Auto-Calibration of Multi-Projector CAVE-like Immersive Environments

Behzad Sajadi, *Student Member, IEEE,* Aditi Majumder, *Member, IEEE,*

Abstract—In this paper, we present the first method for the geometric auto-calibration of multiple projectors on a set of CAVE-like immersive display surfaces including truncated domes and 4 or 5-wall CAVEs (three side walls, floor and/or ceiling). All such surfaces can be categorized as swept surfaces and multiple projectors can be registered on them using a single uncalibrated camera without using any physical markers on the surface. Our method can also handle non-linear distortion in the projectors, common in compact setups where a short throw lens is mounted on each projector. Further, when the whole swept surface is not visible from a single camera view, we can register the projectors using multiple pan and tilted views of the same camera. Thus, our method scales well with different size and resolution of the display. Since we recover the 3D shape of the display, we can achieve registration that is correct from any arbitrary viewpoint appropriate for head-tracked single-user virtual reality systems. We can also achieve wallpapered registration, more appropriate for multi-user collaborative explorations.

Though much more immersive than common surfaces like planes and cylinders, general swept surfaces are used today only for niche display environments. Even the more popular 4 or 5-wall CAVE is treated as a piecewise planar surface for calibration purposes and hence projectors are not allowed to be overlapped across the corners. Our method opens up the possibility of using such swept surfaces to create more immersive VR systems without compromising the simplicity of having a completely automatic calibration technique. Such calibration allows completely arbitrary positioning of the projectors in a 5-wall CAVE, without respecting the corners.

Index Terms—geometric registration, calibration, multi-projector displays, tiled displays, CAVEs, immersive displays

1 Introduction

A common way to build high-resolution immersive virtual reality systems is to tile multiple projectors on non-planar displays. Automated registration of such displays using a single uncalibrated camera is instrumental for their easy and inexpensive deployment. Using a single uncalibrated camera for registration is much simpler than (a) using calibrated stereo cameras to reconstruct the display shape via structured light patterns [1], [2], [3], [4], [5], [6]; and/or (b) attaching obtrusive fiducials on the display [7], [8]. We showed in [9], [10] that when considering a smooth or piecewise-smooth vertically extruded display surface – cylinders or 3-wall CAVEs – automated registration can be achieved without using a calibrated stereo pair – but just a single uncalibrated camera.

Swept surfaces are formed by sweeping a *profile curve* along a *path curve* to create the 3D shape of the display surface (Figure 1). Truncated domes or 4 or 5 wall CAVEs (3 side walls with a ceiling and/or floor) are commonly used swept surfaces. Unlike vertically extruded surfaces which are curved only in the horizontal direction and not in the vertical direction – swept surfaces are curved in both directions. Hence, they can provide greater immersion than vertically extruded surfaces.

Swept surfaces are easy to build which makes them popular choices in mechanical design applications. Further, unlike domes, another popular immersive shape, most swept surfaces lend themselves easily to an intuitive 2D parametrization along the parametrization of the path and profile curves. This provides an intuitive 2D parametrization, important for wallpapering images on a display for collaborative multi-user applications. Wallpapering does not provide a perspectively correct imagery from any viewpoint. However, it provides an acceptable multi-user viewing experience.

Currently, no automatic calibration technique exists for multiple-projector swept surfaces. Even the 5-wall CAVEs, widely used for immersive VR environments, are treated as piecewise planar surfaces rather than swept surfaces. Hence, projectors are not allowed to overlap across the corners. Multiple projectors on each planar wall are calibrated separately and semi-automatic ad-hoc methods are used to achieve registration across the corners.

In this paper, we present the first automatic method for calibration of multiple projectors on most common swept surfaces using a single uncalibrated camera. In order to constrain the system sufficiently, we assume that the camera is a linear device with no radial distortion. However, our projectors need not be linear devices. Further, we assume that the path and profile curves are planar and during the sweeping the profile curve only rotates and translates without any scaling. Finally, we assume that the user provides a reasonable estimate of the rotation angle of the profile curve.

The algorithm has five main steps illustrated in Figure 2 with their input and outputs. These steps are described briefly here:

- *B.Sajadi and A. Majumder are with the Department of Computer Science, University of California, Irvine, Irvine, CA, 92697.*
E-mail: bsajadi@uci.edu, majumder@ics.uci.edu

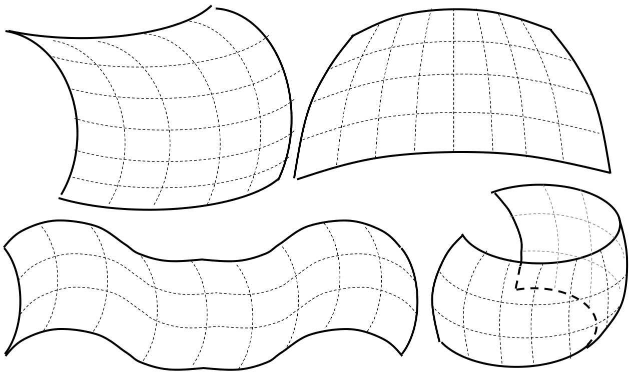

Fig. 1. Examples of swept surface: a cylinder-like swept surface where the extrusion happens along a curve instead of a straight line with no self occlusion (top left) and with self-occlusion (bottom right); a partial dome truncated along a plane parallel to the great circle (top right); and a general swept surface (bottom left).

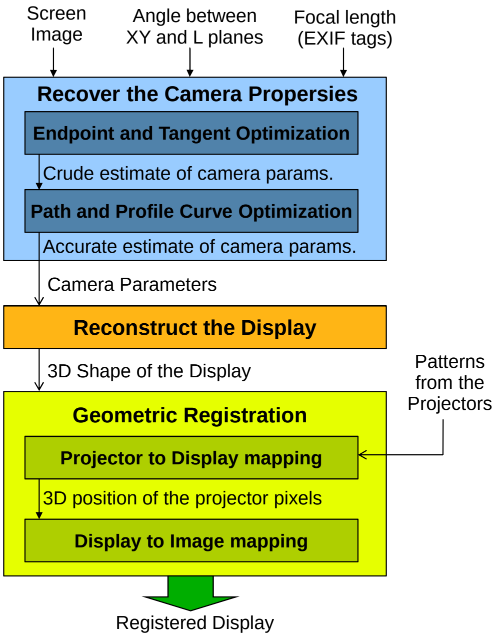
Screen Image
Angle between XY and L planes
Focal length (EXIF tags)
Recover the Camera Propersies
Endpoint and Tangent Optimization
Crude estimate of camera params.
Path and Profile Curve Optimization
Accurate estimate of camera params.
Camera Parameters
Reconstruct the Display
3D Shape of the Display
Patterns from the Projectors
Geometric Registration
Projector to Display mapping
3D position of the projector pixels
Display to Image mapping
Registered Display

Fig. 2. The figure shows the pipeline of our algorithm together with the input and output of each step.

- We use a non-linear optimization that is based on several constraints on image of the corners of the swept surface and the normals of the path and profile curves yielding a crude estimate of the display shape and the camera parameters (Section 3.1.1).
- We then use a more sophisticated non-linear optimization additionally constrained by the shape of the path and profile curves resulting in a more accurate estimation of the camera parameters (Section 3.1.2). This optimization is initialized using the crude estimate from the first optimization for faster convergence.
- Next, we use the recovered camera properties to extract the 3D shape of the display (Section 3.2).
- Following that we project a few patterns from the projectors to find the relationship between the projector coordinates and the camera coordinates and use the 3D display shape to convert them to the display coordinates (Section 3.3.1). We represent this using a rational Bezier patch for each projector.
- Finally, we use the projector to display mapping to find the proper image to be displayed by the projector. We can correct the imagery for the arbitrary viewpoint of a moving user (Section 3.3.2) or wallpaper the image for multiuser applications (Section 3.3.3).

When the display is large without enough space around it and therefore cannot be captured in a single camera image, we use multiple pan and tilted camera views to recover the camera parameters for each view and the display shape. This also allows us to register the display even when the variation in the tangent of the path and profile curves are greater than 180 degrees and the entire display cannot be covered by a single camera view. The main advantages of our method are as follows:

1) We can recover the display shape using a *single uncalibrated* camera without using any markers.
2) Since we use rational Bezier patches to relate the projector coordinates with the display coordinates, we can handle *distorted projectors*, common in compact setups where short throw lenses are mounted on the projectors. Further, this allows us to recover the mapping function with a sparse set of correspondences. Therefore, we can use a *low resolution camera* to register a much higher resolution display.
3) Our method can handle large and surround displays, which cannot be seen by a single camera view, by allowing the use of *multiple pan and tilted camera views* for the registration. This makes the method scalable to displays of any size and resolution.

By providing an easy automatic way to calibrate swept surfaces, our work has the potential to popularize swept surfaces for immersive virtual reality and visualization applications. For 4 or 5 wall CAVEs, we provide the first auto-calibration technique that does not treat them as multiple planar surfaces, but as single swept surfaces. Hence, we can allow the projectors to overlap across the corners and still register the display automatically.

2 Related Work

There has been a large amount of work on registering images on planar multi-projector displays using linear homographies enabled by the planar screens [11], [12], [13], [14], [15], [16], [17], even in the presence of projector non-linearities using rational Bezier patches [18].

Raskar et al. in [1] achieved multi-projector registration on a non-planar display using special fiducials and a large number of structured light patterns for a complete device (camera and projector) calibration and 3D reconstruction of the display surfaces, which are then used to achieve the registration. Aliaga et al. in [3], [2] use a similar 3D reconstruction method to achieve registration on complex

3D shapes, but without using any physical fiducials. To constrain the system sufficiently, this method uses completely superimposed projectors and validates the results from photometric and geometric stereo, resulting in a self-calibrating system. Raskar et al. in [4] use a stereo camera pair to reconstruct special non-planar surfaces called quadric surfaces (spheres, cylinders, ellipsoids, etc) and propose conformal mapping and quadric transfer to minimize pixel stretching of the projected images. Other methods used a stereo camera pair to achieve registration on more general non-planar surfaces like the corner of a room [5], [6].

More recently, Harville et al. [7] and Sun et al. [8] proposed calibration techniques for cylindrical surfaces that do not reconstruct the 3D shape of the display, but find only a 2D display parametrization in the camera space. This allows a wallpapered registration on the display by relating a piecewise linear representation of the projector coordinates with a piecewise linear 2D parametrization of the display in the common camera coordinates. However, to find the 2D parametrization, these methods need precise correspondences between the physical display and the observing camera. This is achieved by pasting a precisely calibrated physical pattern on the top and bottom rim of the cylinder. Further, the insufficient sampling in the interior of the display surface results in distortions or stretching in those regions. Finally, since the 3D shape of the display is not recovered, view-dependent registrations is not possible.

Our work is closest to a body of work on smooth vertically extruded display surfaces [9], [10], [19]. We showed in [9] that the stereo reconstruction is not always necessary when dealing with non-planar surfaces. Using the prior of vertical extrusion and a known aspect ratio it is possible to reconstruct the 3D display shape using a single uncalibrated camera. Consequently, the multiple projectors can be registered on this surface either in a wallpapered fashion or to be correct for any arbitrary viewpoint. Further, registration can be achieved even in the presence of nonlinear distortions in the projectors, common when using short-throw lenses to achieve a compact setup. Our method is similar only in essence to this earlier work, but since the class of surfaces we handle is more general and complex, our optimizations use constraints provided by the nature of a swept surface and are entirely different than those provided by vertically extruded surfaces. Further, unlike vertically extruded surfaces, many swept surfaces do not lend themselves to an easy 2D parameterization. We handle this by providing conformal mapping based parameterization for wallpapering.

3 Single-View Algorithm

A general swept surface is generated by moving a profile curve, $P(t)$, along the path curve, $Q(s)$. The profile curve can be rotated in the process. More precisely, for each s in the domain of the path curve $Q(s)$, the profile curve $P(t)$ is moved to the point $Q(s)$, possibly with a rotation. As mentioned in the introduction, we make some practical assumptions on this general definition. We assume that both $Q(s)$ and $P(t)$ are planar and the plane of the profile curve $P(t)$ is always perpendicular to the tangent of the path curve $Q(s)$. We also assume that the profile curve is only rotated but not scaled during the sweep. Finally, we assume that both $Q(s)$ and $P(t)$ do not have loops.

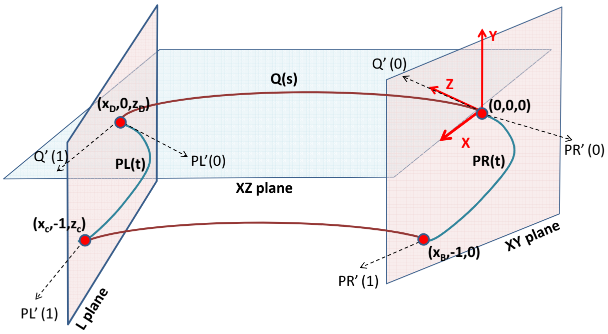

Fig. 3. The 3D setup of the swept surface is illustrated here.

Let us assume the path curve $Q(s)$ lies on the XZ plane and is flanked by two planar profile curves on the two sides. We call the left one $PL(t)$ and the right one $PR(t)$. Without loss of generality, we assume that $PR(t)$ lies on the XY plane. Therefore, PL lies on the plane L given by the rotation of the XY plane about the Y axis and a translation. We assume the two endpoints of each curve correspond to parametric values 0 and 1. We represent the tangent of Q at parametric value s by $Q'(s)$ and we use similar notations for the other curves. We assume the origin is at $Q(0)$, which is the same as $PR(0)$. We also assume that the vertical distance between $PR(0)$ and $PR(1)$ is 1. Similarly the vertical distance between $PL(0)$ and $PL(1)$ is 1. These assumptions are represented graphically in Figure 3.

We assume that N projectors are casually arranged to project on the swept surface S. We denote the 3D display coordinates by (X_s, Y_s, Z_s), the projector coordinates with (x, y), and the camera coordinates with (u, v). As mentioned in the introduction section, we assume that the camera is a linear device with no radial distortion; however, our projectors need not be linear devices. Finally, we assume that the user provides a reasonable estimate of the angle between the XY and L planes. In the rest of the section we describe the steps of our algorithm based on these definitions.

3.1 Recovering the Camera Properties

We first use a single image of the display surface (Figure 4) to recover the intrinsic and extrinsic parameters of the observing uncalibrated camera using a non-linear optimization. In most cameras it is common to have the principal center at the center of the image, no skew between the image axes and square pixels. Similar to Snavely et al.'s work [20], using these assumptions, we express the intrinsic

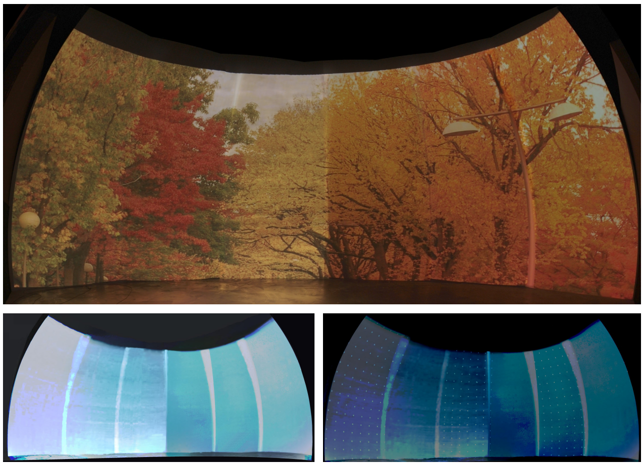

Fig. 4. Top: A truncated dome of 30 ft (9.15 m) radius, 26 ft (7.92 m) height, and 160 degrees angle subtended horizontally with 6 projectors registered to be correct for an arbitrary viewpoint. Bottom: Two of the input images to our algorithm for the truncated dome. Bottom-Left: Single image of the screen. Bottom-Right: Three projectors that do not overlap with each other are projecting blobs which are then captured by the camera. This allows data collection in parallel from multiple projectors. Please zoom in to see blobs.

parameter matrix of a camera, K_c, as

$$K_c = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

The camera calibration matrix that relates the 3D coordinates with the 2D camera image coordinates, (u,v), is given by $M = K_c[R|RT]$ where R and T are the rotation and translation of the camera with respect to the world coordinate system. In this step, we use the geometric constraints of a swept surface to setup a non-linear optimization that can estimate seven parameters of the camera calibration matrix. These include the focal length f, the three rotations that comprise R and the three coordinates of the center of projection of the camera T.

Our non-linear optimization has two phases. In the first phase, *endpoint and tangent constrained optimization* (Section 3.1.1), the seven camera parameters are estimated using just the projection of the endpoints and the tangent vectors of the path and profile curves on the camera image. These estimates are used to initialize the second phase, *path and profile curve constrained optimization* (Section 3.1.2), with a more expensive error function that uses constraints on the entire path and profile curves to refine the camera parameters.

3.1.1 Endpoint and Tangent Constrained Optimization

In order to define the constraints, we first detect the image of the three curves – $Q(s)$, $PR(t)$ and $PL(t)$ – in the single image of the display surface seen by the camera using standard image segmentation and edge detection techniques. For the sake of simplicity, we represent all the 3D curves by upper-case letters and their images by lower-case letters.

Our next step is to back-project these curves, which are on the image plane of the camera, in 3D space using the camera calibration matrix M. We back-project the image of the right profile curve, i.e. pr, on the the XY plane and we denote it by PR_b. For this we intersect an imaginary ray going from the current center of projection (COP) of the camera through the pixels of the curve on the image plane and then intersecting the ray with the proper plane (XY plane in this case). Similarly, we find the back-projection of the image of the path curve, q, on the XZ plane and we denote it by Q_b. Next we find the plane L by rotating the XY plane about the Y-axis by the angle between $Q'_b(0)$ and $Q'_b(1)$, the tangents of Q_b at its two ends, and then translating it by the distance between $Q_b(0)$ and $Q_b(1)$, the endpoints of Q_b. Then we back-project the image of the left profile curve, pl, on this plane, L, and we denote it by PL_b. Note that ideally, after convergence of the optimization, PR_b, PL_b, and Q_b should coincide with PR, PL, and Q. Based on this we setup the following constraints:

1. Based on the definition of the coordinate system, the distance between the Y coordinates of $PR_b(0)$ and $PR_b(1)$ should be equal to 1 unit. Hence, the difference of this distance from 1, denoted by e_1, should be zero.
2. Similarly, the distance between the Y-coordinates of $PL_b(0)$ and $PR_b(0)$ should be 1. Hence, the difference of this from 1, denoted by e_2, should be zero.
3 & 4. Based on the definition of the coordinate system $PR_b(0)$ should be at the origin $(0,0,0)$. Thus, the distance between $PR_b(0)$ along the X and Y axis, denoted by e_3 and e_4, should be zero.
5. Since PR_b and PL_b are supposed to have similar 3D shapes, the euclidian distance between $PR_b(0)$ and $PR_b(1)$ should be equal to the euclidian distance between $PL_b(0)$ and $PL_b(1)$. Hence, the difference between these two distances, denoted by e_5, should be zero.
6. Based on the definition of the coordinate system, $Q'_b(0)$, which should be perpendicular to the plane of PR, should coincide with the Z axis. Therefore the dot product of the two vectors, denoted by e_6, should be zero.
7. Finally, since PR_b and PL_b are supposed to have similar 3D shapes, the angle between $PR'_b(0)$ and $PR'_b(1)$ should be equal to the angle between $PL'_b(0)$ and $PL'_b(1)$. Hence, the difference between them, denoted by e_7, should be zero.

In order to find the tangents vectors we back-project the first few pixels of the image of the curve on the corresponding plane, e.g. XZ plane for E'. Then we fit a line to these points that approximate the direction of the tangent vector. These tangent vectors are illustrated in Figure 5 for the truncated dome.

The above seven constraints are sufficient to solve for the seven unknown camera parameters. Each of the above provide different types of constraints. e_1 and e_2 constrain the size of the display, and hence serve as *scale constraints*. e_3 and e_4 are *positional constraints* and $e_5 \ldots e_7$ serve as *shape constraints*. To keep the scale of the distances

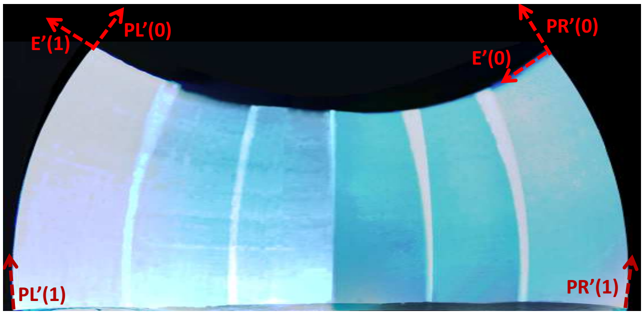

Fig. 5. This figure illustrates the tangents of the surface at the endpoints for the truncated dome.

and angles similar, we express the angles in radians. Our error metric in this first phase of the optimization, denoted by e_f is the root mean square of the weighted error function e_i, $1 \leq i \leq 7$. Formally, we seek to minimize $e_f = \sqrt{\sum_{i=1}^{7}(w_i e_i)^2}$. Note that usually the scale constraints are much more important than shape constraints to guide the solution towards the correct size of the display. Also, the shape and positional constraints are equally important since deviation from any one would not preserve the shape and position of the display. Using these guidelines, we design our weights such that $w_3 = w_4 = w_5 = w_6 = w_7 = 1$ and $w_1 = w_2 = 4$.

Similar to Snavely et al.'s work [20], we use the focal length obtained from the EXIF tags of the captured image to initialize the intrinsic matrix in our non-linear optimization. For initialization of the camera position and orientation, the user needs to provide an estimate of the angle α between the XY and L planes. We initialize the camera position to have a Y coordinate which is roughly at the center of the height of the screen -0.5. To initialize the Z coordinate (how much in front of the screen the camera is), we use some simple image processing to find the width W and height H of the display in the camera image. Note that an estimate of the field of view covered by the screen in the vertical direction is given by $\frac{H}{f}$. From this we can find how much in front of the screen the camera should be placed to achieve this for a unit height screen by $-\frac{1}{2}\cot\frac{H}{2f}$. Finally, to initialize the X coordinate to be in the middle of the length of the screen, we use $\frac{W}{2H}$. Thus, the initialization for the camera position is $(\frac{W}{2H}, -0.5, -\frac{1}{2}\cot\frac{H}{2f})$. The orientation of the camera is computed by rotating the Z axis by $\frac{\alpha}{2}$ about the Y axis.

3.1.2 Path and Profile Curves Constrained Optimization

The seven estimated camera parameters in the previous step are used to initialize the path and profile curve constrained optimization step that attempts to refine these estimates further. For this, we add another error metric e_s to the error metric e_f in the previous step and use a non-linear optimization method that minimizes the error $e = e_f + w_s e_s$. The error metric e_s provides an estimate of the difference in the shape of the left and right profile curves of the display in 3D, and w_s is the associated weight.

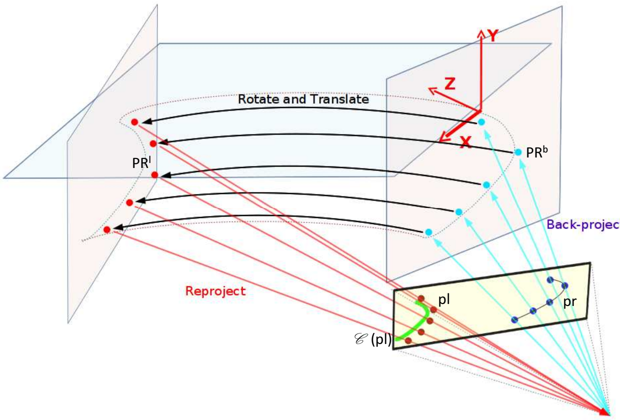

Fig. 6. This figure illustrates our path and profile constrained optimizations to recover the camera and display properties.

The swept surface is constrained by the fact that the points on the left profile $PL(t)$ when translated by the distance between the endpoints of $Q(s)$ and rotated by the angle between the tangent vector at the endpoints of $Q(s)$ should coincide with the right profile curve $PR(t)$. We use the deviation from this constraint to define the error e_s. Please refer to Figure 6 for the following explanation. We first sample the curve pl in the camera space and fit a parametric curve $\mathscr{C}(pl)$ (green curve) to these samples. Next we sample the pr in the camera image (dark blue points) and back-project them on the XY plane in 3D to get samples on PR_b (cyan points). Next we rotate the samples on PR_b (cyan points) by the angle between $Q'_b(0)$ and $Q'_b(1)$ and translate them by the distance between $Q_b(0)$ and $Q_b(1)$. These rotated and translated samples (orange points) are then reprojected on the camera image plane (red points). The distance of these reprojected points from $\mathscr{C}(pl)$ (green curve) provides us the error function e_s. At convergence, these points should lie on $\mathscr{C}(pl)$.

To solve both the optimizations, we use standard gradient descent methods. To assure faster convergence we (a) apply a pre-conditioning to the variables to normalize the range of the values assigned to them; and (b) use decaying step size.

3.2 Recovering the Display Properties

The path and profile curves constrained optimization provides us with a robust estimate of the camera calibration matrix M. In this step, we use the estimated M to find the 3D shape of the swept surface. We represent the 3D swept surface by a dense set of 3D point samples lying on the surface. To generate a sampling of the 3D swept surface, we first find the 3D samples on the back-projected path curve, Q_b. We generate an estimate of the local tangent at each of these samples by considering their immediate neighborhood. Then, we generate 3D samples on PL_b. We translate this set of 3D samples to place it at every sampled 3D point on Q_b and then rotate it by the amount the tangent changes from one sample on Q_b to the adjacent one. Thus, we generate a dense sampling of the 3D swept surface using a method similar to its construction – by sweeping the sampled profile curve along the sampled path curve. Note that since we sample the image of PL uniformly in the camera space and back-project it in 3D, the samples are not uniformly placed on the 3D curves. Hence, the 3D points generated to sample the display are dense but non-uniform. We call this set of 3D points as S_C where C stands for cartesian coordinates.

3.3 Geometric Registration

In the geometric registration step we first need to map the projector coordinates to the display coordinates (Section 3.3.1). This mapping can then be used to relate the image coordinates to the projector coordinates such that the image is corrected for the arbitrary viewpoint of a moving user (Section 3.3.2) or such that the image is wallpapered on the display (Section 3.3.3), proper for multiuser applications.

3.3.1 Finding the Projector to Display Mapping

In the geometric registration step, we first define for each projector, a function $M_{D\leftarrow P}$ that maps the projector coordinates (x,y) to the 3D display coordinates (X,Y,Z) via the camera coordinates (u,v). Mathematically, $(X,Y,Z) = M_{D\leftarrow P}(x,y)$. We use three rational Bezier patches to represent $M_{D\leftarrow P}$. Hence,

$$(X,Y,Z) = (B_X(x,y), B_Y(x,y), B_Z(x,y)). \quad (2)$$

To find B_X, B_Y and B_Z, we find correspondences between (x,y) and (X,Y,Z). We project a set of blobs from each projector and capture them using the camera. Back-projecting the camera blob centers using the estimated M and intersecting with S provides us the corresponding 3D coordinates. We find the intersection using the set of 3D samples S_C, representing the display, as follows.

First, we represent every point in S_C using angular coordinates centered at the center of projection (COP) of the recovered camera. Thus, each 3D point in S_C now has an alternate representation using (θ,ϕ,d) where (θ,ϕ) provides the angular coordinates and d is the distance at which the ray from the COP of the camera at an angle (θ,ϕ) meets the display surface.

To find the 3D display coordinates corresponding to each blob center (x,y) in the projector space, we first find the angular representation (θ,ϕ) of its corresponding point in the camera space. We interpolate the value of d for this coordinate by choosing the k-nearest neighbors of this ray in S_C and then fitting a smooth interpolating surface through these points in the (θ,ϕ) space. Finally we convert the interpolated points to cartesian coordinates to find the corresponding 3D points in the (X,Y,Z) space.

To find the Bezier patches B_X, B_Y and B_Z, we fit a rational Bezier patch to these correspondences using a non-linear least square fitting solved efficiently by the Levenberg-Marquardt gradient descent optimization technique. Rational Bezier is perspective projection invariant. Hence, it can adequately represent the combination of the non-linear distortion due to the shape of the swept surface and the perspective projection of the projector.

3.3.2 Registering for an Arbitrary View Point

$M_{D\leftarrow P}$ allows us to correspond every projector pixel to a 3D display coordinate. However, when putting up an image on the display, we need to define how an image coordinates (s_i,t_i) is associated to the 3D display. Essentially, how is the image mapped on the display. This is application dependent. For example, for a single user head-tracked VR application, one would like to render an image of a 3D scene for a virtual arbitrary viewpoint (completely unrelated to the location of the camera used for calibration) and then projectively texture the image for this virtual arbitrary viewpoint on the display surface. As the user moves, the arbitrary viewpoint will change and so will the projective texture on the 3D display surface. In this case, to find the image coordinate (s_i,t_i) associated with each projector pixel (x,y), we first use Equation 2 to find the corresponding 3D point (X,Y,Z) on the display. Then we project this 3D point on a virtual camera at an arbitrary viewpoint to find the image coordinates (s_i,t_i) on the image plane of this camera. Now, for every projector pixel, we can pick the color from the corresponding (s_i,t_i) to create the image to be projected from this projector to create a registered display. This approach is similar to the 2-pass rendering used by Raskar et al. in [21].

3.3.3 Registering a Wallpapered Image

Function $M_{D\leftarrow P}$ provides a mapping between the projector pixels (x,y) and the 3D display coordinates (X,Y,Z). For wallpapering, we need to associate 2D coordinates obtained by the 2D display parametrization at (X,Y,Z). Instead of repeating this process for every projector pixel, we sample the projector pixels and define (s,t) at the corresponding 3D display coordinates. Then we use Bezier functions, similar to Equation 2, to interpolate the (s,t) coordinates at the other projector pixels.

$$(s,t) = (B_s(x,y), B_t(x,y)). \quad (3)$$

Since high degree Bezier fitting tends to be slow and unstable, to handle severe non-linearities one can use Bezier splines composed of cubic Bzier patches instead. However, in all our experiments we achieved desirable results with a single Bezier patch.

The challenge in wallpapering lies in parametrizing the display to associate a (s,t) parameter with any 3D point (X,Y,Z). A general way to achieve wallpapering, is to use conformal mapping of the image on to the 3D display surface. We use [22] to map an image in an angle-preserving (i.e. conformal) manner on the display mesh. This method considers a 3D mesh of the display surface and associates an image coordinate at every vertex of the mesh. Conformal

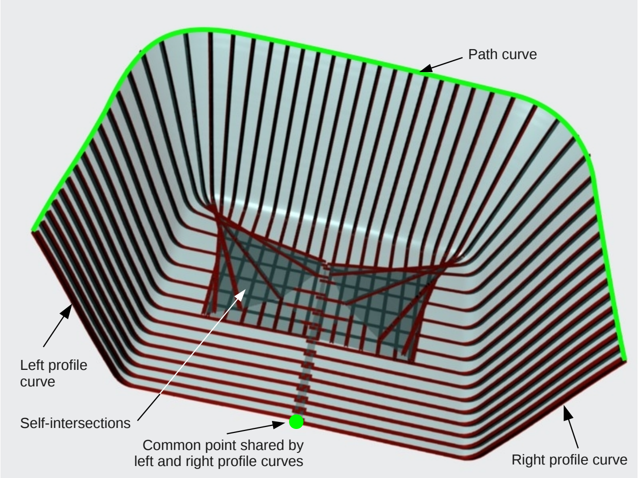

Fig. 7. This figure shows the real self-intersecting bowl display. Note that the different instances of the profile curve as it is being swept on the path curve intersect. Hence, a point in a display can have non-unique (s,t) parametrization. Further, the left and right profile curves share a common point in this display.

mapping trims the image to wallpaper it on the surface while preserving the angles. Therefore there is no guarantee that the final trimmed image is rectangular. Alternatively, if the surface is simple with no self-intersections, we can use the inherent 2D parametrization provided by the path and profile curves.

Note that wallpapering, be it using the inherent parameterization of the surface or conformal parametrization, does not look correct from any single viewpoint. However, since we are used to seeing wallpapered images, the distortions are perceptually acceptable to us. Hence, this is a common way to accommodate multiple users.

4 Complex Surfaces

In this section we show that, with minor modifications, our method is applicable to more complex surfaces that have self-intersections (Section 4.1) or sharp corners (Section 4.2)

4.1 Self Intersecting Surfaces

Depending on the relative orientation of the profile curve with respect to the path curve, it may so happen that as the profile curve is moved on the path curve, the same 3D point is generated multiple times from different positions of the profile curve. The surface is then self-intersecting (Figure 7). Our method can handle such surfaces without any special considerations. However, since 2D parametrization is not well-defined on these surfaces, conformal mapping is required to wall paper an image on them.

Our method can also handle swept surfaces where the left and right profile curves share a common point (Figure 7). However, to separate the left and right curves in the image, we can use either a user input or one physical marker at the point where the profile curves meet. Following this, the two profile curves can be segmented in the camera image and rest of the algorithm can be applied unmodified.

4.2 Piecewise Planar Surfaces

The inherent assumption of our geometric registration algorithm is that the swept-surface display is smooth and therefore the 3D position of the projector pixels can be recovered using a sparse sampling of the pixel positions and represented by set of rational Bezier patches (Section 3.3.1). However, 5-wall CAVEs, though swept surfaces, do not fall into the category of smooth surfaces. In this section we extend our method to handle such surfaces with sharp edges or corners.

Even for such displays, when the projection area of a projector does not overlap across a sharp edge or corner, our algorithm can run unmodified. However, this creates a sharp discontinuity at the edge or corner which can be alleviated by overlapping projectors across the corners and edges of a CAVE. This kind of set-up has been avoided due to the complexity in registration it brings forth. But, our method can handle this case easily. For each projector, we first divide the set of blobs, captured by the camera, to multiple sets where all the blobs in each set appear on the same planar wall. For this, we need to divide the surface to multiple planar walls. We achieve this by dividing each of the path and profile curves to several line segments using common line segmentation techniques. Consequently, during the reconstruction of the surface, we find the normal vector of each point using the cross product of the direction of the line segment of the profile curve and the direction of the line segment of the path curve.

Next, for each set of the blobs we fit a separate set of rational Bezier patches similar to Section 3.3.1. We use each of the sets of patches to estimate the 3D location of all the projector pixels. Therefore, for each projector pixel, we estimate multiple 3D locations. However, the location is valid only when the estimated location and the set of blobs used for the estimation are on the same wall. Therefore, except for the projector pixels that are very close to the corners of the CAVE, there is only one valid estimated location. For the projector pixels that are close to the boundaries, we use a weighted average of all the valid locations to transit from one wall to another. The weights are proportional to the shortest distance of the pixel position from the boundaries of the walls to assure a seamless transition.

Fast Recalibration: Though the sharp corners reduce the implementation simplicity of the geometric registration process, they simplify the estimation of the camera parameters. For piecewise planar CAVEs, we first extract the line segments of the path and profile curves as mentioned before. Then, we use a one step optimization process to find the camera parameters. The constraints of this optimization include all the constraints used in Sections 3.1.1. In addition to these constraints, we also apply curve similarity constraints similar to Section 3.1.2 on the corner points of the two profile curves, i.e. PR and PL. Since both PR and PL are piecewise linear, the similarity in the corners assures the similarity over the entire curves. The error function for this optimization can be computed considerably faster compared to the error function used in Section 3.1.2 since it is only applied to a few corner points instead of the curves. This allows us to combine the two optimization steps to one simple and fast optimization and reduces the average optimization time from more than 4 minutes to 30 seconds. Such a reduction in the optimization time is particularly useful when the display surface or the position of the projectors changes and the system needs to be recalibrated promptly.

5 Multi-View Algorithm

The algorithm described so far assumes that the uncalibrated camera sees the entire display surface. This is often impossible for large displays. So, we design a method that can use multiple views of parts of the screen from the same uncalibrated camera to register the multiple projectors. For this we adapt one our previous methods presented in [23] that does the same for vertically extruded displays. Similar to this work, we assume that the camera captures Q views of the display, in each of which only a small part of the display is visible. The camera is either panned and titled, but not translated, or translated, but not pan or tilted to capture these views. We denote the camera views by V_i, $1 \leq i \leq Q$. The zoom of the camera is not changed across these views. This assures that the intrinsic parameter matrix K_c remains constant. We consider V_1 as the reference view and the extrinsic matrix of V_1 to be C_1. Since the camera is only rotated and not translated, the extrinsic matrix C_i of view V_i, $i \neq 1$, is related to C_1 by a rotation matrix R_i, i.e $C_i = R_i C_1$. We assume considerable overlap between adjacent views.

When handling multiple views, for each camera view a similar set of images as in Section 3 is captured but only for the projectors which are fully or partially visible in that view. We first recover the common intrinsic parameter matrix K_c using angular constraints on pairs of correspondences across multiple views, followed by the relative rotation matrix R_i which relates every C_i to C_1 by finding the minimal spanning tree in a homography graph formed by the homographies relating adjacent overlapping camera views. Finally, the camera calibration matrix C_1 for the reference view is extracted as follows.

To recover the pose and orientation of the reference view with respect to the 3D display, C_1, we extend the method presented in Sections 3.1.1 and 3.1.2, using only the images which do not have any projectors turned on. Similar to our previous work [23], both the back projection and the reprojection happen in the camera views in which the boundary feature or curve is detected. So, in the *endpoint and tangent constrained optimization*, we do all the back-projections from the respective camera views where they are detected using the matrix $K_c R_k C_1$. The error metric is similar to the one in Section 3.1.1 but is summed across the multiple relevant views and minimized to provide a rough estimate of C_1. In the *path and profile curve constrained optimization* stage all the images where the right profile curve is detected are used. However, after the translation and rotation of the points, unlike Section 3.1.2 where the 3D points of the estimated left profile curve are reprojected to the single camera view (red points in Figure 6), we reproject these points to *all* the camera views where a part of the left profile curve is detected.

Each reprojected points may appear within the field-of-view (FOV) of multiple overlapping views. The average of the distance of the reprojected point from the detected 2D left profile curve across all these views define our error for each point. We sum the errors across all the points to provide a reprojection error e_s. We seek to minimize the sum of e_s across all the camera views where the left profile curve is detected to find C_1.

Recovering display properties is similar to Section 3.2 except for when finding the back-projected 3D points on the path and profile curves, we use all the camera views that contain the path or profile curves. In the geometric registration step, a blob can be seen by multiple cameras resulting in multiple (θ, ϕ) estimates for the blob from different views. We take a weighted mean of all these values to find an accurate corresponding point (θ, ϕ) for the blob. The weight is proportional to the minimum distance of the detected blob from the edges of the captured view. The rest of the method is same as in Section 3.3.

6 Implementation and Results

We implemented our method using Matlab on three real displays (Figures 4 and 8). The first one is a truncated dome, 30 ft (9.15 m) in radius, 26 ft (7.92 m) in height and subtends 160 degrees horizontally. We calibrated six Panasonic 6000 series HD projectors arranged in a panoramic fashion on this display. The projectors are rotated 90 degrees to have larger vertical field of view (FOV). The second display is a more complex surface where the left and right profile curves share a common point and the surface self-intersects. This looks like a CAVE whose edges are smoothed out and we call this a *bowl*. This display is about 30 ft (9.15 m) wide, 13 ft (3.96 m) high, and 22 ft (6.71 m) deep. We use our algorithm to register four Digital Projection HD projectors on this display arranged in a CAVE like fashion. Finally, the third one is a miniature 5-wall CAVE, 3 ft (91 cm) wide, 2 ft (61 cm) high, and 2 ft (61 cm) deep. We used our extended method for piecewise planar surfaces (Section 4.2) to register four Epson Powerlite 1810 projectors on this display. Since these displays are too large to be captured in a single view of a standard camera, we illustrate these shapes with panorama images using spherical and rectilinear projection for the truncated dome and the bowl respectively. Further, to achieve color seamlessness for all of our displays, we used the color registration technique presented in [24].

We use a Canon Rebel xSi camera (around $800) for calibrating the display. To remove the brighter overlaps,

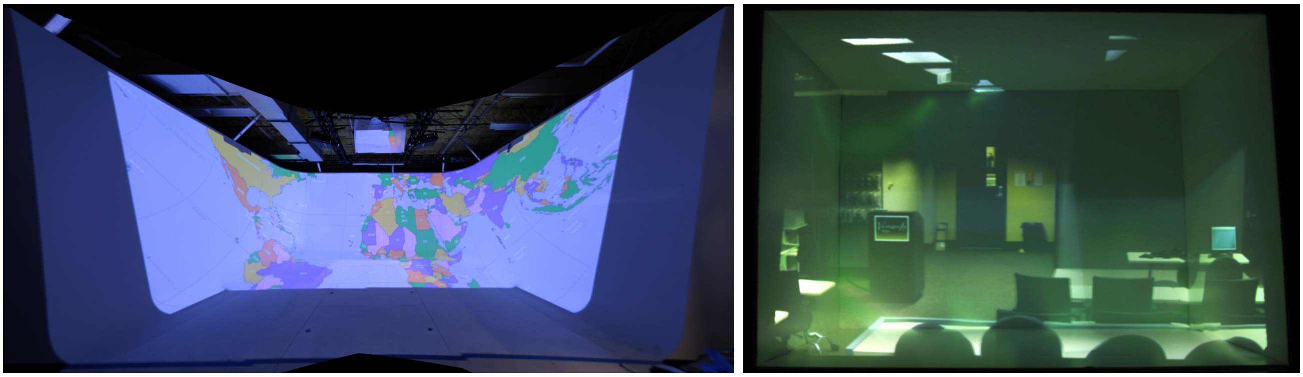

Fig. 8. This shows two registered CAVEs (swept surfaces). Left: A bowl shaped CAVE with smooth corners, about 30 ft (9.15 m) wide, 22' ft (7.92 m) deep, and 13 ft (3.96 m) high, with 4 projectors wallpapered using conformal mapping to visualize the world map. Right: A miniature 5-wall CAVE with sharp corners, about 3 ft (91 cm) wide, 2 ft (61 cm) deep, and 2 ft (61 cm) high, with 4 projectors registered and corrected for an arbitrary viewpoint to visualize the inside view of a presentation room. Note that the low contrast of the image and the visible corners of the CAVE are due to the sharp changes in the reflectance properties of the surface and severe inter-reflections inside of the CAVE. However, there is no geometric misalignment in the registered image.

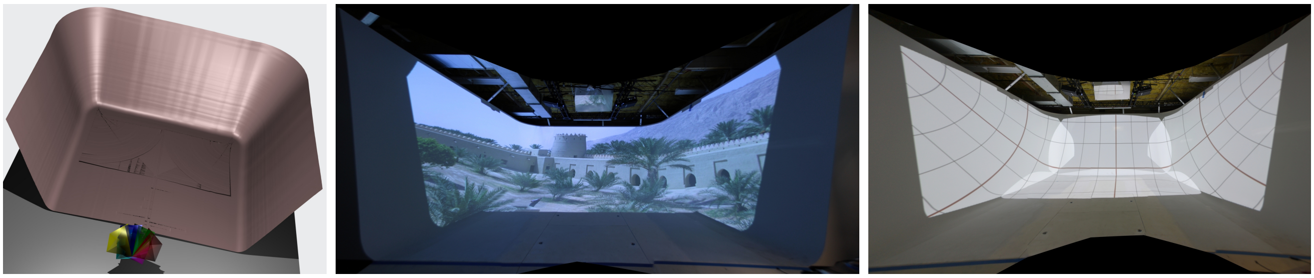

Fig. 9. Results on the real bowl display (30 ft (9.15 m) wide, 22 ft (6.7 m) deep and 13 ft (3.96 m) high). Left: The reconstructed 3D shape of the bowl display along with the six camera views used for this purpose; Middle: View-dependent registration of a castle – the capturing camera is at a different position than the arbitrary viewpoint and hence the distortions are perceived; Right: Registering a grid in a wallpapered manner using non-linear projectors – overlaps are not blended to show the distortion of the projectors. Please zoom in to see details.

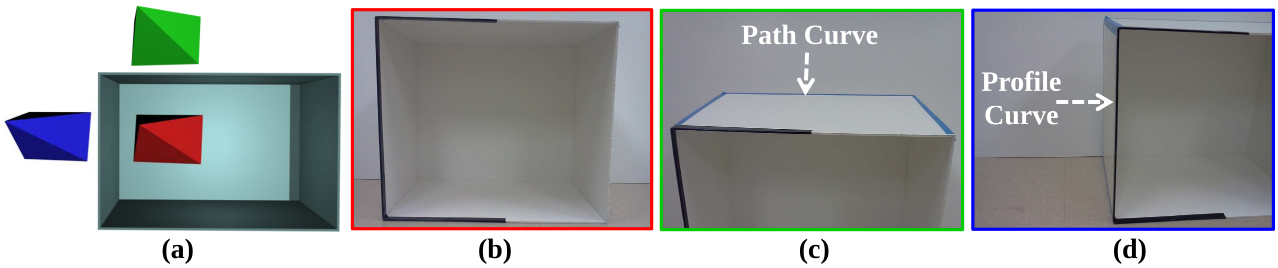

Fig. 10. (a) The reconstructed 3D shape of the real 5-wall CAVE (3 ft (91 cm) wide, 2 ft (61 cm) deep and 2 ft (61 cm) high) along with the three translated camera views used for this purpose; (b-d) The three input images used to reconstruct the display shape and find the camera views. The colors of the borders of the images correspond to the colors of the camera views shown in (a). (c) and (d) are used to segment the path and profile curves respectively.

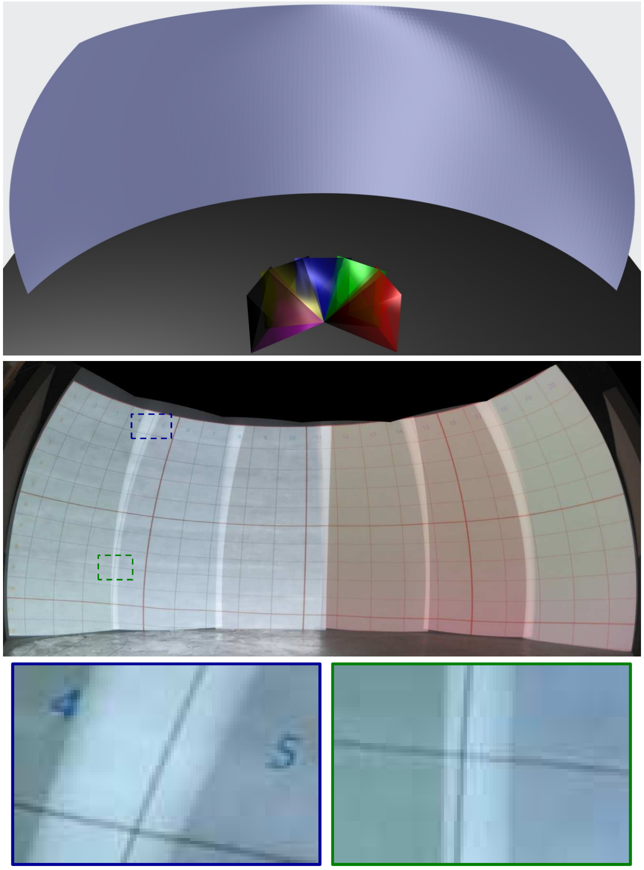

Fig. 11. Results on the real truncated dome. Top: The reconstructed 3D shape of the truncated quadrant of sphere along with the five camera views used for calibration. Since the five views are only panned, the homography graph is a straight line. Middle: The display made of six projectors is wallpapered using the natural curve based parametrization. The overlaps are not blended to show the different projectors. Bottom: Zoomed in views of the registration. Please note the accuracy of the registration in the overlap region. The zoomed-in portions are marked on the image.

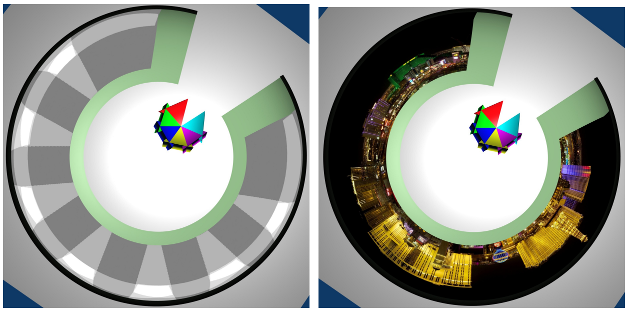

Fig. 12. Results from simulation: Left: The 3D reconstructed display and the six camera views for the truncated ellipsoid display with 2×8 array of sixteen projectors. Since the camera is only panned, the homography graph and tree are both the same and linear. Right: An image registered in a wallpapered fashion using the natural curve based parametrization of the display. Please zoom in to see details.

we use simple edge blending techniques [21]. To find the projector to camera correspondences, we capture a rectangular grid of Gaussian blobs with known projector coordinates displayed by the projector. We binary-encode the blobs and project them in a time sequential manner to recover the exact IDs of the detected blobs and find the correspondences [1], [14] (Figure 4).

The degree of the rational Bezier patches B_X, B_Y and B_Z and the number of blobs used depend on the amount of non-linearities present due to the screen curvature and the distortions in the projectors. Our truncated sphere quadrant shows smooth variation in curvature and hence a rational Bezier of degree 5 in both dimensions and a grid of $16 \times 8 = 128$ blobs for each projector was sufficient. However, the bowl-shaped display and the 5-wall CAVE show sharp changes in the curvature near the corners and hence require degree 7 rational Bezier patches and a denser grid of $32 \times 16 = 512$ blobs for each projector.

The offline registration takes about five minutes. The image correction can be implemented in real-time, similar to our previous work [9], using GPUs through Chromium - an open-source distributed rendering engine for PC clusters [25]. The coordinate-mappings of all pixels of the projector should be first precomputed. This per-pixel projector to screen lookup table can then be used by a fragment shader to map pixels from the projector coordinate space to the screen coordinate space during rendering.

For the truncated quadrant of the sphere and the bowl display, we use five and six pan and tilted camera views respectively. For the 5-wall CAVE we use three translated camera views, two of which are used to capture the path and profile curves from the outside of the CAVE. The camera views and the reconstructed 3D display shapes for these three displays are shown in Figures 9, 10, and 11.

To demonstrate our results for displays with more projectors, we show two complex shapes in simulation. The first is a truncated ellipsoid subtending an angle of 300 and 100 degrees in the horizontal and vertical direction (similar to the surface in the bottom right of Figure 1) lighted by a 2×8 array of sixteen projectors captured by six panned camera views (Figure 12). The second is a large panoramic shape similar to the swept surface in Figure 1(bottom) lighted by a 4×10 array of forty projectors captured by seven panned and tilted camera views (Figure 13).

In Figures 9 and 16 we show two examples of registration for an arbitrary viewpoint, as is common in 3D VR environments. The accompanying video illustrates better the distortions that appear when the viewer deviates from the ideal viewpoint. We also register the bowl display in a wallpapered manner. Since this is a self-intersecting surface, we use a conformal map based wallpapering (Figures 8 and 9). Note that when using conformal mapping, the constraints imposed do not allow us to map the entire image on the display, but clip off some regions near the boundary. In Figures 12 and 13 we show wallpapered registration using the natural curve based parametrization of the display. Finally, we demonstrate the ability of our method to register images in the face of non-linear distortions for the complex bowl display in Figure 9. Our projectors have relatively

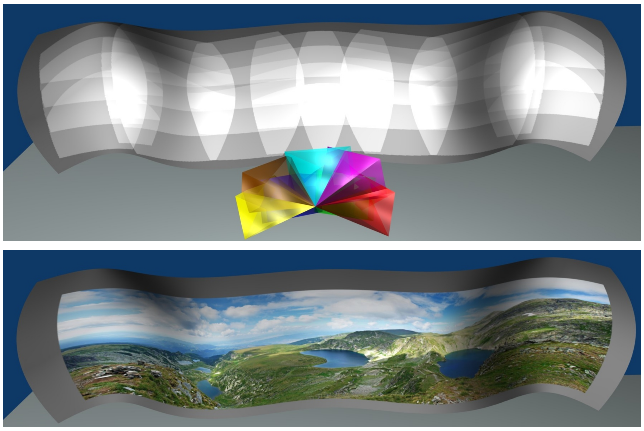

Fig. 13. Results from simulation: Top: The 3D reconstructed display and the seven camera views for the wavy panoramic display lighted with 4×10 array of 40 projectors. Bottom: An image registered in a wallpapered fashion using the natural curve based parametrization of the display. Please zoom in to see results.

large throw-ratios and hence little lens distortions. So, we chose to simulate the distortion digitally by distorting the input images to the projectors. Note that our results use particularly challenging contents like lines and texts to demonstrate accuracy. For better zoom in and fly-through experience of VR applications on our immersive swept surfaces we strongly encourage the readers to see the video.

7 Discussion

Degenerate Cases: Our method achieves markerless registration on swept surfaces using an uncalibrated camera. Even in the presence of the priors on the display surface, there is a set of camera positions and display shapes that lead to degenerate cases for one or both phases of our optimization. First, the camera should not be placed at a location from which any of the path or profile curves would be projected as a line. This happens if the normal to the image plane lies on the XZ plane, $XZplane$, or L plane where Q, PR, or PL will be projected as a line respectively. Second, if the extrusion curve is symmetric about a plane, which is halfway between XY plane and L plane, placing the camera on this plane will result in an ambiguity between the focal length and the depth of the screen from the camera. Hence, all these camera placements should be avoided. Finally, when the profile curve is a line, e_7 will be always zero and also the similarity of the 3D shape of the left and right profile curves becomes a subset of the error metrics in the first optimization. Therefore the problem cannot be constrained sufficiently. Such surfaces can be modeled as vertically extruded surfaces and they can be calibrated using our earlier work [9], [10], [23].

Accuracy and Sensitivity: Two questions are imminent for our method. (a) How accurate are the estimated camera and display parameters?; and (b) how sensitive is the geometric registration to the inaccuracies of these estimates or the priors imposed on the display surface? It is difficult to analyze all the above issues in real systems, hence we have conducted extensive analysis in simulation and real systems (whenever possible) to answer these questions.

TABLE 1
Percentage Errors of the estimated camera and display parameters over a large number of simulations with different device and display configurations.

Parameter	Max	Mean	Std
Camera Orientation (deg)	0.637	0.231	0.195
Camera Position (%)	0.592	0.213	0.189
Focal Length (%)	3.571	1.932	0.891
Image Distortion (%)	0.793	0.341	0.228

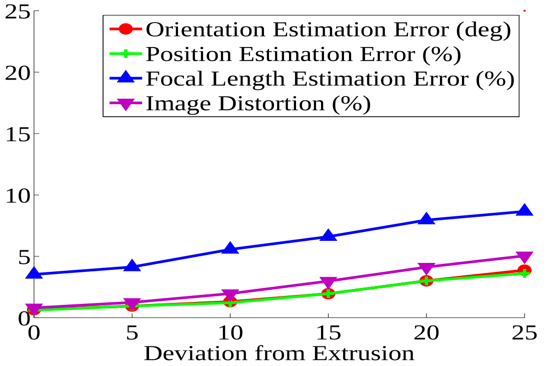

Fig. 14. The error in camera parameters and the distortion of the image when viewed with 45° from the view of the calibrating camera with respect to the deviation of the surface from being a swept surface. Note that even for a large deviation, we can estimate the camera parameters robustly and therefore limit the distortions.

First, we study the accuracy of the estimated camera parameters following our non-linear optimization process. We perform an error analysis by simulating many different camera and display parameters to provide the deviation of the estimated parameters from the actual parameters. For the orientation of the camera, we provide the deviation

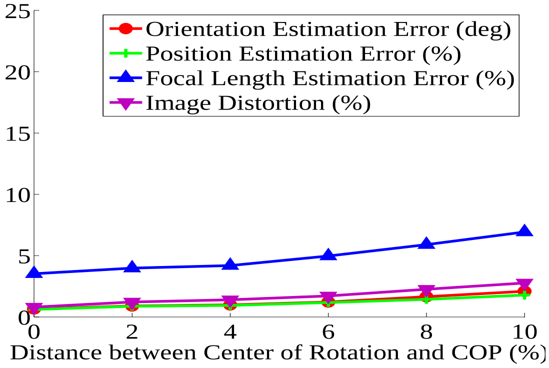

Fig. 15. The error in camera parameters and the distortion of the image when viewed with 45° from the view of the calibrating camera with respect to the distance of the center of rotation of the pan-and-tilt unit and the COP of the camera. All the experiments are performed with 5 camera views. Note that our method can estimate the camera parameters robustly for small distances common in pan-and-tilt units and therefore we can limit the distortions.

in degrees from the actual orientation. For translation, we provide the ratio of the error in estimation with the distance from the screen. Finally, we also study the effect of these inaccuracies on the distortion of the image when viewed from a different viewpoint than the view of the calibrating camera. For this we simulate the view from several viewpoints such that the view direction makes a 45° angle with the direction of the line connecting the COP of the calibrating camera to the center of the screen. On the image plane of the virtual camera, we measure the maximum deviation between the expected and actual location of several projector pixels. Then, we report the maximum deviation divided by the horizontal resolution of the virtual camera. These statistics are reported in Table 1. Please note that in all of our simulations we initialize the optimization with at least 10% error for each of the optimized parameters.

Even though the image may look slightly distorted when viewed from different viewpoints, the use of rational Bezier patches in our method eliminates any geometric misregistration. This is due to the fact that the overlapping pixels from the multiple projectors will still map to the same display coordinates regardless of the viewpoint.

Our registration can also handle small deviation of the surface from a perfect swept surface due to imprecision in the display surface or errors in the display shape estimation. We quantitatively evaluate the effect of deviation of a surface from being a swept surface on the accuracy of the estimated camera parameters in Figure 14. In order to measure the deviation from being a swept surface, we first rotate and translate the right profile curve along the path curve. Then, we measure the maximum distance between this curve and the left profile curve divided by the length of the left profile curve. This plot shows that even in the presence of large deviation of the screen from being a swept surface, our method can estimate camera parameters reasonably well and the image distortion is not very large. Further, our CAVE-like display (Figure 10) is an example of an imperfect hand-made swept surface. However, in Figure 16(top) we demonstrate how our method can register the display from a different viewpoint without any significant distortion. This is particularly apparent from the straight lines in the image that go over the sharp corners of the display with only a minor distortion.

Finally, we analyzed the effect of having a small translation while pan-and-tilting the camera, which is common when using an uncalibrated pan-and-tilt unit. In Figure 15 we measure the accuracy of our method with respect to the distance of the center of rotation of the pan-and-tilt unit from the COP of the camera. The distance is measured with respect to the distance of the camera from the center of the screen. We also calibrated the pan-and-tilt unit used in our experiments and figured out there is a 2.5” distance between the center of rotation and COP of the camera, which translates to around 0.6% for the truncated dome, 0.7% for the bowl-shaped display, and 4% for the CAVE-like display. Figure 15 shows that for such small translations our method can still provide acceptable results.

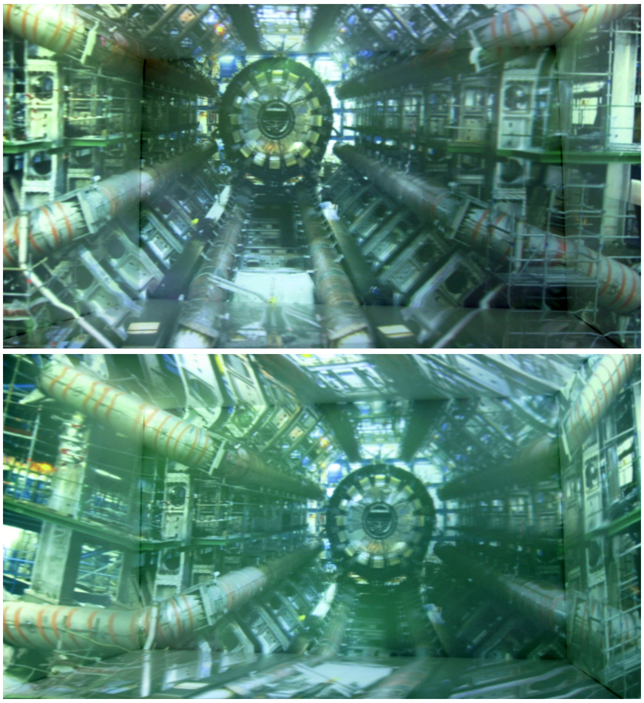

Fig. 16. Results on the real 5-wall CAVE. Top: An image from the inside of a space shuttle registered using view-dependent registration; the image is corrected for the viewpoint of the camera that is used to capture this photo which is not collocated with the calibrating camera. The visible seams on the corners are because of the view-dependent reflectance properties of our display but not any geometric misalignment. Bottom: The same registration when the image is corrected for the calibrating camera view (shown in red in Figure 10) but the capturing camera is at a different location and therefore the image looks distorted.

Camera Non-Linearity and Resolution: We assume the camera is devoid of any non-linear distortion. However, even if this is not true, our method will not result in any pixel misregistration since the camera non-linearity will be accounted for by the rational Bezier patches. But, it will affect the accuracy of the screen reconstruction and hence, the final result may be slightly distorted. In case of severe camera non-linearities one can use standard camera calibration techniques to undistort the captured images [26]. Further, since our method can use multiple views from a camera, we do not need a very high-resolution camera.

8 Conclusion

In summary, we have presented the first work for markerless registration of tiled projection-based displays on swept surfaces, including 4 and 5-wall CAVEs, using an uncalibrated camera. Our method is automated, does not require large spaces due to multi-view calibration capability, and allows the use of compact short throw lenses on projectors. Our automated registration has the potential to increase the popularity of CAVEs and other swept surfaces for immersive VR displays. It also opens up the possibility of the use of smooth swept surfaces (as in Figure 13) for applications like digital signage and aesthetic projections in malls, airports and other public places.

In the future we would like to investigate the problem of color seamlessness in CAVEs. This is particularly challenging due to the presence of sharp corners in the CAVEs that lead to rapid changes in the reflectance properties of the surface. We believe a simple yet effective color registration method for multi-projector CAVEs can dramatically increase the sense of presence in immersive VR systems.

Further, we would like to analyze the robustness of our method to several practical issues including error in the curve segmentation, and use of non-nodal pan-tilt units. We achieved desirable results in face of all these. However, we believe a numerical analysis of the accuracy of our method versus the severity of the issue will be useful.

Acknowledgments

We would like to acknowledge our funding agencies NSF IIS-0846144. We would like to thank the members of the Creative Technologies Group at Walt Disney Imagineering for helping us to test our algorithm on their virtual reality system. We would like to thank Canon and Epson for donating projectors and cameras used for this research.

References

[1] R. Raskar, M. Brown, R. Yang, W. Chen, H. Towles, B. Seales, and H. Fuchs, "Multi projector displays using camera based registration," *Proc. of IEEE Vis*, 1999.

[2] D. Aliaga, "Digital inspection: An interactive stage for viewing surface details," *Proc. ACM Symp. on I3D*, 2008.

[3] D. Aliaga and Y. Xu, "Photogeometric structured light: A self-calibrating and multi-viewpoint framework for accurate 3d modeling," *Proc. of IEEE CVPR*, 2008.

[4] R. Raskar, J. V. Baar, T. Willwacher, and S. Rao, "Quadric transfer function for immersive curved screen displays," *Eurographics*, 2004.

[5] T. Johnson, F. Gyarfas, R. Skarbez, H. Towles, and H. Fuchs, "A personal surround environment: Projective display with correction for display surface geometry and extreme lens distortion," *IEEE Virtual Reality*, 2007.

[6] D. Cotting, M. Naes, M. Gross, and H. Fuchs, "Embedding imperceptible patterns into projected images for simultaneous acquisition and display," *ISMAR*, 2004.

[7] M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, and D. Tanguay, "Practical methods for geometric and photometric correction of tiled projector displays on curved surfaces," *IEEE PROCAMS*, 2006.

[8] W. Sun, I. Sobel, B. Culbertson, D. Gelb, and I. Robinson, "Calibrating multi-projector cylindrically curved displays for "wallpaper" projection," *IEEE/ACM PROCAMS*, 2008.

[9] B. Sajadi and A. Majumder, "Markerless view-independent registration of multiple distorted projectors on vertically extruded surface using a single uncalibrated camera," *IEEE Vis.*, 2009.

[10] B. Sajadi and A. Majumder, "Auto-calibration of cylindrical multi-projector systems," *IEEE Virtual Reality*, 2010.

[11] R. Raskar, "Immersive planar displays using roughly aligned projectors," in *Proc. of IEEE VR*, 2000.

[12] H. Chen, R. Sukthankar, G. Wallace, and K. Li, "Scalable alignment of large-format multi-projector displays using camera homography trees," *Proc. of IEEE Vis*, 2002.

[13] A. Raij and M. Polleyfeys, "Auto-calibration of multi-projector display walls," *Proc. of ICPR*, 2004.

[14] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. S. Brown, "Pixelflex: A reconfigurable multi-projector display system," *Proc. of IEEE Vis*, 2001.

[15] R. Yang, A. Majumder, and M. Brown, "Camera based calibration techniques for seamless multi-projector displays," *IEEE TVCG*, 2005.

[16] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs, "Pixelflex 2: A comprehensive automatic casually aligned multi-projector display," *IEEE PROCAMS*, 2003.

[17] M. Ashdown, M. Flagg, R. Sukthankar, and J. M. Rehg, "A flexible projector-camera system for multi-planar displays," *IEEE CVPR*, 2004.

[18] E. Bhasker, R. Juang, and A. Majumder, "Registration techniques for using imperfect and partially calibrated devices in planar multi-projector displays," *IEEE TVCG*, 2007.

[19] B. Sajadi and A. Majumder, "Auto-calibrating projectors for tiled displays on piecewise smooth vertically extruded surfaces," *IEEE TVCG*, 2011.

[20] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d," *ACM SIGGRAPH*, 2006.

[21] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: A unified approach to image based modeling and spatially immersive display," in *ACM Siggraph*, 1998.

[22] B. Springborn, P. Schröder, and U. Pinkall, "Conformal equivalence of triangle meshes," *ACM ToG*, vol. 27, no. 3, pp. 1–11, 2008.

[23] B. Sajadi and A. Majumder, "Scalable multi-view registration for multi-projector displays on vertically extruded surfaces," *Proceedings of EuroVis*, 2010.

[24] B. Sajadi, M. Lazarov, A. Majumder, and M. Gopi, "Color seamlessness in multi-projector displays using constrained gamut morphing," *IEEE Vis.*, 2009.

[25] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahem, P. Kirchner, and J. Klosowski, "Chromium : A stream processing framework for interactive rendering on clusters," *ACM SIGGRAPH*, 2002.

[26] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *ICCV*, 1999.

Behzad Sajadi is a PhD student at the Department of Computer Science in University of California, Irvine since 2007. He received his Bachelor's degree from Sharif University of Technology in 2006. His main research area is Computer Graphics and Visualization with particular interest in multi-projector displays and computational projectors and cameras. He has published several works on geometric and photometric registeration of multi-projector displays with the focus of practical designs for commodity large area displays. He has won the Best Paper award in Virtual Reality (VR) 2010 and the Second Best Paper award in Visualization 2009.

Aditi Majumder is an associate professor at the Department of Computer Science in University of California, Irvine. She received her PhD from Department of Computer Science, University of North Carolina at Chapel Hill in 2003. Her research area is computer graphics and vision, image processing with primary focus on multi-projector displays. Her research aims to make multi-projector displays truly commodity products and easily accessible to the common man. She has won three best paper awards in 2009-2010 in premier venues of IEEE Visualization, IEEE VR and IEEE PROCAMS. She is the co-author of the book "Practical Multi-Projector Display Design". She was the program and general co-chair of the Projector-Camera Workshop (PROCAMS) 2005 and the program chair of PROCAMS 2009. She was also the conference co-chair for ACM Virtual Reality Software and Technology 2007. She has played a key role in developing the first curved screen multi-projector display being marketed by NEC/Alienware currently and is an advisor at Disney Imagineering for advances in their projection based theme park rides. She is the recipient of the NSF CAREER award in 2009 for Ubiquitous Displays Via a Distributed Framework.