counter the effect of physical misalignment and color imbalance. These corrections are then applied to images themselves before they are sent to the projectors. We implemented the automatic alignment and color balancing algorithms for our 8-projector display wall, with good results. We also devised an optical edge blending method that can create smooth luminance transitions between adjacent projectors. Unlike traditional edge-blending technique, our optical method deals effectively with the light leakage problem typically found on commodity projectors.

Our alignment algorithm relies only relative measurement that requires no camera calibration. It solves the inherent tension between the relatively low resolution of a camera and the very high resolution of a scalable display wall, by allowing the camera to freely zoom in on a measurement target. The result is a highly accurate computational alignment among projectors. In addition, since no camera calibration is required, our algorithm can be fully automated. Although our algorithm currently ignores the radial distortion on a projector, the global minimization technique can be applied to solving for radial distortion parameters. We are currently do such experiments. There is also a lot of room left to improve speed of alignment data collection and alignment computation, as we haven't tried to optimize these processes.

Our color-balancing algorithm balances multiple projector's color temperature using an inexpensive camera. It is however, still at the preliminary stage. One of the problem is the relative low accuracy of luminance measurement on a commodity camera. We are currently working on an iterative algorithm that will make this problem less significant in color balancing.

Although aperture modulation is a well-known technique in the photography community, using it to edge-blend multi-projector displays is new. Our experiments show that optical edge blending works very well with commodity projectors. We also developed a digital tuning technique to fine-tune the results of optical edge blending.

Besides improving the accuracy of our algorithms, we also plan to integrate precise alignment and color balancing into the graphics pipelines of existing displaywall applications such as OpenGL, Virtual Display, High-Definition MPEG videos, and Multi-media playback [1].

A Proof of the 3x3 perspective representation

An ideal lens system is what is known as a pin-hole lens. In such a lens system, the principle ray from the pixel source (x, y) travels straight through a unique optical center, or the pin hole, to its image point (u, v) on the screen.

We establish a global 3-dimensional coordinate system (u, v, w) with its U-V plane co-incide with the display surface. The W coordinate for a point on the display surface is zero. A pixel (x, y) in the projector space can be expressed in the global coordinate system as:

$$p \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = M' \cdot \begin{bmatrix} x - c_x \\ y - c_y \\ 0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z - f \end{bmatrix} \tag{9}$$

$$where, M' = R_x \cdot R_y \cdot R_z \cdot S_x \cdot S_y \tag{10}$$

R_x, R_y, and R_z are standard rotation matricies. S_x and S_y are standard scaling matrices that convert the pixel units on the image plane into proper units in the global coordinate system $SPACE_s$. We have separate scaling factors for X and Y axises because the pixels on the image plane need not be perfect squares. $[t_x, t_y, t_z]^T$ is the position of the optical center in the global coordinate system. (c_x, c_y) is the center of the projector space. It may not be the center of the LCD or DMD chip if the projector provides key-stone correction.

According to the definition of an ideal lens system, the three points, (x', y', z'), its image on the screen (u, v, w), and the optical center (x_o, y_o, z_o) are co-linear and hence satisfy the following parametric line equation

$$\begin{aligned} u &= x_o + (x' - x_o) \cdot \lambda \\ v &= y_o + (y' - y_o) \cdot \lambda \\ w &= z_o + (z' - z_o) \cdot \lambda \end{aligned}$$

Setting w to 0, we obtain an expression for λ in terms of z_o and z'. Substituting λ with this expression in the first two equations gives us

$$\begin{aligned} u &= \frac{x_o \cdot z' - x \cdot z_o}{z' - z_o} \\ v &= \frac{y_o \cdot z' - y \cdot z_o}{z' - z_o} \end{aligned} \tag{11}$$

According Equation 9, x', y', and z' are each represented by an expression linear in x and y. Substituting x', y', and z' result in something like

$$\begin{aligned} u &= \frac{m_{11} \cdot x + m_{12} \cdot y + m_{13}}{m_{31} \cdot x + m_{32} \cdot y + m_{33}} \\ v &= \frac{m_{21} \cdot x + m_{22} \cdot y + m_{23}}{m_{31} \cdot x + m_{32} \cdot y + m_{33}} \end{aligned} \tag{12}$$

One can also show that m_{33} is non-zero, because otherwise a solution does not exist for pixel (0, 0). We can normalize the matrix by dividing it with m_{33} without affecting the values of u and v. This gives the projective transformation seen Section 3. Note that our derivation takes into consideration an image-generation plane that may tilt at an arbitrary angle to the optical axis in the projector. Perfect alignment of the image-generation with the optical axis is not possible due both to manufacturing difficulties and to the zoom motion on some lenses.

B Perspective matrix calculation from basic parameters

The pose parameters are rotations, $(r_x,\ r_y,\ r_z)$, of the projector around its optical center, translations of the optical center, $(t_x,\ t_y,\ t_z)$, in the global coordinate system, the focal length f, and the center of the projector space $(c_x,\ c_y)$. Given these parameters, one can write down a projective matrix as follows:

$$\begin{aligned} m'_{11} &= t_x \cdot r_{31} - t_z \cdot r_{11}\ , \quad m'_{12} = t_x \cdot r_{32} - t_z \cdot r_{12} \\ m'_{13} &= t_x \cdot r_{33} - t_z \cdot r_{13}) \cdot f - \ (t_x \cdot r_{31} - t_z \cdot r_{11}) \cdot c_x \\ &\quad - (t_x \cdot r_{32} - t_z \cdot r_{12}) \cdot c_y \\ m'_{21} &= (t_y \cdot r_{31} - t_z \cdot r_{21}\ , \quad m'_{22} = t_y \cdot r_{32} - t_z \cdot r_{22} \\ m'_{23} &= (t_y \cdot r_{33} - t_z \cdot r_{23}) \cdot f - \ (t_y \cdot r_{31} - t_z \cdot r_{21}) \cdot c_x \\ &\quad - (t_y \cdot r_{32} - t_z \cdot r_{22}) \cdot c_y \\ m'_{31} &= r_{31}\ , \quad m'_{32} = r_{32} \\ m'_{33} &= r_{33} \cdot f - r_{31} \cdot c_x - r_{32} \cdot c_y \end{aligned}$$

$$m_{ij} = m'_{ij} / m'_{33} \tag{13}$$

Where, r_{ij} is the element of a 3x3 rotational matrix R that represents rotations r_x, r_y, and r_z.

C Proof of the manual method

The following equations illustrate the calculations taken to figure out the matrix. Let us assume that $(x_i, y_i), i = 1..4$ represents the four measured locations for four pixels $(p_x^i, p_y^i, i = 1..4)$. The unknown matrix is M,

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix}$$
$$u = x'/z', \quad v = y'/z'$$

Using equation [?], we establish 8 equations:

$$x_i \cdot (m_{31} \cdot p_x^i + m_{32} \cdot p_y^i + 1) = m_{11} \cdot p_x^i + m_{12} \cdot p_y^i + m_{13}$$
$$y_i \cdot (m_{31} \cdot p_x^i + m_{32} \cdot p_y^i + 1) = m_{21} \cdot p_x^i + m_{22} \cdot p_y^i + m_{23}$$
$$i = 1..4$$

This system of equations is linear in m_{ij}, and one can further prove that it has a unique non-trivial solution. We thus obtain M.

References

[1] Han Chen, Yuqun Chen, Douglas W. Clark, Perry Cook, Stefanos Damianakis, Georg Essl, Adam Finkelstein, Thomas Funkhouser, Allison Klein, Kai Li, Zhiyan Liu, Emil Praun, Rudrajit Samanta, Ben Shedd, Jaswinder Pal Singh, George Tzanetakis, and Jiannan Zheng. Early experiences and challenges in building and using a scalable display wall system. *Computer Graphics and Applications*, 220:671–680, 2000.

[2] Yuqun Chen, Douglas Clark, Adam Finkelstein, and Kai Li. A method to align high-resolution multi-projector displays using an uncalibrated camera. Technical report, Princeton Unversity, Computer Science Department, March 2000.

[3] Yuqun Chen, Stefanos N. Damianakis, Sanjeev Kumar, Xiang Yu, and Kai Li. Porting a user-level communication architecture to nt: Experience and performance. In *3rd Usenix Windows NT Symposium*, July 1999.

[4] Yuqun Chen, Czarek Dubnicki, Stefanos N. Damianakis, Angelos Bilas, and Kai Li. Utlb: A mechanism for translations on network interface. In *The 8th International Conference on Architectural Support for Porgramming Languages and Operating Systems (ASPLOS-VIII)*, October 1998.

[5] Raskar et al. Multi-projector displays using camera-based registration. In *IEEE Visualization '99*, 10 1999.

[6] J.D. Foley, A. van Dam, S. K. Feiner, and J.F. Hughs. *Simulated Annealing Methods*, chapter 0, pages ???–??? Addison-Wesley, 2 edition, 1996.

[7] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[8] Aditi Majumder, Herman Towles, and Greg Welch. Color matching of projectors for multi-projector displays. In *submitted for publication*, 2000.

[9] Theo Mayer. New options and considerations for creating enhanced viewing experiences. In *Computer Graphics*, pages 32–34, May 1997.

[10] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. *Journal of Chemical Physics*, 21:1087–1092, 1953.

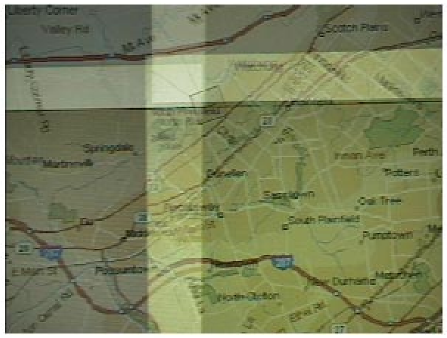
[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Simulated Annealing Methods*, chapter 10, pages 444–455. Cambridge University Press, 2 edition, 1992.

[12] R. Raskar. Immersive planar display using roughly aligned projectors. In *IEEE Virtual Reality 2000*, March 2000.

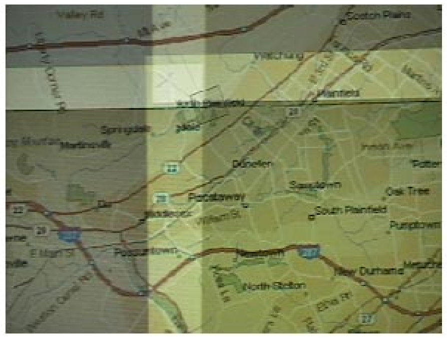
[13] R. Raskar, M. Cutts, G. Welch, and W. Stuerzlinger. Efficient image generation for multiprojector or multisurface displays. In *9th EuroGraphics Rendering Workshop*, 1998.

[14] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: A unified approach to image-based modeling. In *SIGGRAPH 98*, pages 179–188, July 1998.

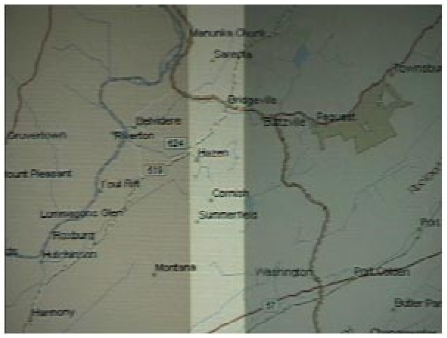
[15] Rajeev Surati. *A Scalable Self-Calibrating Technology for Large Scale Displays.* PhD thesis, MIT, 1999.



physically mis-aligned

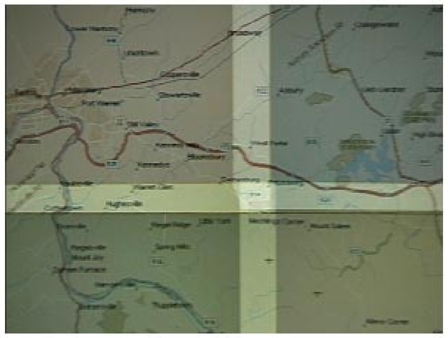


computational aligned

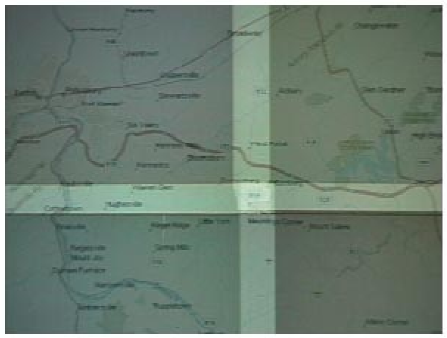


computational aligned

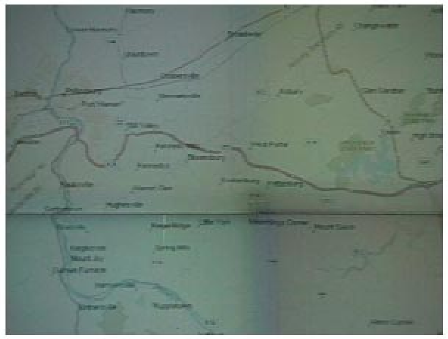
Color Plate 1: alignment results



unbalanced color temperatures



color-balanced image



color balance + optical edge blending

Color Plate 2: color balance and optical blending

Figure 10: Color Plates