Multi-Projector Imagery on Curved Surfaces

Ramesh Raskar Jeroen van Baar Srinivas Rao Thomas Willwacher

Mitsubishi Electric Research Labs

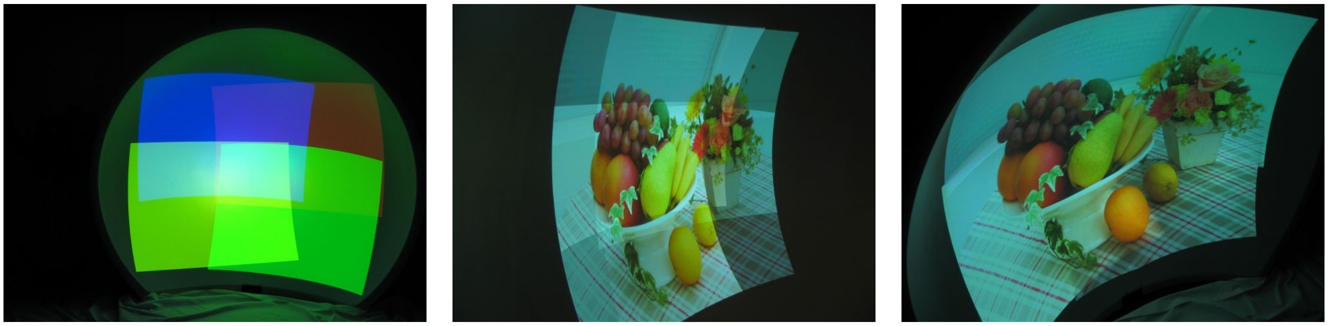

Figure 1: Visualization on curved quadric surfaces. (Left) Dome with casually aligned overlapping projectors (Middle) Registered images (Right) Displayed with intensity correction. The parametric approach leads to accurate geometric registration and perceptually seamless images. (Embedded images are high-resolution. Please zoom with PDF browser.)

Abstract

Curved screens are increasingly being used for high-resolution immersive visualization environments. We describe a new technique to display seamless images using overlapping projectors on curved quadric surfaces such as spherical or cylindrical shape. We define a new quadric image transfer function and show how it can be used to achieve sub-pixel registration while interactively displaying two or three-dimensional datasets. Current techniques for automatically registered seamless displays have focused mainly on planar displays. On the other hand, techniques for curved screens currently involve cumbersome manual alignment to make the installation conform to the intended design. We show a seamless real-time display system and discuss our methods for smooth intensity blending and efficient rendering.

CR Categories and Subject Descriptors: I.3.3, I.3.7 [Computer Graphics]: Picture/Image Generation, Virtual Reality; I.4.1, I.4.5

Keywords: camera calibration, seamless display, quadric transfer

1 Introduction

Large seamless displays using overlapping projectors is an emerging technology for constructing high-resolution semi-immersive visualization environments capable of presenting high-resolution images from scientific simulation, large format images for entertainment and surround environment for instruction. In this way, they complement other multi-projector technologies such as the CAVE [Cruz-Neira et al. 1993], Blue-C [Staadt et al. 2000] and well defined tiled planar [Jupiter Inc 2002] or curved displays [Trimension Systems Ltd 2002]. In the last few years, we have seen a number of ideas for creating seamless displays on planar screens using electro-optic approaches such as vignetting [Li and Chen 1999] or using camera in the loop [Surati 1999; Chen et al. 2000; Yang et al. 2001; Brown and Seales 2002; FuturesLab 2002] to determine the registration and blending parameters. In this paper, we extend the camera-based techniques to deal with curved screens.

1.1 Overview

Accurate estimation of geometric relationship between overlapping projectors is the key for achieving seamless displays (Fig 1). They influence the rendering algorithms and also determine soft edge blending efforts. General techniques that can handle setups where projectors have been casually installed and exploit geometric relationship between projectors and display surface eliminate cumbersome manual alignment and reduce maintenance costs. While camera-based approaches for planar screens have exploited the homography, a 3 by 3 matrix, induced due to the plane of the screen, as far as we know, there has been no work to exploit a similar parametric relationship for curved surfaces.

The relationship for surfaces that adhere to quadric equations, such as spheres, cylinders, cones, paraboloids and ellipsoids, can be defined using a quadric image transfer [Shashua and Toelg 1994].

Contributions In this paper, we present a complete set of techniques to generate seamless displays on curved quadric surface. Our technical contributions are as follows.

- Simplification of quadric transfer
- Calibration methods
- Automatic sweet-spot detection and area of display
- Software blending scheme using parametric approach
- Fast rendering strategy to exploit hardware

The sweet-spot is the ideal viewing location. Parametric approaches lead to reduced constraints on camera resolution, better tolerance to pixel localization errors, faster calibration and finally a simpler parameterized warping process. Non-parametric approaches interpolate discrete samples and hence errors in feature localization are immediately visible. Another advantage is that, unlike [Jarvis 1997; Raskar et al. 1998], our cameras do not need to be placed at or near the sweet-spot.

The scope of the paper is limited to description of algorithms related to quadric transfer, its estimation and use with graphics hardware. We do not focus on photometric issues [Majumder et al. 2000], data distribution [Humphreys et al. 2001] and load balancing [Samanta et al. 1999] during rendering. Our approach, in the current form, lacks ease of use because of the time involved in nonlinear optimization involved in estimating the parameters for image transfer, but this is an active area of research.

1.2 Related Work

Seamless Displays In commercially available planar displays, alignment is typically performed manually. However, many research groups have exploited camera-based approaches to automate this process [Surati 1999; Chen et al. 2000; Yang et al. 2001; Brown and Seales 2002]. In addition, recent software developments ease the use of PC clusters equipped with graphics cards to power immersive projection environments where multiple video projectors form a high resolution and large surface display [Humphreys and Hanrahan 1999].

Multi-projector alignment for curved screens is sometimes aided by projecting a 'navigator' pattern [Trimension Systems Ltd 2002; Jarvis 1997]. Then all the overlapping projectors are manually aligned with the grid [Trimension Systems Ltd 2002]. We have heard that, at Hayden Planetarium [Hayden Planetarium 2002] in New York, two technicians spend one hour each morning trying to adjust the registration between the seven overlapping Trimensions Prodas projectors.

An automated approach using a non-parametric process involves putting a camera at the sweet-spot. The camera observes the structured patterns projected by projector. The sampled readings are then used to build an inverse warping function between the input image and projected image by means of interpolation [Jarvis 1997; Raskar et al. 1998; Surati 1999].

We opt for a parametric method and extend the homography-based approach for planar screens to an approach based on quadric transfer. However, curved screens present new challenges. Many Euclidean quantities are required to be calibrated early in the process and non-linear optimization of parameters makes robustness an issue.

Quadric Surfaces In computer vision literature, some relevant work has used quadrics for image transfer [Shashua and Toelg 1994; Cross and Zisserman 1998]. In multi-projector systems however, although several approaches have been proposed for seamless multi-projector planar displays based on planar transfer (planar homography) relationships [Yang et al. 2001; Chen et al. 2002; Brown and Seales 2002], there has been little or no work on techniques for parameterized warping and automatic registration of higher order surfaces. This is an omission because quadrics do appear in many shapes and forms in projector-based displays. Large format flight simulators have traditionally been cylindrical or dome shaped [Scott and McColman 1999], planetariums and OmniMax theaters use hemispherical screens [Albin 1994], and many virtual reality setups use a cylindrical shaped screen.

Recently, we have proposed the use of quadric equations for curved screen applications. In [Raskar et al. 2003], we described the basic approach. In this paper, we present a complete solution for building practical multi-projector systems. We describe new calibration methods, practical resolution of robustness issues, rendering design and a fully parametric intensity blending scheme.

2 Quadric image transfer

We present the basics of quadric transfer, explain our simplification and how it is used to register overlapping projector images.

2.1 Basics

Mapping between two arbitrary perspective views of an opaque quadric surface, Q, in 3D can be expressed using a quadric transfer function, Ψ. While planar homography transfers can be computed from 4 or more pixel correspondences, quadric transfer requires 9 or more correspondences. The quadric transfer can be defined in a closed form using the 3D quadric surface, Q and additional parameters that relate perspective projection of the two views. The quadric transfer in our case means image transfer from first view to the second view.

The quadric, Q, is a surface represented by a 4×4 symmetric matrix, such that 3D homogeneous points X (expressed as a 4×1 vector) that lie on the surface satisfy the quadratic constraint,

$$X^T Q X = 0$$

The quadric, Q, has 9 degrees of freedom corresponding to the independent elements of the matrix. The matrix is symmetric and defined up to an overall scale.

The homogeneous coordinates of the corresponding pixels, x in the first view and x' in the second view are related by

$$x' \cong Bx - \left(q^T x \pm \sqrt{(q^T x)^2 - x^T Q_{33} x} \right) e$$

Given pixel correspondences (x, x'), this equation is traditionally used to compute the 21 unknowns: the unknown 3D quadric Q, a 3x3 homography matrix B and the epipole, e, in homogeneous coordinates. The epipole, e, is the image of the center of projection of the first view in the second view. The sign $\cong$ denotes equality upto scale for the homogeneous coordinates. Matrix Q is decomposed as follows.

$$Q = \begin{bmatrix} Q_{33} & q \\ q^T & 1 \end{bmatrix}$$

Thus, Q_{33} is the top 3×3 symmetric submatrix of Q and q is a 3 vector. $Q(4,4)$ is non-zero if the quadric does not pass through the origin i.e. the center of projection of the first view. Hence, it can be safely assigned to be 1.0 for most display surfaces. The final 2D pixel coordinate for homogeneous pixel x' is $(x'(1)/x'(3),\ x'(2)/x'(3))$.

2.2 Simplification

The form described above is used in [Shashua and Toelg 1994] and even in later papers such as, [Wexler and Shashua 1999] and it contains 21 variables, 4 more than needed. A simple observation is that we can remove part of this ambiguity by defining

$$A = B - eq^T \qquad E = qq^T - Q_{33}$$

and obtain the form we use,

$$x' \cong Ax \pm \left(\sqrt{x^T E x} \right) e$$

Here $x^T E x = 0$ defines the outline conic of the quadric in the first view. (The Outline conic can be geometrically visualized as the image of the silhouette or the points on surface where the view rays are locally tangent to the surface, e.g. the elliptical silhouette of a sphere viewed from outside the sphere.) A is the homography via the polar plane between the first and the second view. Note that this equation contains, apart from the overall scale, only one ambiguous degree of freedom resulting from relative scaling of E and e. This can be removed by introducing an additional normalization constraint, such as $E(3,3) = 1$. Further, the sign in front of the square root is fixed within the outline conic in the image. The sign is easily determined by testing the equation above by plugging in coordinates for one pair of corresponding pixels.

Note that the parameters of the quadric transfer can be directly computed from 9 or more pixel correspondences in a projective coordinate system. So it is tempting to follow a approach similar to estimating planar homography for planar displays without computing any Euclidean parameters. However, as described later, in practice it is difficult to robustly estimate the epipolar relationship in many cases. Hence, we follow a pseudo-Euclidean approach as described below.

2.3 Approach

All registration information is calculated relative to a camera stereo pair. We assume that the stereo camera pair can see the entire 3D surface. One of the cameras is arbitrarily chosen as the origin. The cameras here are used to determine only the 3D points on the display surface and not for any color sampling. Hence, any suitable 3D acquisition system can be used. The outline of our technique is as follows. The details are in Section 3 and 4.
During **pre-processing**, the following steps are performed.

- For each projector i
  - ▷ Project structured light pattern with projector
  - ▷ Detect features in stereo camera pair and reconstruct 3D points on the display surface, which correspond with the structured light features.
- Fit a quadric, Q, to all the 3D points detected
- For each projector i
  - ▷ Find its pose wrt the camera using the correspondence between projector pixels and 3D coordinates of points they illuminate
  - ▷ Find the quadric transfer, Ψ_{0i}, between the camera and projector
  - ▷ Find intensity blending weights, Φ_i, in overlap regions

At **run time**, the rendering for 2D images or 3D scenes follows the following steps.

- Read the input image in the 2D video or compute the input image by rendering a 3D scene from the virtual viewpoint
- For each projector i
  - ▷ Pre-warp input image into projectors framebuffer using quadric transfer, Ψ_{0i}
  - ▷ Attenuate pixel intensities with blending weights, Φ_i

This approach, however, involves several issues. The quadric transfer estimation, although a linear operation, requires non-linear optimization to reduce pixel re-projection errors. It is difficult to estimate projector pose (external parameters) because the 3D points projected on the quadric are usually nearly planar leading to a degenerate condition. These plus other issues, and practical solutions are discussed below.

3 Calibration

The goal is to compute the parameters of quadric transfer, $\Psi_{0i} = \{A_i, E_i, e_i\}$, so that the projected images are geometrically registered on the display surface. The method to calculate quadric transfer parameters directly from pixel correspondences involves estimating the 4x4 quadric matrix, Q, in 3D [Shashua and Toelg 1994], [Cross and Zisserman 1998] using a triangulation of corresponding pixels and a linear method. If the internal parameters of the two views are not known, all the calculations are done in projective space after computing the epipolar geometry, i.e. the epipoles and the fundamental matrix. However, we noticed that when projectors rather than cameras are involved, the linear method produces very large re-projection errors in estimated 3D quadric, Q. The errors are of the order of 30 pixels for XGA projector. In addition, the fundamental matrix is inherently noisy given that the 3D points on the quadric surface illuminated by a single projector do not have significant depth variation in display setting such as segments of spherical or cylindrical surfaces. We instead follow a pseudo-Euclidean approach where the internal and external parameters of the camera and the projectors are known *approximately*. and are used to estimate Euclidean rigid transformations. Hence, unlike the planar case, computation of accurate image transfer for curved screens, involves three-dimensional quantities. We describe our approach and use of approximate Euclidean parameters for estimating warping parameters.

3.1 Quadric Surface

We use a rigid stereo camera pair, C_0 and C'_0, as a base for computing all the geometric relationships. We arbitrarily choose one of the cameras to define the origin and coordinate system. We calibrate the small baseline stereo pair with a small checkerboard pattern [Zhang 1999]. Note that the cameras do not necessarily represent the sweet-spot in this setup which is an important difference with respect to some of the non-parametric approaches.

The stereo pair observes the structured patterns projected by each projector (Fig 2) and using triangulation computes a set of N 3D points, $\{X_j\}$, on the display surface. The quadric, Q, passing though each X_j is computed by solving a set of linear equations, $X_j^T Q X_j = 0$, for each 3D point. This equation can be written in the form

$$\chi_i \, V = 0$$

where, χ_i, is a 1×10 matrix which is a function of X_i only and V is a homogeneous vector containing the distinct independent unknown variables of Q. With $N \geq 9$, we construct a $N \times 10$ matrix X and solve the linear matrix equation

$$\mathsf{X} V = 0$$

Given points in general position, the elements of V (and hence Q) are the one dimensional null-space of X.

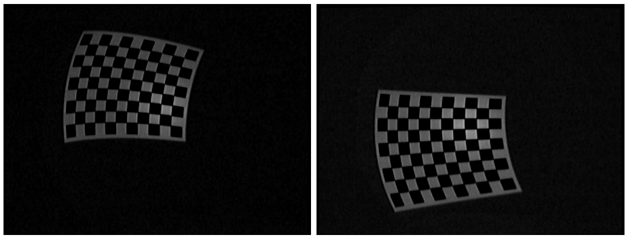

Figure 2: Images captured by the 640x480 resolution camera during calibration. The resolution of each projector is significantly higher at 1024x768 and yet is captured in only a small part of the camera view.

3.2 Projector View

In addition to the quadric, Q, we need to know the internal and external parameters of each projector with respect to the camera origin. We use the correspondence between the projector pixels and coordinates of the 3D points they illuminate to compute the pose and internal parameters.

However, finding the pose of a projector from known 3D points on a quadric is error-prone because the 3D points are usually quite close to a plane leading to an unstable solution [Faugeras 1993; Forsyth and Ponce 2002]. Dealing with near-planar points is a difficult problem. If points are distributed in depth, we can easily use a linear method to estimate the internals as well as externals of the projector. On the other hand, if the points are known to be planar, we can estimate the externals if some of the internals are known.

How about bringing in a temporary surface to add non-planar points for the sake of calibration? In most cases this is impractical or cumbersome. The temporary surface will have to be approximately the same size as the display surface. Our goal is to compute the quadric transfer completely automatically.

For dealing with near-planar surfaces, we are required to use an iterative algorithm. If we know the projector internal parameters, we can first find an initial guess for external parameters based on homography and then use an iterative algorithm described in [Lu et al. 2000]. We use Powell's method for nonlinear refinement of reprojection error. However, estimating projector internals is equally difficult. If the projectors cannot be easily moved, as mentioned above, calibrating them usually requires large surfaces illuminated in two or more positions.

Our strategy is to use projector internal parameters that are approximately known. We find internal parameters of just one projector and use these internal parameters for all projectors for all future settings. The same projector at a later time and other projectors will clearly have different zoom settings and have other mechanical or optical deviations. In addition, the external parameters computed by iterative method in [Lu et al. 2000], will also be approximate.

3.3 Camera to Projector Transfer

The idea is to use the perspective projection parameters of the camera along with approximate projection matrix of the projector to find the camera to projector quadric transfer using linear methods. Then refine the solution using non-linear optimization.

The quadric transfer parameters, $\Psi_{0i} = \{A_i, E_i, e_i\}$, are easy to calculate from Q, camera projection matrix $[\ I\ |0]$ and projector projection matrix $[\ P_i\ |e_i]$.

$$A_i = P_i - e_i q^T \qquad\qquad E_i = qq^T - Q_{33}$$

In our tests the parameters found by the linear method turned out to be far too imprecise for our purpose. We observed misalignment between the projectors of 15-30 pixels on screen. Since seamless displays require sub-pixel accuracy, we have to apply a nonlinear minimization to refine the results obtained via the linear method. As objective function we take the total squared transfer error for all pixels

$$\varepsilon_i = \sum_j \left(\frac{x_i^j(1,2)}{x_i^j(3)} - \frac{\hat{x}_i^j(1,2)}{\hat{x}_i^j(3)} \right)^2$$

with $\hat{x}_i^j$ being the transferred points for each x_i^j

$$\hat{x}_i^j = A_i x_i^j \pm \sqrt{{x_i^j}^T E_i x_i^j}\, e_i$$

Note that the sign found using the linear method, which is same for all the pixels, remains the same during the nonlinear optimization. We used Nelder-Mead Simplex and obtained very satisfactory results.

3.4 Partial Euclidean Reconstruction

There are two questions here. (i) Why not ignore Euclidean approach altogether and directly go for projective space and nonlinear optimization and (ii) If we have accurate projector internal parameters, can we avoid non-linear optimization stages?

As mentioned earlier, ignoring Euclidean viewing parameters and solving the quadric transfer purely from pixel correspondences leads to poor re-projection errors. The estimated 3D quadric, Q, cannot be used as a good guess for further non-linear optimization. In fact, in most of our tests, the solution did not converge.

Using accurate projectors internals only reduces the reprojection errors but does not eliminate them. This is because, many kinds of errors are propagated in the three dimensional Euclidean calculations, including estimating 3D points on the display surface by triangulation, estimating the 3D quadric using linear methods and finding the projector pose. The non-linear optimization attempts to minimize the physical quantity we care about the most, i.e. pixel re-projection error in image transfer from camera to projector for known corresponding set of pixels. Since the correspondence between overlapping projector pixels is indirectly defined by this image transfer equation, minimizing pixel re-projection errors ensures geometric registration between the displayed projector pixels.

4 Rendering

The rendering involves a simple two-pass approach. For 2D data, we extract the appropriate input image. For 3D scenes, we first render the 3D models from the sweet-spot. In the second pass, the resultant image is then warped using the quadric image transfer into the projector image space.

4.1 Virtual View

When 3D scenes are displayed on a curved screen, the images are perspectively correct from only a single point in space. This 3D location is popularly known as the sweet-spot or the virtual viewpoint. As the viewer moves away from it, the images look distorted. In addition, one needs to specify the view frustum i.e. the viewing direction (principal axis) and the extent (field) of view. In some cases it is possible to automatically determine the sweet spot. For example, for a concave spherical dome, the center of the dome can be considered a good sweet-spot. This can be determined directly from the equation of the quadric, Q i.e. $Q(1,1)\ q$. For cylindrical screen, a point on the axis of the cylinder that is midway along the extent of the cylinder is a good choice. Sometimes, the sweet-spot is decided by practical considerations e.g. a spot that is approximately at the human height is considered ideal. The images are almost always aligned with the world horizontal and vertical. In our case, the virtual viewpoint can be interactively moved (and fixed) because we have approximate Euclidean reconstruction of the display geometry. One potential problem is how to physically mark the sweet-spot location. Since the location's 3D coordinates are known in the camera coordinate system, a cheap method is to just use a tape measure and locate with respect to the camera!

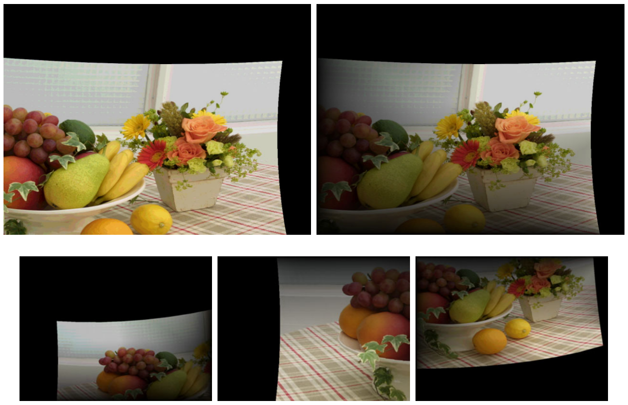

Figure 3: Top row: Image in top-right projector's framebuffer, before and after attenuation with alpha map Φ_i. Bottom row: Other three projectors with intensity correction. Note the outer black areas which are automatically generated after quadric transfer.

Sweet-spot from Surface Points A robust strategy, when it is difficult to determine the parameters of the virtual viewpoint for rendering i.e. location, lookat and field of view, is to find the best-fit plane to the set of points found on the illuminated part of the display surface. We fit an oriented bounding box (OBB) to fit the set of 3D points $\{X_i\}$ on the display surface [Gottschalk, S. and Lin, C. and Manocha, D. 1996]. A point at a suitable distance in front of the screen, along the vector passing through the center of this box and normal to the best-fit plane can be chosen as the sweet spot.

The procedure to find the OBB in our case is simple because all the 3D points on a quadric lie on the convex hull of those 3D points. Lets say Y is the $N \times 3$ matrix of $\{X_i - \overline{X}\}$ where $\overline{X} = (\overline{x}, \overline{y}, \overline{z})$ is the centroid of the N 3D points. The eigenvector corresponding to the smallest eigenvalue of the 3×3 matrix $Y^T Y$ gives the normal to the best-fit plane i.e. axis of minimum variance. On the other hand, the largest side of the oriented bounding box, i.e. the extent of the 3D points projected in the best-fit plane gives the approximate 'diameter' of the screen. The distance of sweet spot in front of the screen can be chosen to be a proportional to this diameter depending on the application and the desired field of view. For immersive applications, one would like to have a large field of view and hence the distance would be about half of the diameter. For group viewing, the distance would be comparable to the diameter.

We recalculate the quadric transfer, Ψ_i, between virtual view image space and each projector framebuffer. The process is very similar to computing Ψ_{0i}. We first find projection of 3D points on quadric surface into the virtual view image space. Then the corresponding pixels between virtual view and projector frame buffer, in addition to internal and external parameters of the virtual view and projector are sufficient to update Ψ_i.

4.2 Display Region

The view frustum for the virtual view is defined using the sweet-spot and the extents of the OBB. The look-at vector is from the virtual viewpoint toward the center of the OBB. However, since the cameras can see beyond the union of the area illuminated by casually installed overlapping projectors, we need to crop the view frustum to an aesthetic shape such as a rectangle or a disk.

For 3D applications, we simply draw a set of black quadrilaterals to cutout the areas outside the desired display region. For example, for a rectangular view, the viewport is made by four large quads near the outer edge of the viewport in the framebuffer. The black quads along with rest of the 3D models get rendered and warped as described below (Figure 3).

For 2D applications, areas outside the input image to be displayed are considered black.

4.3 Image Transfer

Given a 3D vertex, M, in the scene to be rendered, we find its screen space coordinates, m, in the virtual view. Then, we find the transferred pixel coordinate, m_i, in the framebuffer of projector i, using the quadric transfer, $\Psi_i = \{A_i, E_i, e_i\}$. The polygons in the scene are then rendered with vertices M replaced with vertices m_i. Thus the rendering process at each projector is very similar. Each projector framebuffer automatically picks up the appropriate part of the virtual view image and there is no need to explicitly figure out the extents of the projector.

```
• At each projector, $i$
  ▷ For each vertex $M$
      Compute pixel $m$ via VirtualViewProjection( $M$ )
      Compute warped pixel $m_i$ via quadric transfer $\Psi_i(m)$
  ▷ For each triangle $T$ with vertices $\{M^j\}$
      Render triangle with 2D vertices $\{m_i^j\}$
```

There are two issues with this approach. First, only the vertices in the scene, but not the polygon interiors, are accurately pre-warped. Second, visibility sorting of polygons needs a special treatment.

After quadric transfer, the edges between vertices of the polygon, theoretically, should map to second-degree curves in projector frame buffer. But scan conversion converts them to straight-line segments between the warped vertex locations. This problem will not be discernible if only a single projector is displaying the image. But, overlapping projectors will create individually different deviations from the original curve and hence the edge will appear mis-registered on the display screen. Therefore, it is necessary to use sufficiently fine tessellation of triangles. Commercial systems are already available that tessellate and pre-distort the input models on the fly [Evans and Sutherland 2002; Idaszak et al. 1997; Kelly et al. 1994] so that they appear straight in a perspectively correct rendering on the curved screen. So our method is compatible with fine tessellation provided by such systems. Pre-distortion of the scene geometry in commercial systems is used to avoid the two-pass rendering, which involves texture-mapping result of the first pass. In our case, instead of pre-distorting the geometry, we pre-distort the image space projection. Our approach, arguably, is more practical thanks to the programmable vertex shaders now available (see Appendix).

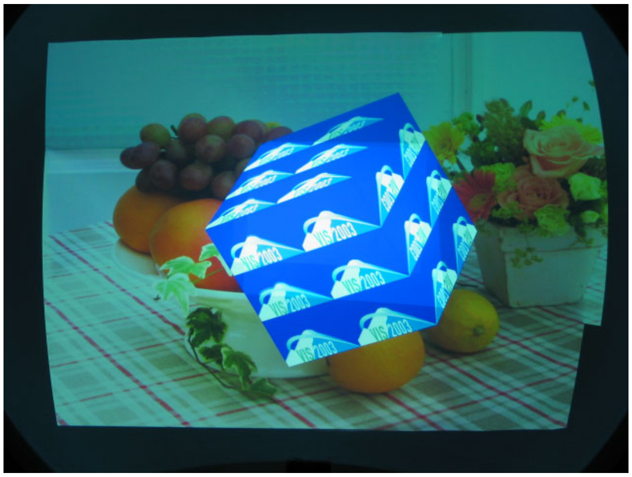

Figure 4: A mix of real-time 3D scene displayed in front of a 2D background with correct depth sort and texture mapping (Please see the video).

Scan Conversion Issues When pixel locations in the projection of a triangle are warped, information needs to be passed along so that the depth buffer will create appropriate visibility computations. In addition, for perspectively correct color and texture coordinate interpolation, the appropriate 'w' values need to be passed. Our solution is to post-multiply the pixel coordinates with 'w'.

$$\begin{aligned} m(x,y,z,w) &= \text{VirtualViewProjection}(M(X)) \\ m'_i(x'_i, y'_i, w'_i) &= \Psi_i(m(x/w, y/w), 1) \\ m_i(x_i, y_i, z_i, w_i) &= [wx'_i/w'_i, wy'_i/w'_i, z, w] \end{aligned}$$

Thus, m_i has appropriate final pixel coordinate $(x'_i/w'_i, y'_i/w'_i)$ due to quadric transfer along with original depth and w values. Appendix shows the code for vertex shader and Figure 4 shows a photo of the dome displaying a sample interactive animation that is synchronized and displayed on the dome with four overlapping projectors.

For rendering 2D input images, we densely tessellate the virtual view image space into triangles and map the image as a texture on these triangles. Vertex, m, of each triangle is warped using the quadric transfer into vertex (and pixel) m_i as above. Scan conversion automatically transfers colors and texture attributes at m to m_i and interpolates in between. (3D scenes can also be rendered in this manner, by first computing the input image.)

Note that warping an image using a quadric transfer is different than rendering quadric surfaces [Watson and Hodges 1989].

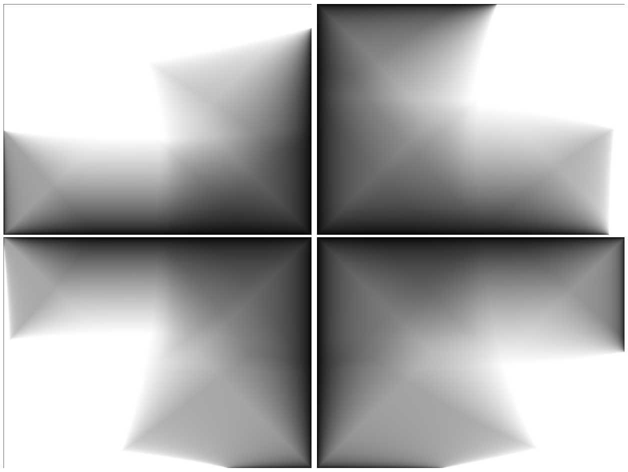

Figure 5: Calculated alpha maps, Φ_i, for intensity correction of four overlapping projectors.

4.4 Intensity Blending

Pixels intensities in the areas of overlapping projectors are attenuated using alpha blending of the graphics hardware. Using the parametric equations of quadric transfer, the alpha maps are calculated robustly and efficiently.

For every projector pixel, x_i in projector i, we find the corresponding pixels in projector j using the equation

$$x_j \cong \Psi_{0j} \left(\Psi_{0i}\right)^{-1} (x_i)$$

For cross-fading, pixels at the boundary of the projector frame buffer are attenuated. Hence, the weights are proportional to the shortest distance from the frame boundary. The weight assigned to pixel x_i, expressed in normalized window pixel coordinates (u_i, v_i) which are in the range $[0,1]$, is

$$\Phi_i(x_i) \cong d(x_i) \Big/ \sum_j d(x_j)$$

where, $d(x)$ is $min(u, v, 1-u, 1-v)$ if $0 \leq u, v \leq 1$, else $d(x) = 0$. Thanks to a parametric approach, we are able to compute corresponding projector pixels and hence the weights at those locations at sub-pixel registration accuracy. The sum of weights at corresponding projector pixels accurately adds to 1.0.

At each projector, the corresponding alpha map is loaded as a texture map and rendered as screen aligned quads as the last stage of the rendering (Fig 5).

5 Results

Please see http://www.merl.com/people/raskar/Viz03/ for more images and higher resolution videos of the results. (The submitted images and videos are lower resolution due to file size limits.)

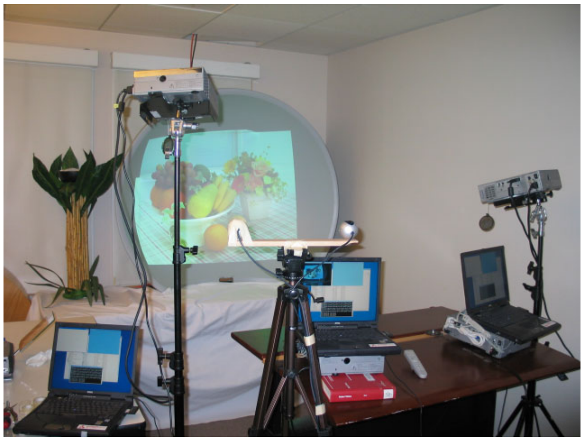

Figure 6: The setup of four casually installed projectors and stereo camera pair with a concave dome. The camera pair is not near the sweet spot.

Our **prototype system** (Fig 6) involves four Mitsubishi X80 LCD projectors, XGA (1024x768) resolution at 1500 lumens. The stereo camera pair is built with two cheap low-end Logitech USB VGA (640x480) resolution webcams which suffer from Bayer color patterning. The color interpolation negatively affects the accuracy of feature location. We intentionally chose the lower quality cameras to demonstrate that a parametric approach can overcome feature localization errors as long as the errors are Gaussian distributed and independent. Each projector is assigned to a Dell Inspiron Laptop for flexibility wrt to system setup and location. The laptops lack rendering performance, however they can easily be replaced with a cluster of PCs with higher end graphics hardware. To demonstrate our techniques we use an eLumens VisionStation hemispherical dome with a diameter of 1.5 meter. The accompanying video shows registration and intensity correction for both a concave spherical segment, as a well as a convex spherical segment (the back-side of the eLumens dome).

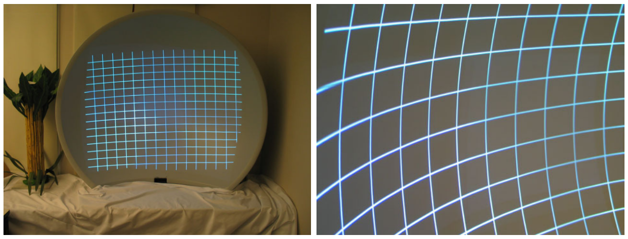

Figure 7: Registration accuracy. (Left) Four overlapping projectors projecting a grid (Right) Closeup of display region where all four projectors overlap.The embedded images are 2000x1500 pixels. Please zoom with your PDF browser.

Our **calibration process** is slow compared to the planar homography based systems [Raskar et al. 2002], [Brown and Seales 2002] which take just two seconds per projector. The two time consuming steps are computing the projector pose from near-planar 3D points using an iterative scheme in [Lu et al. 2000], and the non-linear refinement of $\Psi_{0i} = \{A_i, E_i, e_i\}$ to minimize pixel re-projection errors. Currently our non-optimized code takes about 45 - 60 seconds per projector. The linear optimization for projector pose takes about 4 seconds and the non-linear optimization takes about 40-55 seconds. The time needed for the non-linear optimization increases with the number of features detected from the projected checker patterns. Typically around 100-200 iterations are necessary and we noticed that the iterations converge in most cases.

To determine how well the registration of projectors using pose estimation from near-planar 3D points performs, we compare our results to the pixel **re-projection errors** of a fully calibrated system. In the fully calibrated system we determine internals and externals for the two cameras and the projectors. We use OpenCV library [Intel 2002] to determine the camera internals. The externals are determined from point correspondences of planar surfaces at multiple depths and orientations. Projector internals and externals are then found from triangulated 3D points and corresponding projector image checker corner pixels. Table 1 shows the results of theoretical pixel re-projection errors for the linear estimation of Q and after non-linear refinement. The setup contains four projectors, Proj1 and Proj3 were closer to the camera(-pair). As mentioned earlier, temporarily installing large surfaces at multiple depth and orientations in not practical in most of the real settings. Hence, in our pseudo-Euclidean method, we assume the same approximate projector internal matrix for all projectors and compute the projector external parameters from the near-planar points observed in the given display setting. The RMS re-projection errors after linear estimation are large (Fig. 8), but this is a good initial guess. After nonlinear refinement, the error is about one pixel which is acceptable. It is important to note that the computed pixel re-projection errors do not directly predict pixel mis-registration errors on the display surface. As seen in Fig. 8 and in the video, after linear estimate the visible errors on display surface are 15 to 30 pixels although predicted projector to projector re-projection error is about 12 pixels (twice the camera to projector transfer error of 6 pixels).

	Fully Calibrated		Pseudo-Euclidean	
	Linear	NL Optim	Linear	NL Optim
Proj1	3.5	0.3	5.4	0.73
Proj2	5.3	0.5	7.1	1.02
Proj3	3.7	0.35	5.6	0.79
Proj4	4.8	0.45	6.5	0.91

Table 1: Comparing results with an ideal system. RMS re-projection pixel error in quadric transfer of camera pixels to projector pixels.

We have used different resolution $m \times n$ checker patterns to determine the optimal one wrt to detection reliability, computation time to compute the quadric transfer, and pixel re-projection error. Higher resolution patterns typically result in smaller re-projection errors, but detection reliability decreases and non-linear refinement time significantly increases. Lower resolution patterns decrease computation time, but are too coarse for a 3D quadric surface and result in larger re-projection errors . We have found that a 11×9 checker pattern best matched our criteria (Fig. 2).

The casual placement of the dome surface results in a small deviation from its intended spherical form. However we demonstrate that our technique still produces accurate results (Fig. 7). The accompanying website contains more images demonstrating registration quality.

We have also applied our technique for registering the projections on the convex side of the dome surface. This demonstrates the generality of our technique. Projecting on the convex side would be particularly useful if the dome was made of transparent material. The rear-projection with user inside the dome will provide full immersive experience without blocking projectors. Fig. 9 shows a setup of three overlapping projectors. The left image shows registration, the right image also has intensity correction applied. We observed that projected patterns are more distorted due to the convexity of the surface in all directions. Therefore feature detection using checker patterns is more difficult in this case.

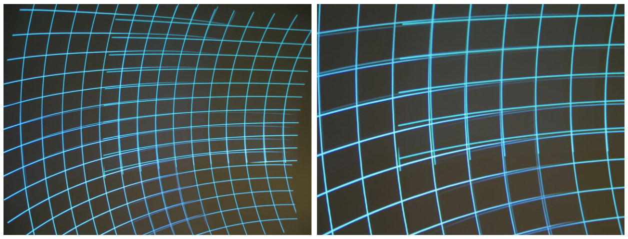

Figure 8: Registration after linear estimate of quadric transfer. The errors on screen are 15 to 30 pixels.

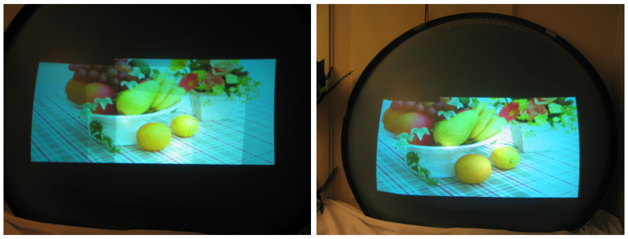

Figure 9: Three projector system on a convex quadric surface, before and after intensity blending.

6 Conclusion

We have presented techniques that are ideal for building multi-projectors curved displays for 2D or 3D visualization without expensive infrastructure and casual alignment. Our automatic registration exploits quadric image transfer and eliminates tedious setup and maintenance, and hence reduces cost. The techniques could simplify the construction, calibration and rendering process for widely used applications in flight simulators, planetariums and high-end visualization theaters. New possible applications are low-cost and flexible dome displays, shopping arcades and projection on cylindrical columns or pillars. The approach and proposed ideas can also be treated as an intermediate step between planar to arbitrary free form shaped displays.

The new parametric approach allows an elegant solution to a problem that has so far been solved by discrete sampling. An advantage is that, unlike [Jarvis97][Raskar98], our cameras do not need to be placed at the sweet-spot. This is important in real-world applications where sweet-spot is usually a busy area crowded with observers and where expensive equipment cannot be kept. In some cases, keeping a camera at the sweet-spot means using a very wide-field of view camera which is expensive and will have radial or fish-eye distortion.

We are currently working on extending our technique to work for general curved surfaces or surfaces that have minor deviations from quadric properties. We also would like to extend to the case where quadric transfers cannot be easily cascaded using a small number of cameras, such as wrap-around displays in traditional inside-looking-out and translucent or opaque outside-looking-in convex surfaces.

(Please see Paper Videos and Images http://www.merl.com/people/raskar/Viz03/)

7 Appendix

Code for Vertex Shader for quadric transfer (in Cg language)

```
vertout main( appin IN, uniform float4x4 modelViewProj, uniform float4 constColor,
uniform float3x3 A, uniform float3x3 E, uniform float3 e) {

    vertout OUT;
    float4 m1 = float4(IN.position.x, IN.position.y, IN.position.z, 1.0f );
    float4 m, mi ; float3 m2,mp; float scale;
    m = mul( modelViewProj, m1);
    m2.x = m.x/m.w; m2.y = m.y/m.w; m2.z = 1;
    scale = mul(m2, mul(E,m2));
    mp = mul(A,m2) + sqrt(scale)*e;
    mi.x = m.w * (mp.x)/(mp.z);
    mi.y = m.w * (mp.y)/(mp.z);
    mi.zw = m.zw;
    OUT.position = mi;
    OUT.color0 = IN.color0; // Use the original per-vertex color specified
    return OUT;

}
```

References

Albin, E. F. 1994. Planetarium special effects: A classification of projection apparatus. In *The Planetarian*, vol. 23, 12–14.

Brown, M. S., and Seales, W. B. 2002. A Practical and Flexible Large Format Display system. In *The Tenth Pacific Conference on Computer Graphics and Applications*, 178–183.

Chen, Y., Chen, H., Clark, D. W., Liu, Z., Wallace, G., and Li, K. 2000. Automatic Alignment of High-Resolution Multi-Projector Displays Using An Un-Calibrated Camera. In *IEEE Visualization 2000*.

Chen, H., Sukthankar, R., Wallace, G., and Li, K. 2002. Scalable Alignment of Large-Format Multi-Projector Displays Using Camera Homography Trees . In *IEEE Visualization*.

Cross, G., and Zisserman, A. 1998. Quadric Surface Reconstruction from Dual-Space Geometry. In *Proceedings of 6th International Conference on Computer Vision(Bombay, India)*, 25–31.

Cruz-Neira, C., Sandin, D., and DeFanti, T. 1993. Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE. In *SIGGRAPH 93 Conference Proceedings*, vol. 27, 135–142.

Evans and Sutherland, 2002. http://www.es.com.

Faugeras, O. 1993. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press.

Forsyth, and Ponce. 2002. *Computer Vision, A Modern Approach*.

FuturesLab, 2002. ActiveMural, Argonne National Labs.

Gottschalk, S. and Lin, C. and Manocha, D. 1996. OBBTree: a hierarchical structure for rapid interference detection . 171 – 180.

Hayden Planetarium, 2002. http://www.amnh.org/museum/press/feature/rosefactsheet.html.

Humphreys, G., and Hanrahan, P. 1999. A Distributed Graphics System for Large Tiled Displays. In *IEEE Visualization*.

Humphreys, G., Eldridge, M., B., I., Stoll, G., Everett, M., and Hanrahan, P. 2001. WireGL: A Scalable Graphics System for Clusters. In *Proceedings of SIGGRAPH 2001*.

Idaszak, R. L., Zobel, J., Richard, W., and Bennett, D. T., 1997. US Patent 6,104,405, Alternate Realities, now Elumens, USA, Feb 26, 1997. Systems, methods and computer program products for converting image data to nonplanar image data.

Intel, 2002. Open Source Computer Vision Library (OpenCV), http://www.intel.com/research/mrl/research/opencv/.

Jarvis, K. 1997. Real Time 60Hz Distortion Correction on a Silicon Graphics IG. *Real Time Graphics 5*, 7 (Feb.), 6–7.

Jupiter Inc, 2002. http://www.jupiter.com.

Kelly, W. A., Quick, L. T., Sims, E. M., and Tackaberry, M. W., 1994. US Patent 5,319,744, General Electric Company, USA, June 7, 1994. Polygon fragmentation method of distortion correction in computer image generating systems.

Li, K., and Chen, Y. 1999. Optical Blending for Multi-Projector Display Wall System. In *Proceedings of the 12 th Lasers and Electro-Optics Society 1999 Annual Meeting*.

Lu, C., Hager, G., and Mjolsness, E. 2000. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 6, 610–622.

Majumder, A., He, Z., Towles, H., and Welch, G. 2000. Color Calibration of Projectors for Large Tiled Displays. In *IEEE Visualization 2000*.

Raskar, R., Welch, G., and Fuchs, H. 1998. Seamless Projection Overlaps Using Image Warping and Intensity Blending. In *Fourth International Conference on Virtual Systems and Multimedia*.

Raskar, R., VanBaar, J., and Chai, X. 2002. A Lost Cost Projector Mosaic with Fast Registration. In *Fifth Asian Conference on Computer Vision*, 161–168.

Raskar, R., vanBaar, J., Beardsley, P., Willwacher, T., Rao, S., and Forlines, C. 2003. Geometrically aware projector and self-configurable displays.

Samanta, R., Zheng, J., Funkhouser, T., Li, K., and Singh, J. P. 1999. Load Balancing for Multi-Projector Rendering Systems. In *SIGGRAPH/Eurographics Workshop on Graphics Hardware*.

Scott, K., and McColman, R. 1999. Report of the IPS Technical Committee: Full-Dome Video Systems. In *The Planetarian*, vol. 28, 25–33.

Shashua, A., and Toelg, S. 1994. The quadric reference surface: Theory and applications. Tech. Rep. AIM-1448.

Staadt, O., Gross, M., Kunz, A., and Meier, M. 2000. The blue-c: Integrating real humans into a networked immersive environment. In *ACM Collaborative Virtual Environments 2000*.

Surati, R. 1999. *Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays*. PhD thesis, Massachusetts Institute of Technology.

Trimension Systems Ltd, 2002. , jan. http://www.trimension-inc.com/.

Watson, B., and Hodges, L. 1989. A fast algorithm for rendering quadratic curves on raster displays. In *Proc. 27th Annual SE ACM Conference*.

Wexler, Y., and Shashua, A. 1999. Q-warping: Direct Computation of Quadratic Reference Surfaces. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June, 1999*.

Yang, R., Gotz, D., Hensley, J., and Towles, H. 2001. PixelFlex: A Reconfigurable Multi-Projector Display System. In *IEEE Visualization 01*.

Zhang, Z. 1999. Flexible camera calibration by viewing a plane from unknown orientations. In *7th Int. Conference on Computer Vision*, 666–673.