A Low-Cost Projector Mosaic with Fast Registration

Ramesh Raskar Jeroen van Baar Jin Xiang Chai

Mitsubishi Electric Research Laboratories, Cambridge MA, USA

{raskar, jeroen, chai}@merl.com

Abstract

We describe a calibration and rendering technique for a projector array that can render a seamless rectangular image on a planar surface. The projectors utilize an attached camera to automatically compute the relative pose among them. We describe calibration of the full system using planar display surface to achieve registration and intensity blending of overlapping images. We present an efficient rendering method to pre-warp images so that they appear correctly on the screen, and show experimental results.

1. Introduction

A photo-mosaic is a collection of images registered together to form one large image. The images are either taken from the same viewpoint in different directions or taken from different viewpoints of a planar scene [Szeliski97][Chen95]. Can we create a similar *projector mosaic* on a display surface by seamlessly merging output of overlapping projectors? For photo-mosaic, the images are captured by casually orienting and positioning the camera. Can we similarly create a large display using casually aligned projectors? Currently, large displays are generated by tiling together a two dimensional array of projectors by precisely aligning them. In most systems, the projected images are abutting [Panoram]. Some systems allow partial overlap [Trimensions] [Chen00][SMural], but special care is still taken to ensure each projector generates precise rectangular image aligned with world horizontal and vertical, so that the overlap itself is a well-defined rectangle. Due to such design constraints, the installation and operation of such systems is extremely expensive.

A projector is a dual of a camera, and the image projection process can be expressed using the well-known pinhole camera model. Thus far, however, projectors have received little attention in the computer vision community. Similar to photo-mosaics, where sophisticated registration techniques in software simplify the constraints on the process of capturing, we can exploit the geometric relationship between projectors to simplify the task of positioning the projectors. Rare examples of such an approach are described in [Raskar99][Surati00]. However, for registration, both those systems attempt to create a per-pixel warping function and do not exploit the well-known geometric relationships between images of a planar scene. Specifically, in case of projectors, the projected images are related by a simple planar homography if the display surface is a 3D plane. In this paper, we present a set of techniques to compute the pixel mapping between uncalibrated projectors by exploiting planar homographies. We then demonstrate a method for correctly displaying rectangular images on a planar surface even under oblique projection.

1.1. Motivation

Large projectors arrays are popular because they allow a practical solution to generate high-resolution and bright images by tiling together a set of projectors. Older systems, such as video walls [Panoram], use a two dimensional matrix of rear-projectors with abutting image edges separated by a small but visible gap. Newer systems [Trimensions][ComView] use overlapping projectors with facility for precise manual electro-mechanical adjustment for image registration and cross-fading. The setting up of these displays is still quite tedious, requiring precise projector overlap, often needing orthogonal projection to the screen. This arguably is the most prominent drawback of large format display design. Research into techniques for automating this registration process is helping alleviate this time consuming setup.

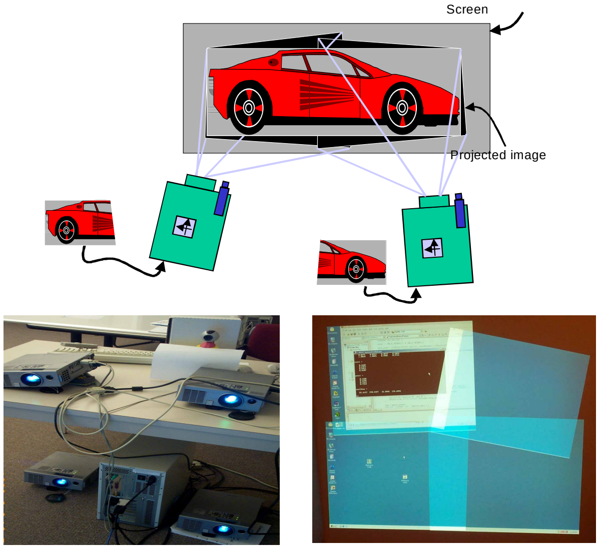


Figure 1. A casually installed projector array. (Top) a schematic diagram (bottom) our four projectors prototype, with viewing camera on the top and resulting projection.

Our goal is to significantly reduce the support and infrastructure cost by providing a flexible calibration and rendering technique that can *adapt* to a given projector array configuration. Further, our goal is to reduce the time required for setup and registration to a few seconds. The calibration technique is designed to be very simple to use and free of any human interactions. A novelty of the system here is that we calibrate using blank planes onto which calibration patterns are projected.

1.2. Previous Work

The authors in [Raskar99][Surati99][Chen00] have described various approaches to build a multi-projector display. [Raskar99] provide a general solution to the seamless display problem. They use a series of calibrated stereo cameras to determine the display surface and individual projector's intrinsic and extrinsic parameters all in a common coordinate frame. The result is an exhaustive description of the entire display environment. Although this approach allowed for a general solution, the computational effort and resources needed to implement this

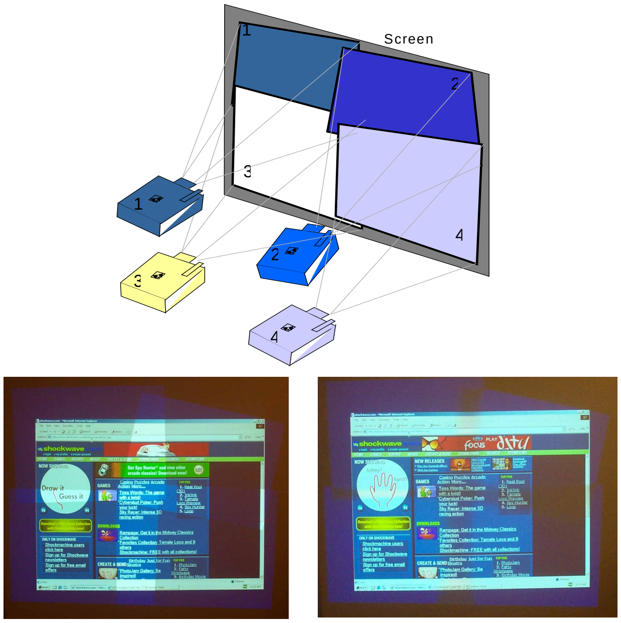


Figure 2. Keystoned projector generates a quadrilateral on the display plane. A pre-warped image creates a correct view inside the inscribed rectangle without intensity blending (bottom left) and with intensity blending (bottom right)

approach introduce their own level of complexity. [Chen00] provide a mechanism to help reduce mechanical alignment using a camera with controllable zoom and focus, mounted on a pan-tilt unit. The data collection and computation take over 30 minutes. [Surati99] presented a solution that also used a camera to establish the relative geometry of multiple projectors. Using a camera that had been calibrated by looking at a regularly spaced grid (printed on a physical paper) placed in front of the display surface, subsequent projector imagery can be registered to the grid. Thus, a user must use a physical large calibration pattern. Although our approach produces results of better quality, the main difference is the use of a very low cost camera and blank planar screen as calibration pattern. The major advantage of our method is that it allows fast registration (few seconds compared to several minutes) without any human interactions.

1.3. Projector Mosaic

The purpose of our self-correcting array of projectors is to generate a rectangular image of known aspect ratio, even when the individual projectors are aimed at an arbitrarily inclined planar surface. We created the simple prototype in Figure 1 by taking four standard commercial projectors and rigidly mounting a low-cost camera viewing the projections. The system is calibrated with the technique described in Section 3. The rendering technique, is described in Section 4, and involves pre-warping of images using the homography between each projector and display surface, so that the complete projected image is correct.

2. Methods

Rather than carry out a full calibration for the projector-camera system, such as computing the intrinsic parameters for both devices plus their relative pose, our projector mosaic system hinges only on the homography between the single static camera and multiple source projectors. To obtain a homography, each projector displays a checkerboard calibration pattern on a blank planar surface, and four or more point (or line) correspondences are automatically built and then used to compute the homography between the projector image and the camera image. This step repeats for each camera-projector pair relative to the same physical plane. The outline of our algorithm is described as follows:

```
During Pre-processing
For each projector
        –   Project structured pattern
        –   Extract the features from pattern and find the camera-
            projector pixel correspondences
        –   Compute homography between cam-proj
Find usable rectangular display area in the union of projections
        –   Compute homography between proj and display area
        –   Compute intensity weights for feathering
During Rendering
For each projector
        –   Warp input image using homography
        –   Modify intensities using computed weights
```

The preprocessing stage is completely automatic, allowing a quick and easy operation after installation. The rendering exploits using 3D graphics hardware for real-time execution.

2.1. Configuration and assumptions

First, we review how a pair of homographies can be used to define the relationship between image coordinates. Given two cameras, viewing points on a 3D plane Π, the point positions in the two images are related by a 3x3 homography matrix H, defined up to scale. If m_1 and m_2 are projections of a 3D point M which belongs to Π, then

$$m_2 \sim= H\, m_1 \tag{2.1}$$

where m_1 and m_2 are homogeneous coordinates and $\sim=$ means equality up to scale.

In our projector mosaic system, we aim to seamlessly stitch the N source images from multiple projectors P_i, i =1,…,N. In order to align projector with projector each other, we use one single

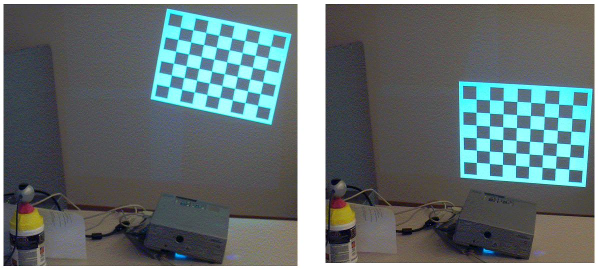


Figure 3. Calibration checkerboard projected from different projectors, one at a time, is viewed by the camera to compute homography with respect to the projector.

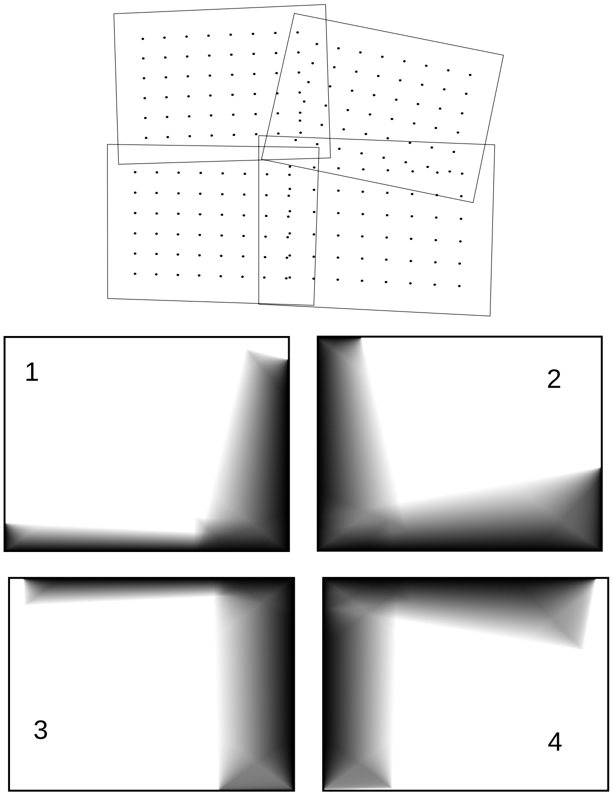


Figure 4. Intensity blending (Top) Projection of illuminated quadrilaterals as recorded in camera coordinates system. The dots indicate points on checkerboard detected for homography computation. (Bottom) The intensity weight for the four projectors. The gray scale indicates the weight in [0,1]. The weights taper off to zero in the overlap region near the projector framebuffer boundary.

camera C to record all the projector images. The projector to camera mapping as well as relative projector to projector mapping are then described by the above plane homographies due to the planar display surface used.

For notation, we choose homogeneous coordinates for both 2D camera coordinates $\mathbf{x}_c=(x, y, 1)^T$ and for 2D projector coordinates $\mathbf{u}_i=(u, v, 1)^T$, i = 1,..,N, from multiple source projectors. In this context, the input consists of N projector images captured by the single camera C with the known homography matrix, H_{c1}, H_{c2},…, H_{cN}, satisfying

$$\mathbf{u}_i \sim= H_{Ci}\,\mathbf{x}_C \qquad i=1,\dots,N \qquad (2.2)$$

In the final display, the projector mosaic is cropped to a rectangle with a known aspect ratio, say 4:3 (see Figure 2). The display coordinates inside this rectangle, which is equivalent to the coordinates of the input image, say a desktop to be displayed, are denoted as $\mathbf{x}_r=(x, y, 1)^T$. The relationship between the display coordinates and camera coordinates can be described by another 2D projective matrix H_{rc}. Obviously, we have

$$\mathbf{u}_i \sim= H_{Ci}\,\mathbf{x}_C \sim= (H_{Ci}\,H_{rc})\,\mathbf{x}_r \quad i=1,\dots,N \qquad (2.3)$$

For the simplicity of illustration, we define a set of new 3x3 matrices: $H_{ri} = H_{ci}H_{rc}$, i=1,…,N, which specify the geometrical relationship between the individual projector and final display coordinate directly. Equation 2.3 enables us to conveniently determine the pixel mapping between two arbitrary projectors as follows

$$\mathbf{u}_j \sim= H_{rj}\,H_{ri}^{-1}\,\mathbf{u}_i \qquad (2.4)$$

where $\mathbf{u}_i$ and $\mathbf{u}_j$ denote the corresponding pixels in projector P_i and P_j, respectively.

3. Calibration

To generate seamless projector mosaic, we must ensure, first, that the projected images are registered with the camera and, second, that the intensities across the overlap region appear to transition smoothly.

3.1. Registration

In this subsection, we describe the technique to compute relationship between projector pixels. The same relationship will be implicitly used for rendering to ensure that the projected images are registered. For projector mosaic, we do not need explicitly calibrate either camera or projectors in order to stitch the projector images seamlessly. Instead, the homographies computed with a static observing camera are sufficient.

We first project the checkerboard pattern from each source projector sequentially and then record the projected image on the display surface by the single camera (see Figure 3). By extracting the feature points from the 2D camera images corresponding to known 2D points from the projector pattern, we can determine the 3x3 homography between the static camera and each projector based on Equation (2.2).

More formally, the homography estimation between camera and projector can be formulated as maximum likelihood estimation problem. Given n corresponding feature points between the camera image $\mathbf{x}$ and projector image $\mathbf{u}$, the maximum likelihood estimate can be obtained by minimizing the following function:

$$\sum_{i=1}^{n} \| \mathbf{u}_j - H_{3x3}\,\mathbf{x}_j \| \qquad (3.1)$$

where H_{3x3} is the homography from the camera to projector up to a scale. With four or more correspondences between camera image and projector pattern, the 8 unknown parameters of H_{3x3} can be computed using the standard least-squares method while more sophisticated techniques can improve its robustness to noise or outliers. The above alignment step is very similar to standard camera calibration technique with the aid of checkerboard pattern [Zhang99]. The property of projector entails us to estimate homography without any human interactions with the active structured light technique we are using.

Further, as mentioned earlier, the relative mapping between any two projectors can be computed by combining Equation (2.4) and (3.1).

In order to display the final projector mosaic in the way user desires, we need to find the maximum rectangle inscribing the union of illuminating areas covered by all projectors. Finding optimal inscribed rectangle for a union of quadrilaterals is a 2D

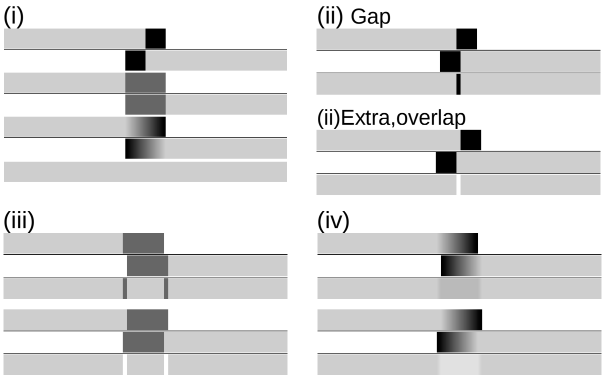


Figure 5. Effect of physical shift on resultant intensity contribution. The first row shows contribution of left projector and second row shows contribution of right projector. The bottom row shows result of addition. (i) In the ideal case with accurate registration. For (ii),(iii) and (iv), we show effect of gap versus additional overlap. (ii) Blending with only one projector contribution leads to sharp transition in overlap region (iii) Blending with half intensities (iv) Our solution, blending with linear ramps.

problem. The problem can be formulated as a constrained linear optimization problem. Since the problem is NP hard, we resort to a simple heuristic that is very easy in it's implementation but still yields very satisfactory results.

The illuminated quadrilaterals are specified in camera coordinate system. We want to compute the largest axis aligned rectangle of a given aspect ratio inscribed in the union of N quadrilaterals. The union is a simple polygon L. We assume that at least one vertex of the rectangle lies on the edges L. Since, the aspect ratio is fixed, the rectangle has 3 degrees of freedom, position of one of the vertex in 2D and scale of the rectangle. To find the approximate solution, we discretize the edges of L and solve for the largest rectangle by testing for each possible position of the rectangle vertex resting on edges of L and each possible scale. The test checks whether the rectangle edges intersect the edges of polygon L. After we have found the near optimal inscribed rectangle for projector mosaic, we update the homographies from the display coordinate system to projector pixels using Equation (2.3).

Figure 2 shows one camera snapshot of the final desktop output displayed inside the rectangle.

3.2. Intensity Blending

Regions of the display surface that are illuminated by multiple projectors appear brighter, making the overlap regions very noticeable to the users, as shown in the top of Figure 7. To make the overlap appear seamless we use the intensity blending technique, commonly known as cross-fading or feathering. We create a mask for each projector, which assign an intensity weight in [0.0, 1.0] for every pixel in the projector. Figure 3 shows one example of arrangement of four projectors used in our setup. It can be clearly seen that multiple (more than two) projectors can overlap at the same illuminated point. Weights of all projected pixels illuminating the same display surface point should add up to unity.

In the ideal case, the weights can be easily determined once we have registered all of the projectors. Some choices are shown in Figure 5(i). For example, we may allow only a single projector to illuminate a given point (5(ii)) or allow equal contribution from all projectors in the overlap region (5(iii)). In practice, due to the small errors in the registration, perspective distortions, and nonlinear barrel distortions, the projected images do not match exactly at their respective edges. In addition, over time, electro-mechanical vibrations disturb the positions of projectors. Hence, there is a need to achieve a smooth transition of weights in the overlap. The intensities in the resulting superimposition then would have reduced sensitivity to the static calibration and dynamic registration error.

Our blending algorithm uses a strategy where the weights assigned to pixels near the edges is near zero (Figure 5(iv)). The weight for pixels in non-overlap regions is clearly one, and neighboring pixels in the overlap region are assigned weights close to one. More specifically, to find the weight $A_m(\mathbf{u})$ associated with projector P_m's pixel $\mathbf{u}=(u,v,1)$, we apply the simple feathering technique, i.e., we weigh the pixels in each projector proportionally to their distance to the edge, or more precisely their distance to the nearest invisible pixel. The homographies are computed with normalized projector coordinates so that the u and v coordinates vary between [0,1]. Hence, the distance of a pixel to the closest edge in the projector P_i is described by

$$d_i(\mathbf{u}) = w(u, v) \min(u, v, 1- u , 1- v) \qquad (3.2)$$

where, $w(u, v) = 1$ if $u \in [0,1]$ and $v \in [0,1]$, = 0 otherwise. This reduces the weights assignment problem, to a simple *min* function. Further, based on the implicit pixel correspondences across the multiple projectors, we can ensure that the weight of pixels illuminating the same display surface adds up to unity. The pixel weight $A_m(\mathbf{u})$ associated with pixel $\mathbf{u}$ of projector P_m is evaluated as follows:

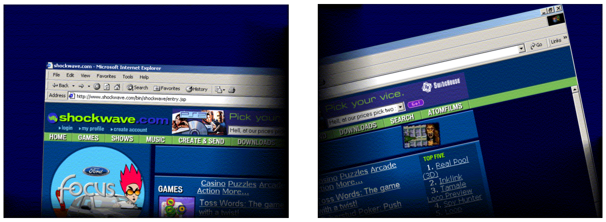
$$A_m(\mathbf{u}) = d_m(\mathbf{u}) \;/\; (\textstyle\sum_i d_i (H_{ri} H_{rm}^{-1} \mathbf{u})), \qquad i = 1,..N \qquad (3.3)$$

Figure 5 shows the effect of shift on resultant intensity contribution using various blending techniques. Intensity blending with linear ramps is more stable in presence of small mis-registration, whether the mis-registration results in a gap or extra overlap. As explained later in the rendering process, the weights are implemented using the commonly used alpha-maps for transparency in 3D graphics hardware.

Similar feathering techniques are used to mosaic multiple images taken by a camera into a single panoramic image in order to increase its insensitivity to static registration error.

4. Rendering

The final output is a scaled and seamless version of the rectangular input image even under oblique projection of individual casually installed projectors. We must pre-warp the input image, so that when projected on the planar display surface it appears aligned inside a rectangle with the given aspect ratio. To achieve this, each projectors warps and projects appropriate segment of the input image inside the inscribed rectangle. The



Projector 1 Projector 2

Figure 6. Rendering with pre-warping using homography and intensity blending. Note the clipped rectangle and intensity blending using alpha maps from Figure 4.

pixels outside the inscribed rectangle remain black, as shown in figure 6. Note that, the rendering steps as well as the input image for rendering program for each projector are identical. Hence, we do not need to explicitly segment the input image before passing on to the N different warping and blending programs.

4.1. Pre-warping using homography

The pre-warping required for the projected image is defined by the homography between pixel coordinates of the corners of projector image and pixel coordinates of the projection of the chosen inscribed rectangle in projector image space. This has been already computed as H_{ri} for each projector i.

At each projector, the rendering process to pre-warp the input image is independent. The input image, typically a full snapshot of the desktop, is loaded in the texture memory. We then texture map a unit rectangle in x-y plane (i.e. with extents [0:1,0:1]). Since $x_i \sim= H_{ri}\, u_r$, for projector i, we use the projection matrix

$$B_i = H_{ri} [\, I \mid 0 \,]$$

to render the appropriately pre-warped transformation of the input image. To be precise, we use an orthographic projection matrix, which is pre-multiplied by H_{ri}. The regions outside display coordinates [0:1,0:1] also appear in projector framebuffer. This corresponds to the region outside the inscribed rectangle but inside the polygon L, union of N illuminated quadrilaterals. We keep the background color black to avoid contributing to this region. The 3x3 homography matrix H_{ri} is converted into a 4x4 matrix used in traditional graphics pipeline by simply adding a third row and column for depth buffer calculations. The resultant depth values are ignored.

The next step is intensity correction for feathering. We load the per-pixel intensity weight into the alpha channel of a second texture map. However, the colors channels are black. The same unit rectangle is rendered with this texture map modifying the intensities. The alpha channel acts a transparency function, multiplying the underlying intensities of the warped input image.

Thus, both operations, warping using homography and the intensity correction are achieved efficiently using 3D graphics hardware. Figure 6 shows examples of the result of rendering.

5. Implementation

We implemented the system using four Mitsubishi X-80 projectors (1024x768 pixels) and a single low cost Logitech Quickcam Pro USB camera (640x480 pixels) for closed-loop calibration (Figure 1). The setup can be used in front or rear projection mode. The homography between the camera and each projector is computed using the projected checkerboard, which allows detection of 48 corresponding features. There are three main steps in our single click-and-go implementation. The execution times are as follows. (1) The projection of four different checkerboards and detection of corner features in each image (8 seconds). (2) Computing homographies and generating intensity weights (2 second) (3) Generating texture maps for alpha blending in the rendering program (3 seconds). The resulting rendering is real time. Our current bottlenecks are synchronization of the USB camera with projected images. Thus, in the current implementation, as seen in the video, after the projectors have been casually installed, there is a ‘warm-up’ time of about 15 seconds before we actually display a seamless image. However, we believe, a well-integrated should take less than 5 seconds.

We use a current generation graphics card (ATI Radeon Dual Display, cost around US $100) on a PC for rendering and display on multiple projectors. The rendering system takes a snapshot of a windows desktop, loads the resultant image, updates the texture memory, pre-warps the image and applies the intensity weights stored in alpha maps in real time. The rendering technique of perspectively correct texture mapping exploits the high quality texture filtering available on the graphics cards. As seen in Figure 1, we can skew the projectors with respect to the flat surface and with respect to each other and still correct the image so that it appears aligned and rectangular with correct aspect ratio. Given the large number of sample points (48) used for homography computation, we easily achieve subpixel registration accuracy.

It is possible to build a scalable multi-projector system by adding more projectors, even if a single camera cannot view all the projections. The only requirement is to define the relationship between any one projector and corresponding (desktop) display coordinates. This is relatively simple. In our current implementation, the inscribed rectangle in the viewing camera defines the relationship. However, any mapping between one of the camera and display coordinates in an Euclidean frame can be used. Hence, even if a single camera does not view all the projections, it is easy to propagate the relationship ($H_{rj}\ H^{-1}_{ri}$), between neighboring projectors i and j, as long as any one camera can view both the projections.

5.1. Issues

Currently we need to compute radial distortion of the viewing camera and we ignore the minor radial distortion in the projectors. Actually, it is possible to estimate the radial distortion of the camera directly from one or more homographies.

The techniques are valid only when the assumed pin-hole projection model (dual of the pin-hole camera model) is valid. Projectors typically have a limited depth of field and hence when they are oblique all pixels may not be in focus. Thus, the imagery is in focus for only a limited range of angles between the screen plane normal and the projector optical axis.

We have addressed only the geometric issues but there fundamental photometric problems. The intensities across the screen are not uniform when projector image plane is not parallel to the screen. In addition, due to manufacturing artifacts, there is visible color non-uniformity within a projector or among neighboring projectors. The gamma correction, among other design factor, introduces non-linearity in projector response.

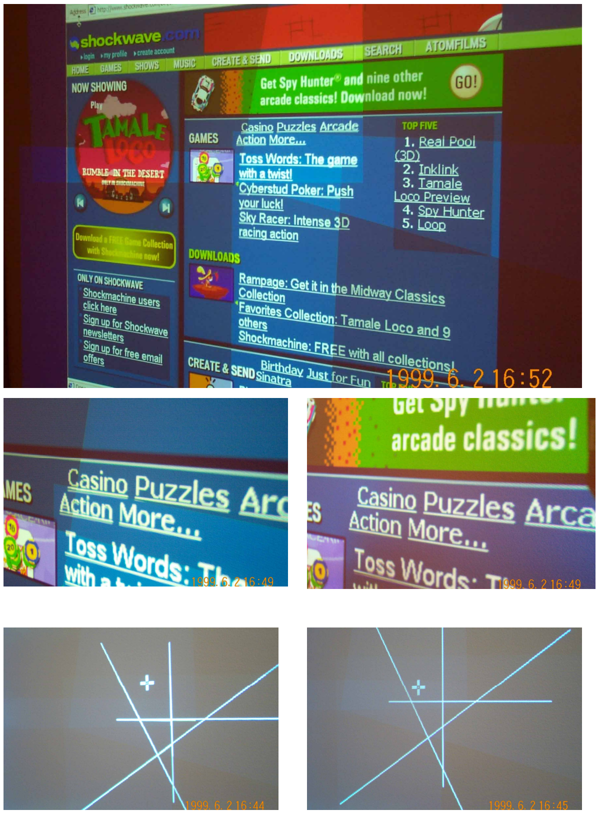


Figure 7. Close up of results. (Top) Registration Without intensity blending. (Middle) The word 'Words' is projected by all four projectors. (Bottom) Single pixel wide lines. (Left without blending, right with blending).

However, the projector behavior can be measured using a camera in the loop and compensated by changing the intensity weights during rendering (using, for example alpha blending) [Majumder00].

6. Conclusion

We have demonstrated that it is possible to create a low-cost but easy-to-use multi-projector display by exploiting computer vision techniques. The cost of the system is simply the cost of the projector in addition to the inexpensive camera and software. If right tools, such as closed-loop camera based vision algorithms and rendering software, are available, many people can pursue the dream of owning a high resolution, large format display at a store, office or even home. Our techniques reduce the time required for alignment following installation to a few seconds of 'warm-up' time. This involves projecting structured pattern and computing warping and blending parameters. Comparable to starting a digital camera or a TV, if the warm-up time is less than 10 seconds, and the alignment process is completely automatic, consumers can start using such large displays to watch movies, play games or simply as a large desktop. Quick setup and ease-of-use of a projector array becomes essential if it is intended for casual, transient use on table-tops, walls and other surfaces. As far as we know, there are no easy to use methods available to register and blend images from a set of projectors that work in under 15 sec.

Our two main contributions are (i) a fast efficient technique to find registration parameters and intensity weights with calibration using blank physical planes and (ii) a complete corrective warping technique using homographies that directly exploits the 3D graphics hardware. They lead to a system that is low-cost, easy-to-setup and operate. Our current research includes further work on extension of the current ideas to handle automatic photometric correction based on intensity measurements from the camera. We also would like to add other camera-like functions such as autogain and autofocus features. The linear intensity blending function can be improved with adaptive and real-time image-dependent [Burt83] filters.

7. References

[Burt83] P. J. Burt and E. H. Adelson. A Multiresolution Spline with Applications to Image Mosaic. ACM Trans. on Graphics, (2):217–236, 1983.

[Chen95] S.E. Chen. Quicktime VR - an image-based approach to virtual environment navigation. Comp. Graph. Proc., Annual Conf. Series (Siggraph '95), pp.29--38, 1995.

[Chen00] Y.Chen, H.Chen, D.W.Clark, Z.Liu, G.Wallace, and K.Li. Automatic alignment of high-resolution multiprojector displays using an uncalibrated camera.In IEEE Visualization 2000 ,Salt Lake City,UT,October 8-13 2000.

[ComView01] ComView Visual Systems. Comview viewscreen and view-maestro product sheets. http://www.comview-vs.com.

[Faugeras93] O. Faugeras. Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge, Massachusetts, 1993.

[Lyon85] P. Lyon. Edge-blending Multiple Projection Displays On A Dome Surface To Form Continuous Wide Angle Fields-of-View. In Proceedings of 7th I/ITEC, pages 203–209, 1985.

[Majumder00] A. Majumder, Z. He, H. Towles, and G. Welch. Color calibration of projectors for large tiled displays. In IEEE Visualization 2000, Salt Lake City, UT, USA, October 2000.

[Panoram] Panoram Technologies.http://www.panoramtech.com.

[Raskar99] R.Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In IEEE Visualization, pages 161-68, San Francisco, October 1999.

[Raskar00] Raskar, R., Immersive planar display using roughly aligned projector. In IEEE Virtual Reality, March 2000.

[SMural] Stanford Interactive Mural http://graphics.stanford.edu/~iwork/lab/mural/

[Surati99] Surati, R., Scalable Self-Calibration Display Technology for Seamless Large-Scale Displays. PhD thesis, Massachusetts Institute of Technology, 1999.

[Str-May99] Sturm, P. and Maybank, S., On plane-based camera calibration: A general algorithm, singularities, applications, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp.432—437, Fort Collins, Colorado, June 1 999.

[Szeliski97] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and texture-mapped models. *Computer Graphics (SIGGRAPH'97)*, pages 251-258, August 1997.

[Trimension] Trimension Systems. http://www.trimension-inc.com.

[Zhang99] Zhang, Z., Flexible camera calibration by viewing a plane from unknown orientations, Proc. 7th Int. Conference on Computer Vision, pp. 666—673, 1999

http://www.merl.com/projects/ProjectorMosaic/