Auto-Calibration of Cylindrical Multi-Projector Systems

Behzad Sajadi*
Department of Computer Science
University of California, Irvine

Aditi Majumder†
Department of Computer Science
University of California, Irvine

ABSTRACT

In this paper we present a novel technique to calibrate multiple casually aligned projectors on a fiducial-free cylindrical curved surface using a single camera. We impose two priors to the cylindrical display: (a) cylinder is a vertically extruded surface; and (b) the aspect ratio of the rectangle formed by the four corners of the screen is known. Using these priors, we can estimate the display's 3D surface geometry and camera extrinsic parameters using a single image without any explicit display to camera correspondences. Using the estimated camera and display properties, we design a novel deterministic algorithm to recover the intrinsic and extrinsic parameters of each projector using a single projected pattern seen by the camera which is then used to register the images on the display from any arbitrary viewpoint making it appropriate for virtual reality systems. Finally, our method can be extended easily to handle sharp corners - making it suitable for the common CAVE like VR setup. To the best of our knowledge, this is the first method that can achieve accurate geometric auto-calibration of multiple projectors on a cylindrical display without performing an extensive stereo reconstruction.

Keywords: Multi-Projector Displays, Tiled Displays, Calibration, Registration

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality

1 INTRODUCTION

Cylindrical virtual reality systems are very common for large number of applications like gaming, entertainment, and 3D visualization. An inexpensive and popular way to increase the resolution of such displays is to tile multiple projectors on the cylindrical display surface. The challenge lies in automatic registration of the multiple projected imagery on the display surface to create one seamless image. The problem is further complicated when this needs to be achieved quickly without involving a time-consuming complete 3D reconstruction via structured light or attaching any special fiducials to the display surface.

Registering images from multiple projectors on non-planar displays requires 3D reconstruction of the display surface which in turn requires multiple cameras. Though there is a large body of literature that addresses such a reconstruction and registration [19, 8, 9, 12, 27, 13, 11, 22], these are complex procedures requiring camera calibration and multiple physical fiducials on the display surface. Hence, many methods try to avoid the complexity of using multiple cameras when using non-planar screens. Brown et al. [25] register multiple projectors with respect to the single point of view of a calibrating camera. This still achieves a seamless registration of the multiple projectors, but avoids the 3D reconstruction of the display surface entirely. Though widely adopted by commercial multi-projector display auto-calibration vendors, this registration is correct only from one view, the view of the calibrating camera. Hence, when viewed from a different point, distortions reflecting the camera perspective and the display surface geometry are visible. So, this registration is particularly unsuitable for VR applications where the user moves around deviating considerably from the position of the calibrating camera. [11, 22] try to avoid the view-dependency for cylindrical surfaces by relating the camera coordinates to the physical coordinates of the display by pasting special fiducials on the two rims of the display surface. Though this 'wall-papers' the imagery on the display, lack of a 3D display reconstruction does not allow registration from arbitrary viewpoints.

1.1 Main Contributions

In this paper, we present a new method that can register images from multiple projectors on a cylindrically non-planar display using a single uncalibrated camera and without using any fiducials in a *view-independent* manner – i.e. the registration does not depend on the view (pose and orientation) of the calibrating camera, hence, we can compute the correct image for any arbitrary view point. Contrary to multi-frame structured light patterns, we avoid using explicit correspondences between the display surface and the observing camera by imposing two priors: (a) a cylindrical display is a vertically extruded surface; and (b) we know the aspect ratio of the planar rectangle formed by the four corners of the display. With these priors, we can use a single image of the cylindrical surface from the camera to recover the camera pose and orientation and the 3D display geometry via a non-linear optimization. We design a deterministic geometric algorithm which uses these recovered properties to auto-calibrate (i.e. find both the intrinsic and extrinsic parameters) each projector from a single pattern captured by the camera. Once auto-calibrated, we achieve geometric registration on the display surface via a ray-casting method.

Unlike any existing method that use a single camera to register multiple projectors on a non-planar display [25, 11, 22], we can reconstruct the shape of the 3D display. This enables us to parametrize the display directly in a 3D coordinate system, rather than in the camera image space, to achieve a view-independent geometric registration. Finally, unlike the above methods which achieve registration without completely calibrating the projectors, we achieve a complete auto-calibration of the projectors. This results in a simple and accurate method to compute correspondences between the display and the projector. Hence, for static display surfaces, once the 3D display geometry is recovered, our auto-calibration method can be used for quickly changing the projector position and orientation to create displays of different scale, resolution and aspect ratio.

Our method can handle any smooth vertically extruded surface which can be interesting for entertainment purposes. More importantly, it can be easily extended to handle extruded surfaces with sharp edges. This opens up the possibility of using our algorithm in a CAVE like setup. Finally, the image correction required to register the images can be achieved in real-time on GPUs making our method especially suitable for real-time VR applications. To the best of our knowledge, this is the first method that can achieve complete auto-calibration and consequently a view-independent registration on specialized non-planar displays - vertically extruded surfaces like cylinders - without using any physical fiducials on the

*e-mail: bsajadi@ics.uci.edu
†e-mail:majumder@ics.uci.com

display surface.

Organization: We present a survey of related work in Section 2. We present our auto-calibration algorithm followed by two different view-independent registration techniques, suited for different applications, in Section 3. Next, we provide a proof of concept that our method can be extended to handle the common CAVE like VR systems in Section 4. We present our results in Section 5. Finally, we conclude the with future work in Section 6.

2 Related Work

Our work is related to a large body of literature that deals with various aspects of calibration in projection-based displays. Considering *planar surfaces and single cameras*, Raij and Pollefeys [17] and Raskar et al. [20] describe techniques to automatically calibrate multiple projectors on planar display surfaces. PixelFlex [16, 23] provided a multi-projector display on planar surfaces where each projector image can be easily and quickly repositioned to create new display configurations that can be calibrated within minutes. Bhasker et al. [4] achieve the same in presence of projector non-linearities (i.e. radial and tangential distortions) using rational Bezier patches.

Chen et al. [7] used *multiple cameras on planar displays* to achieve a homography-tree based registration across multiple projectors. Moving away from a centralized architecture where the multiple cameras and projectors are controlled from a central server, Bhasket et al. [5] present a distributed framework where each projector is augmented by a single camera and has the responsibility of registering itself with the rest of the display. An asynchronous distributed calibration algorithm runs on each augmented projector in a SIMD fashion to create a seamless display.

When considering *non-planar displays*, especially arbitrary ones, using *multiple cameras* becomes necessary for 3D reconstruction of the non-planar surface. Raskar et al. in [19] use special 3D fiducials to achieve a complete device (camera and projector) calibration and 3D reconstruction of the display surface using a large number of structured light patterns, which are then used to achieve the geometric registration. Aliaga et al. in [2, 1] also achieve a 3D reconstruction to register multiple images on complex 3D shapes, but without using any physical fiducials. To constrain the system sufficiently, this method uses completely superimposed projectors and cross-validates calibration parameters and display surface estimates using both photometric and geometric stereo, resulting in a self-calibrating system. Raskar et al. in [18] use a stereo camera pair to reconstruct special non-planar surfaces called quadric surfaces (spheres, cylinders, ellipsoids and paraboloids) and propose conformal mapping and quadric transfer to minimize pixel stretching of projected pixels after the geometric registration.

All of the above methods achieve a pre-calibration, sometimes in a few minutes. A complementary set of techniques exist that can focus on continuous image registration during the display time for change in the display shape and movement in the projectors. Yang and Welch [24] use the projected content (as opposed to special patterns) at the display time to automatically estimate the shape of the display surface and account for the changes in its shape over time. Using a projector augmented by two stereo cameras, Cotting et al. [8, 9, 12] estimate the shape of the display surface and the pose of a single projector continuously over time by embedding imperceptible calibration patterns into projected imagery. Zhou et al. [27] achieve the same by tracking displayed image features. Johnson et al. [13] show that multiple such units can be used in a distributed framework to achieve continuous geometric calibration in a multi-projector setup. Zollman et al. [28] present a hybrid technique that can compensate for small changes in display configuration using optical flow, and will resort to active structured light projection when the optical flow becomes unreliable.

Our work belongs to the body of literature that tries to avoid the complexity of using multiple cameras when using non-planar screens. Brown et al. [25, 6] avoid reconstructing the display geometry by registering multiple projectors with respect to the single point of view of a calibrating camera. More recently, [11, 22] tried to avoid this view-dependency in registration for the special case of cylindrical surfaces by finding a way to relate the 2D parametrization of the cylindrical display with that of the camera image space without reconstructing the 3D display surface. A precisely calibrated physical pattern is pasted along the top and bottom curves of the cylinder to provide a physical 2D display parametrization. By identifying the corresponding images of these fiducials in the observing camera, a piecewise planar representation of the display is achieved in the camera space. The projectors can then be registered directly in the display space rather than the camera space resulting in a 'wall-papered' registration. However, since it is not possible to have fiducials at a high spatial density on a display and the fiducials only samples the rims of the display, these methods result in distortions or stretching, especially towards the middle of the display surface. The more important point to note here is that in both these methods, since the display surface in not reconstructed, *registering images from an arbitrary viewpoint as is required in a virtual reality system, is not possible.* Our work uses a single uncalibrated camera, does not need to use physical fiducials, and can still achieve a calibration from any arbitrary viewpoint.

Technically, our work is close to [17] that achieves a similar goal of auto-calibration of projectors for planar screens, but our method is entirely different catered towards cylindrical screens. In particular, unlike [17] where the projector auto-calibration results from an involved optimization process, our projector auto-calibration is achieved by an efficient and fast deterministic algorithm allowing quick recalibration in the event of change in pose and orientation of the projectors. Further, we do not make restrictive assumptions like square projector pixels and identical vertical shift for all projectors.

3 Auto-Calibration Algorithm

Let the display surface, the image planes of the camera, and the projector be parametrized by (s,t), (u,v), and (x,y) respectively. We denote the 3D coordinates of the point at (s,t) on the display by $(X(s,t),Y(s,t),Z(s,t))$. Using the fact that a cylinder is a vertically extruded surface we impose the following constraints on the display surface. The four corners of the display lie on a planar rectangle, whose aspect ratio a is known. We define the world 3D coordinate system with Z axis perpendicular to this plane and X and Y defined as the two orthogonal basis of this planar rectangle. We also consider this planar rectangle to be at $Z = 0$ and the top and bottom curves of the cylinder to lie respectively on $Y = 1$ and $Y = 0$ planes in this coordinate system. Hence, $Y(s,0) = 0$ and $Y(s,1) = 1$. Further, these two curves are identical except for a translation in the Y direction. Consequently, $\forall s$, $(X(s,0),Z(s,0)) = (X(s,1),Z(s,1))$. These are illustrated in Figure 3.

We make the following practical assumptions to simplify the problem:

- Our camera and the projectors are linear devices with no radial distortion.
- Projectors are considered dual of a pin-hole camera.
- The camera intrinsic parameters are known, but not its pose and orientation.

For an n projector system, our auto-calibration takes $n+1$ images as input. The first image, I_0, is that of the display surface with no projectors turned on. Next, for each projector i, $1 \leq i \leq n$, we take a picture I_i of the same display surface with projector i projecting a special line pattern (Figure 1).

Our algorithm consists of three steps:

1. With I_0 as input we estimate the camera and display surface properties using a non-linear optimization (Section 3.1).

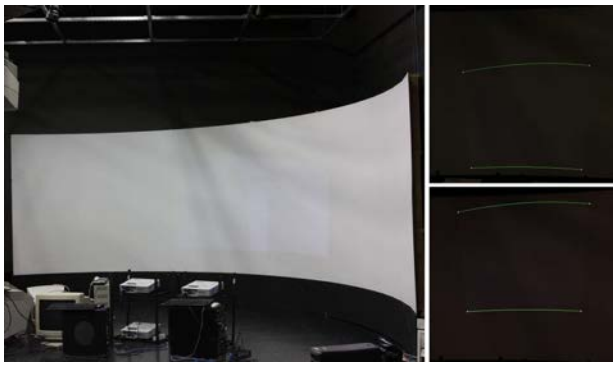

Figure 1: Left: The image I_0 used for estimating camera and display properties. Right: The zoomed in view of I_1 and I_2 for one of our setups, where the projector 1 and 2 are projecting the single pattern used for auto-calibration.

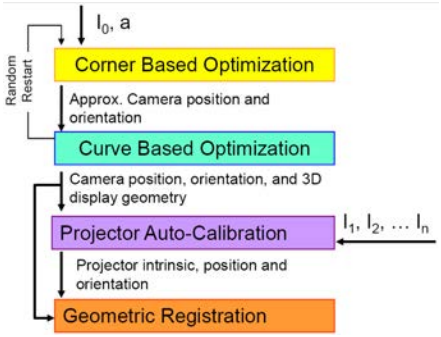

Figure 2: The pipeline of our algorithm.

2. Using the recovered camera and display properties and the image I_i, we find the intrinsic and extrinsic parameters of projector I_i, thus auto-calibrating the projectors. For this we use a deterministic algorithm which is fast and efficient enabling quick changes in projector properties (position, orientation, and zoom) (Section 3.2).
3. We use the recovered projector properties to register images seamlessly on the cylindrical display (Section 3.3). We present two ways to register the images. (a) The first kind of registration provides seamless imagery that looks correct from an arbitrary view point which can change without requiring a recalibration. This is suitable for a VR application with a head-tracked single user. (b) The second type of registration wall-papers the image seamlessly on the display. This is suitable for multi-user visualization applications.

The complete pipeline of our method is illustrated in Figure 2.

3.1 Camera and Display Property Estimation

The input to this step is the image I_0, the 3x3 intrinsic parameter matrix of the camera, and the aspect ratio a. The output is an estimation of the 3x4 extrinsic parameter matrix (defining position and orientation) of the camera and the 3D geometry of the display defined by the top and bottom 3D curves. The novelty of this step is to estimate the camera parameters and the 3D display geometry from a single image without using any correspondences. The correspondences are avoided by exploiting the fact that the top and the bottom curves of the display are identical except for being in two different XZ planes in 3D.

To estimate the camera parameters, we use a two phase nonlinear optimization method. In the first step we gather a *rough estimate* of the camera extrinsic parameters (pose and orientation) using the projection of just the corners of the display surface on the camera image. These rough estimates are then used to initialize the second optimization step with a more expensive error function that refines these camera extrinsic parameters to provide an *accurate estimate*. The recovered extrinsic camera parameters are then used to *estimate the 3D display geometry*.

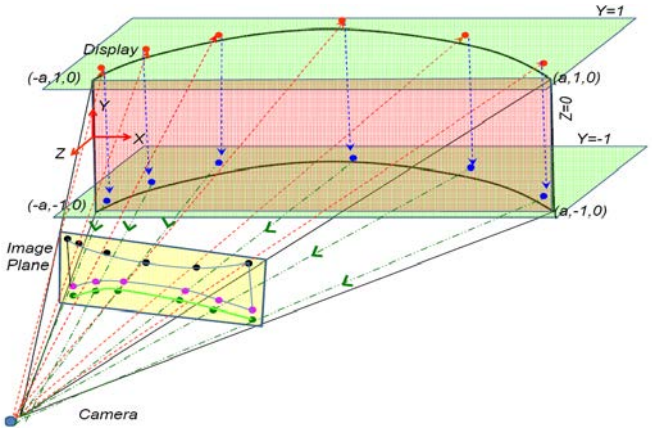

Figure 3: A curve is fitted through the sampled points on the 2D bottom curve (purple line). The sampled points on the 2D top curve in the camera (black) are reprojected in 3D to estimate the 3D top curve (red), and translated down to estimate of the 3D bottom curve (blue), and finally projected back on the camera (green). The distance between these points and the purple curve is minimized in the curve based optimization step.

Rough Estimation of Camera Parameters: The camera coordinates, (u,v), of any 3D point $(X(s,t),Y(s,t),Z(s,t))$ on the display are given by,

$$(uw,vw,w)^T = M(X(s,t),Y(s,t),Z(s,t),1)^T \quad (1)$$

where $(uw,vw,w)^T$ is the 2D homogeneous coordinate corresponding to the camera coordinate (u,v) and $M = K(R|RT)$ is the camera calibration matrix comprising of the 3×3 intrinsic parameter matrix K and the 3×4 extrinsic parameter matrix $(R|RT)$. We assume that K is known or estimated using standard camera calibration techiniques [26]. We estimate the $(R|RT)$ matrix that provides the pose and orientation of the camera. $(R|RT)$ comprises of six parameters including three rotations to define the orientation and a 3D center of projection (COP) of the camera to define the position. Given our 3D world coordinate system, the 3D locations of the four corners of the cylindrical display in a counter-clockwise manner starting from top left are given by: $(-\frac{a}{2},1,0)$, $(\frac{a}{2},1,0)$, $(\frac{a}{2},0,0)$, and $(-\frac{a}{2},0,0)$. In this step, we estimate the six camera extrinsic parameters by minimizing the reprojection error E_r, (i.e. sum of the distances of the projection of these corners on the image plane from the camera captured positions) of the 3D corners from the detected corners in the image.

To initialize this optimization, we use the following. The angles of rotations about the X, Y, and Z axes that comprise R are initialized to zero. T, the COP of the camera is initialized roughly at the center of the planar rectangle formed by the four corners of the display at a depth of a similar order of magnitude as the size of the display. This is achieved by initializing T to $(0,0,a)$.

Accurate Estimation of Camera Parameters: These rough estimates of the camera extrinsic parameters achieved in the previous step are used to initialize a second optimization that attempts to refine these parameters. Here, we augment the error function from the previous optimization step, E_r, with another error function, E_c, which is the reprojection error of the estimated 3D top and bottom curves of the cylindrical display (Figure 3). We seek to minimize the weighted combined error, $w_rE_r + w_cE_c$.

Let C_T and C_B be the 3D top and bottom curves of the display.

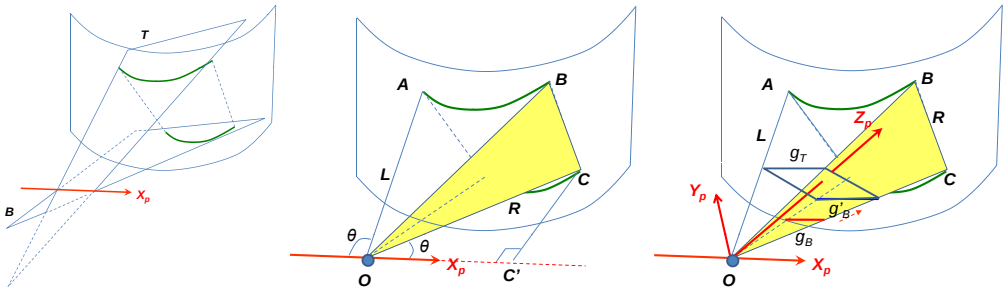

Figure 4: Illustrations demonstrating the methods to find X_p (left), enter of the camera O (middle), and Y_p and Z_p (right).

We first use segmentation and contour detection techniques to sample the 2D projected coordinates of C_T and C_B in the camera space, denoted by the set of samples $S_1^T, S_2^T, \ldots S_N^T$ and $S_1^B, S_2^B, \ldots S_N^B$ respectively. Prior to the optimization, we fit a curve C'_B through $S_1^B, S_2^B, \ldots S_N^B$. During the optimization, we reproject $S_1^T, S_2^T, \ldots S_N^T$ to the 3D coordinate system. To reproject the kth sample S_k^T, $1 \leq k \leq N$ in 3D, we cast a ray from the center of projection of the camera through $M^{-1}(S_k^T, 1)^T$. We intersect this ray with the $Y = 1$ plane to find the corresponding reprojected 3D coordinate, $r(S_k^T)$, where r denotes this reprojection function. We know that $r(S_k^T)$ when translated -1 unit in the y direction, i.e. $r(S_k^T) + h$, where $h = (0, -1, 0)$, should lie on C_B since the cylinder is a vertically extruded surface. We then project these translated 3D points, $r(S_k^T) + h$, back on to the camera image plane to generate a new set of points $Q_k^B = M(r(S_k^T) + h)$ where M is the 3D to 2D perspective projection of the camera. If the estimated camera calibration parameters are accurate, then all the samples $Q_1^B, Q_2^B, \ldots Q_N^B$ would lie exactly on C'_B. So, E_c is defined as the sum of the distances between $Q_1^B, Q_2^B, \ldots Q_N^B$ and C'_B in a least squares sense.

To solve both the above optimizations, we use standard gradient descent methods. To assure faster convergence we (a) apply a preconditioning to the variables so that the range of the values that can be assigned to them is normalized; and (b) use decaying step size.

Estimation of the Display Geometry: After convergence of the non-linear optimization process, we use the estimated camera calibration parameters to reproject $S_1^T, S_2^T, \ldots S_N^T$ and $S_1^B, S_2^B, \ldots S_N^B$ in 3D to find C_T and C_B respectively. Due to accumulated errors, C_T and C_B may not be identical. So, we project both the curves on $Y = 0$ plane and find their average to define C_B. This is then translated to $Y = 1$ to define C_T. We then use a polynomial curve fitting to find a parametric representation of C_T and C_B.

3.2 Projector Calibration

In this step, we project a pattern from each projector comprising of four corner blobs and a top and bottom line. An image I_i of this pattern is captured by the camera. Using I_i and the estimated camera calibration parameters and 3D display geometry, we estimate the intrinsic and extrinsic parameters of each projector.

Let the image of the top and bottom lines for the projector in I_i be denoted by l_T and l_B respectively. Let the blobs be denoted by b_A, b_B, b_C and b_D from the top left corner in a clockwise manner (Figure 1). Note that though these lines are straight in the projector space, they look curved in 3D due to projection on a curved surface. The auto-calibration of each projector consists of two steps. First, we find the view frustum of the projector defined by its center and five planes (top, bottom, left, right and the image plane) that define the extrinsic parameters of the projector. Next we use this view frustum and the known projector resolution ($W \times H$) to recover its intrinsic parameters. Most projectors have a vertical principle center offset to avoid occlusion with the table or the ceiling where the projector is mounted. This results in an offset in the y-direction for the principle center. We assume that the x-coordinate of the principle center coincides with the center of the x-direction. Additionally we do not consider any skew. This results in a simplified intrinsic parameter matrix K_p for the projectors given by

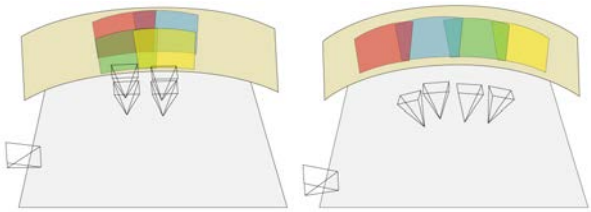

Figure 5: The estimated setup (camera, 3D display and projectors) using our algorithm for the 2×2 array (top) and 1×4 array of four projectors.

$$K_p = \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Hence, to recover the projector intrinsic parameters, we determine three parameters for each projector: the focal lengths in the two image coordinate directions (f_x and f_y) and the offset in the y direction (o_y). Our method is absolutely deterministic without using any optimizations and hence is accurate and efficient.

Estimation of the Extrinsic Parameters: Let us consider a 3D local coordinate frame for each projector defined by the COP, O, (position) and axes X_p, Y_p, and Z_p (orientation). We use a three step procedure to reconstruct the view-frustum of the projector. (a) First, we find a line that is parallel to X_p and passes through O. (b) Second, we find the position of O on X_p. (c) Finally, we recover the other two local coordinate axes Y_p and Z_p.

Finding X_p: We first sample l_T and l_B in the 2D camera space and reproject these samples in 3D using the estimated camera pose and orientation. Each sample defines a 3D ray. We find the intersection of these rays with the display via a line-curve intersection. This gives us the corresponding samples on the 3D curves l'_T and l'_B. Note that the samples of l'_T and l'_B lie respectively on the top and bottom planes of the projector view frustum, T and B. So, first we fit a plane to the samples of l'_T and l'_B in a linear least squares sense to estimate T and B. Then we find the intersection of T and B to find X_p.

Finding O: The center of projection, O, is on X_p. Since the projector view-frustum is symmetric in the horizontal direction, the center O is a point on the line X_p constrained by the fact that the two vertical planes formed by the view frustum, L and R, should make the same angle with the line X_p (Figure 4). We first reproject the blobs b_A, b_B, b_C, and b_D in 3D using the camera calibration matrix. This generates the four points A, B, C, and D where the four rays of the projector view frustum meet the display surface. For

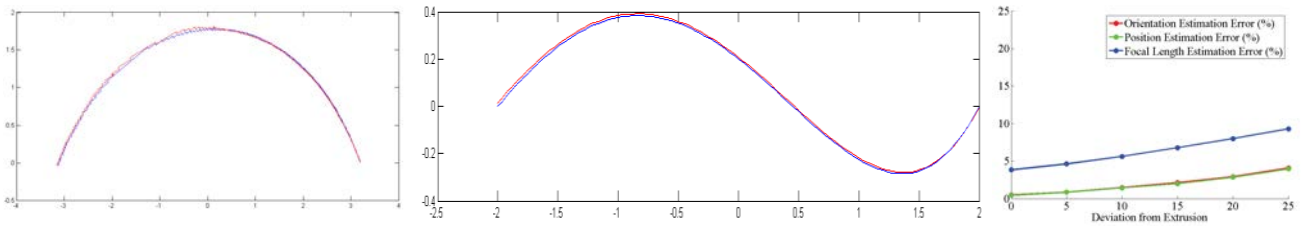

Figure 6: Left: In the real system, red and blue are the estimated 3D top and bottom curves of the display. Middle: In simulation, the blue curve is the original display surface curve and the red curve is the estimation provided by our method. Right: The plot shows the pixel misregistration as the surface deviates from being an extruded surface. This shows that if we assume a roughly tiled configuration and can tolerate 1 or 2 pixel misregistration, we can allow 4% and 6% deviation from an extruded surface respectively.

each of these, let A', B', C', and D' be the projection of A, B, C and D respectively on X_p. Due to equal angle constraint, the triangles OCC' and ODD' will be similar to each other and so will be OBB' and OAA'. Thus, the position of O on $D'C'$ will be given by the weighted average of D' and C' where the weights are inversely proportional to the lengths CC' and DD'. This yields to two estimates of O, one from the similar triangle constraint on OCC' and ODD' and another from the similar triangle constraint on OBB' and OAA'. A mean of these two estimates results in a robust estimate of O.

Finding Y_p and Z_p: We have thus found the four lateral planes of the projector view frustum. Now, we want to find the view direction Z_p. Note that for any plane P perpendicular to Z_p, the length of the intersections of P with OAB and ODC are equal (Figure 4). We use this constraint of equal length to find Z_p. We first consider two lines on OAB and ODC, g_T and g_B respectively. Both are parallel to X_p. g_T lies at unit distance from O on OAB. Similarly, g_B lies at unit distance from O on OCD. Note that g_T and g_B will not have equal length. Assuming $|g_T| > |g_B|$, if we move g_B along B keeping it parallel to X_p such that the distance from O becomes $\frac{g_T}{g_B}$, the new line g'_B thus formed is equal in length to g_T. Hence, the plane passing through g_T and g'_B satisfies the constraint of equal length (Figure 4). Z_p is the normal to this plane and $Y_p = Z_p \times X_p$.

Estimation of the Intrinsic Parameters: Let the resolution of the projector between the four blobs in the pattern be $P \times Q$. To find f_x and f_y, we first project the 3D points A, B, C, and D on a plane perpendicular to Z_p and at unit distance from O. Let these points be A'', B'', C'', and D'' respectively. Then, f_x is given by $\frac{P}{|A''B''|}$. Similarly, f_y is given by $\frac{P}{|A''C''|}$. To find o_y, we consider the center of the 3D line AB. Since we know the 3D coordinate of this point and f_x and f_y, we can find the projector y-coordinate for this point assuming $o_y = 0$ and subtract $\frac{Q}{2}$ from it to obtain o_y.

3.3 Geometric Registration

Geometric registration from an arbitrary viewpoint is achieved when an image rendered from the viewpoint is projectively mapped on the 3D display. This kind of registration is especially suitable for virtual reality applications like virtual walkthroughs. However, for other applications like visualization, collaboration, or teleconferencing that tend to have multiple users, correcting for a single viewpoint presents distortions for others. An image wallpapered on a surface has been time tested for multi-viewing purposes (e.g. in museums, airports, and other public places). For such scenarios, the images from the projectors are pasted or seamlessly 'wall-papered' on the display surface. In this section, we describe both these registrations.

Registration from Arbitrary Viewpoint: After extracting the geometry of the screen we can choose any arbitrary viewpoint and define an image plane for that viewpoint. Afterwards, we can find a mapping between the image plane coordinate system and the screen coordinate system by shooting rays from the viewpoint to the desired image plane coordinates and intersecting these rays with the screen. This mapping can be used then to correct any image for the defined coordinate system. The corrected image will show perspective distortion from other viewpoints and therefore it can be used only for a single user. This is well suited for single-user VR applications which use head tracking to find the proper viewpoint.

Wallpapered Registration: Following auto-calibration of the projectors, we use the projector and the display parameters to register the images from the multiple projectors on the display in a 'wallpaper' fashion. To wallpaper the image on the display, we seek a 2D length preserving parametrization of the 3D display surface with (s,t). As per our setup, t is automatically parametrized since $Y = t$. Also, $Z = f(X)$. Hence, we find a curve length based parametrization given by $s = \int_0^X \sqrt{1 + f'(x)}\, dx$.

The geometric registration involves finding the function that relates the projector coordinates (x,y) to the display parameters (s,t). Assuming the image to be wall-papered to have the same parametrization of the display, we first cast a ray through each pixel (x,y) using the auto-calibrated projector coordinates and find the 3D coordinates of its intersection with the cylindrical surface. Then we find the corresponding (s,t) values and bilinearly interpolate the color in the image to generate the color at the projector pixel (x,y).

4 Extension to Piecewise Planar CAVEs

Our algorithm assumes a vertically extruded surface. Since we assume that the top and bottom boundaries of the surface are smooth curves, the algorithm implicitly assumes a smooth vertically extruded surface. However, the basic algorithm remains unchanged even if we have a piecewise linear curve, instead of a smooth one. CAVE like VR setups are built on vertically extruded surfaces with piecewise linear boundaries, and hence our method can be easily extended to such situations. Currently, since it is difficult to calibrate multiple projectors on such surfaces, most CAVE setups use multiple projectors on each of the planar faces, but does not allow overlap of projectors across different faces. This does not allow blending regions for good color calibration [15, 14] and also makes it difficult to achieve automatic geometric calibration across the different planar faces. Our method removes this restriction by allowing the projectors to overlap even across the edges of the planar surfaces (Figure 10).

Our algorithm needs a few small changes to accommodate a CAVE kind of setup. When detecting the top and bottom curves in the camera image, we have to fit a piecewise linear function, instead of a smooth curve. Automatic piecewise linear regression (also referred to as segmented regression) pose ill-conditioned optimization problems for completely unconstrained data sets, but can be solved automatically if constrained sufficiently by providing the breakpoints between the number of segments [10]. Such inputs can easily be provided manually in the image I_0. For the particular case of displays, where the points on the detected display boundary are extremely structured (very little noise), it is very easy to manually

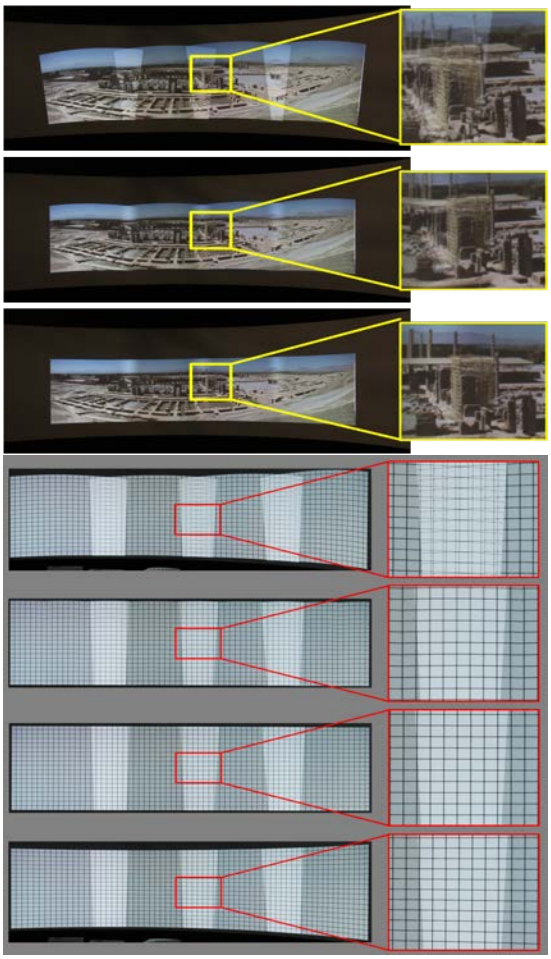

Figure 7: These two sets of images show our geometric registration for the panoramic configuration. The top three images for Persepolis from top to bottom show a naive method, homography, and our method. The bottom four images for a grid from top to bottom show homography, piece-wise linear method with a sparse set of correspondences, piece-wise linear method with a dense set of correspondences, and our method. Please note that the piecewise linear images are only correct from a single view-point and shod perspective distortion from any other viewpoint.

identify the piecewise linear line segments in the image I_0. Hence, we take this route of manual detection of the piecewise linear curve in the image I_0. Following this manual step, the rest of the process remains unchanged and still automatic.

5 Results

We have implemented our method on a cylindrical display using four projectors. We used Epson 1825p LCD projectors (about $600 each). Our display has a radius of about 14 feet and an angle of 90 degrees. We arranged the projectors in two different configurations: a panoramic configuration where projectors are arranged in a 1×4 array (projectors in a row) and a second one where they are arranged in a 2×2 array. Our unoptimized matlab implementation of the algoritm takes about 6 minutes. The non-linear optimization for estimating the camera parameters and display geometry takes about 5 minutes. Auto-calibration of the projectors takes about 10 seconds per projector. In the curve-based optimization step, we use $w_r = 3$ and $w_c = 1$.

Figure 5 provides a visualization of the estimated camera, display, and projector locations and orientaions in 3D using our algorithm for the two different setups. Figure 6 shows the error between the reconstructed top and bottom curves of the display. They coincide demonstrating the accuracy of our method.

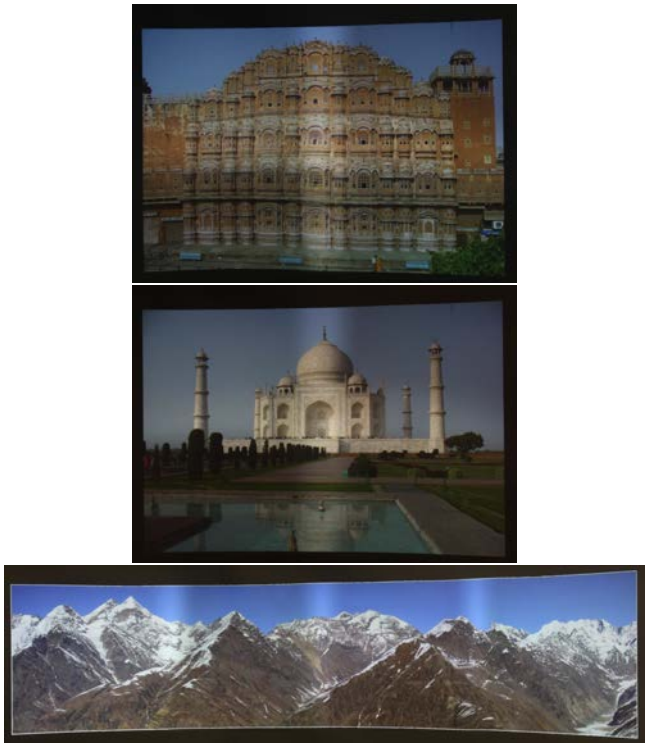

Figure 8: Top two: Geometric registration on our 2×2 projector display using our algorithm. Bottom: A registered and wall papered panorama of the himalayas in the panoramic setup.

The accuracy of our method is demonstrated by a robust geometric registration. Empirically, we have seen a maximum misregistration of less than a pixel. Figure 7, and 8 show the results on our displays. Since all prior methods can achieve geometric registration only with precise physical fiducials or complete 3D reconstruction, it is difficult to find a fair comparison to our method that does not use either of them. In the absence of fiducials, we compare our method with a naive homography-based registration [3, 7] and the registration with respect to a single view-point of the calibrating camera [25]. In addition to the obvious misregistrations, the shape of the display shows that wall-papering cannot be achieved on the curved surface without recovering the 3D geometry of the screen. Please zoom in to see the quality of registration. To reduce the higher brightness in the overlap region, we use a simple cosine blending function [19, 23]. Photometric seams can be further removed by using [15].

To demonstrate that our method is not limited to just cylinders, but can handle any smooth vertically extruded surface, we made an inexpensive flexible display using a rectangular sheet of flexible white styrene. This was supported by five poles to which the styrene sheet was attached (Figure 9). The shape of the profile curve of this extruded display can be changed by simply changing the position of the poles. Figure 9 illustrates the accuracy of our auto-calibration on such a display.

When a projector is moved after auto-calibration, we only need to find the change in the parameters of the moved projector. Since we use a deterministic method, we can achieve this in less than 10 seconds. Thus, we can achieve quick recalibration in the event of a projector movement.

We also demonstrate the extension of our algorithm to CAVEs.

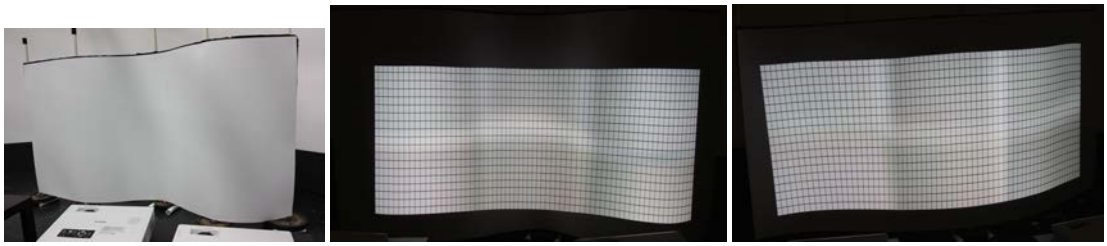

Figure 9: On the left: Our flexible screen. On the right: Geometric registration on our flexible screen with two different shapes.

Table 1: Percentage Errors of the estimated parameters over a large number of simulations with different configuration of the devices and the 3D display.

Parameter	Max	Mean	Std
Camera Orientation (deg)	0.494	0.192	0.167
Camera Position (%)	0.432	0.186	0.150
Top and bottom curves (%)	0.547	0.217	0.153
Projector Position (%)	0.313	0.115	0.972
Projector Orientation (deg)	0.131	0.052	0.050
Projector Focal Length (%)	0.295	0.105	0.895
Projector Offset (%)	1.251	0.486	0.452

Since we do not own a CAVE setup, we tried to create a proof-of-concept demo using our same flexible display setup. Instead of a smooth curve, we rearranged the poles to created a CAVE like setup. Figures 10 shows this setup and the results of our algorithm on it. Note that due to the very flexibility of this display, the curve deviates considerably from a piecewise linear curve. Hence, errors creep in and we cannot achieve a sub-pixel accuracy – but see 2-3 pixels misregistration, as shown in the video. However, in simulation, we can achieve the same sub-pixel accuracy as for our smooth extruded surface.

Figure 10: Top: A picture of our CAVE setup with the projectors overlapping on the edges. Bottom: Geometric registration on our CAVE setup. The calibrating camera is visible in the top-left corner of the picture.

5.1 Evaluation

We have conducted an extensive analysis on the accuracy of the estimated parameters using a simulator. The maximum, mean and standard deviation errors are presented in Table 1. For the orientation of devices (projectors and cameras), we provide deviation in degrees from the actual orientation. For translation, we provide the ratio of the error in estimation with the distance from the screen. For all other parameters, we measure the deviation from the original value of the parameter devided by the original value.

To compare the accuracy of the estimation of the display curves, we sample the estimated curves densely. Then, for each sample, we find the minimum distance to the original curve. The ratio of the maximum of these distances to the length of the original curve is considered to be the accuracy of the display geometry reconstruction and is reported in Table 1. For this, we did not limit the error analysis to cylindrical displays only, but ran experiments with any vertically extruded surface including ones with piecewise linear boundaries like CAVE. To evaluate the accuracy of the geometric registration, we find the deviation of (s,t) parameter to which a projector pixel will be mapped to in the original setup and compare it with the same from the estimated setup. We find a maximum of 0.3 pixel misregistration from a single projector. Hence, assuming a roughly tiled configuration, the worst case misregistration in any direction will be $0.3 \times 2 = 0.6$ pixels. This is consistent with our empirical observation of geometric misregistration of less than a pixel. Such accurate registration and calibration for cylindrical tiled displays has never been reported in the literature prior to our work. Finally, we also show the generality of our method for handling any vertically extruded surface (not necessarily a cylinder). The accuracy of reconstructing the display curves are evaluated over extrusions of different shapes. One example is shown in Figure 6.

5.2 Discussion

Most screens designed for commercial purposes are quite rigid infra-structure as is the screen we used for this work. However, we studied the effect of small deviation from extruded surface on the geometric registration in simulation. For this, the deviation is simulated using the same metric as is used to measure the accuracy of estimating the curves. The results in Figure 6 shows that the surface need not be perfectly extruded. 4 to 6% deviation from thereof results in less than 1 to 2 pixel misregistration.

The projectors we used, even if inexpensive, were close to perfect linear devices. However, sometimes they may have small radial distortions. In such a case, a pattern can be used that has more then just two lines. If m equally placed lines are used, X_p will be provided by the intersection of the m planes each containing a planar curve in 3D corresponding to the lines on the projector image plane. When fitting each plane, the eigenvalues of the matrix used for the linear least square fit provides some insights on the 3D curve shape. A small third eigenvalue indicates a degenerate case where the curve is close to a line and one cannot robustly fit a plane. A high forth eigenvalue indicates a large fitting error, i.e. the curve does not lie on a plane due to the presence of radial distortion in the projectors. Hence, when finding X_p using linear least squares intersection of the planes, the equations due to each plane can be weighted by a ratio of its third and forth eigenvalues found during the prior plane fitting step. This assures that curves which indeed lie on a plane are given larger weight than either the degenerate case or when severe radial distortion is present. To avoid an infinite weight resulting from a forth eigenvalue which is close to 0 (the best case

of a curve robustly lying on a plane), we provide a threshold to the maximum weight. Our simulation shows acceptable registration when using this method in the presence of small radial distortions.

Finally, we found that knowing the intrinsic parameters of the camera is not critical for our method. A large number of image formats like jpg or tiff store EXIF tags for images provide some of the camera parameters. One of these is the focal length, the most important parameter of the intrinsic parameter matrix K of the camera. To convert the focal length to the unit of pixels, we divide resolution of the camera by the CCD sensor size and multiply it with the focal length specified in the EXIF tags. The sensor size of the camera is available in its specifications. Also, in most cameras today, it is common to have the principal center at the center of the image, no skew between the image axes, and square pixels. Hence, similar to [21], we use these assumptions to initialize the intrinsic parameter matrix of a camera, K, as

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

Our non-linear optimization can accurately refine the single parameter in the intrinsic matrix and we do not see any degradation in the quality of the registration.

6 Conclusion

In summary, we have presented the first work to auto-calibrate projectors on vertically extruded surfaces without using display to camera correspondences. Our projector auto-calibration is achieved via a deterministic efficient algorithm that allows interactive changes in the projector position, orientation and zoom factor. Our method can have tremendous applications in auto-calibration of large cylindrical displays commonly used for edutainment purposes. It can also be extended to CAVEs to allow projector overlap across the planar screens of the CAVE.

However, our method is limited to extruded surfaces and cannot handle another kind of commonly used screens, the domes. In future, we would like to extend similar fundamentals of using prior knowledge of the screen to design methods to achieve geometric registration without the use of fiducials on other types of screen. Further, we would like to design a variation of our method that can tolerate greater deviation from extruded surfaces. Reasonable deviation from perfectly extruded surfaces will allow lower precision in the screen manufacturing, making these displays more affordable.

Acknowledgements

We would like to acknowledge our funding agencies NSF IIS-0846144. We would like to acknowledge Maxim Lazarov for helping us creating the video and building the flexible vertically extruded surface. We would like to thank the members of the Creative Technologies Group at Walt Disney Imagineering for helping us to test our algorithm successfully on their virtual reality system.

References

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

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

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

[4] E. Bhasker, R. Juang, and A. Majumder. Registration techniques for using imperfect and partially calibrated devices in planar multi-projector displays. *IEEE TVCG*, 13(6):1368–1375, 2007.

[5] E. Bhasker, P. Sinha, and A. Majumder. Asynchronous distributed calibration for scalable reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics (Visualization) - To Appear*, 2006.

[6] M. Brown, A. Majumder, and R. Yang. Camera based calibration techniques for seamless multi-projector displays. *IEEE TVCG*, 11(2), March-April 2005.

[7] 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.

[8] D. Cotting, M. Naef, M. Gross, , and H. Fuchs. Embedding imperceptible patterns into projected images for simultaneous acquisition and display. *International Symposium on Mixed and Augmented Reality*, pages 100–109, 2004.

[9] D. Cotting, R. Ziegler, M. Gross, and H. Fuchs. Adaptive instant displays: Continuously calibrated projections using per-pixel light control. *Proc. of Eurographics*, pages 705–714, 2005.

[10] G. Ferrari-Trecate and M. Muselli. A new learning method for piecewise linear regression. *ICANN Lecture Notes in Computer Science, Eds. J. D. Dorronsoro, Berlin: Springer*, 2415:444–449, 2002.

[11] 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.

[12] T. Johnson and H. Fuchs. Real-time projector tracking on complex geometry using ordinary imagery. *IEEE CVPR Workshop on Projector Camera Systems (PROCAMS)*, 2007.

[13] T. Johnson, G. Welch, H. Fuchs, E. L. Force, and H. Towles. A distributed cooperative framework for continuous multi-projector pose estimation. *IEEE Virtual Reality Conference*, pages 35–42, 2009.

[14] A. Majumder and R. Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. *IEEE Transactions on Vis and Computer Graphics*, 10(2), March–April 2003.

[15] A. Majumder and R. Stevens. Perceptual photometric seamlessness in tiled projection-based displays. *ACM TOG*, 24(1), January 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] A. Raij and M. Polleyfeys. Auto-calibration of multi-projector display walls. *Proc. of ICPR*, 2004.

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

[19] 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.

[20] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: Geometrically aware and self-configuring projectors. *ACM TOG*, 22(3), 2003.

[21] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press.

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

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

[24] R. Yang and G. Welch. Automatic projector display surface estimation using every-day imagery. *9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2001.

[25] R. Yang, G. Welch, and G. Bishop. Real-time consensus-based scene reconstruction using commodity graphics hardware. *Proceedings of Pacific Graphics*, 2002.

[26] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision*, 1999.

[27] J. Zhou, L. Wang, A. Akbarzadeh, , and R. Yang. Multi-projector display with continuous self-calibration. *IEEE/ACM Workshop on Projector-Camera Systems (PROCAMS)*, 2008.

[28] S. Zollmann, T. Langlotz, and O. Bimber. Passive-active geometric calibration for view-dependent projections onto arbitrary surfaces. *In Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR*, 2006.