

Autocalibrating Tiled Projectors on Piecewise Smooth Vertically Extruded Surfaces

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

Abstract—In this paper, we present a novel technique to calibrate multiple casually aligned projectors on fiducial-free piecewise smooth vertically extruded surfaces using a single camera. Such surfaces include cylindrical displays and CAVEs, common in immersive virtual reality systems. We impose two priors to the display surface. We assume the surface is a piecewise smooth vertically extruded surface for which the aspect ratio of the rectangle formed by the four corners of the surface is known and the boundary is visible and segmentable. Using these priors, we can estimate the display's 3D geometry and camera extrinsic parameters using a nonlinear optimization technique from a single image without any explicit display to camera correspondences. Using the estimated camera and display properties, the intrinsic and extrinsic parameters of each projector are recovered using a single projected pattern seen by the camera. This in turn is used to register the images on the display from any arbitrary viewpoint making it appropriate for virtual reality systems. The fast convergence and robustness of this method is achieved via a novel dimension reduction technique for camera parameter estimation and a novel deterministic technique for projector property estimation. This simplicity, efficiency, and robustness of our method enable several coveted features for nonplanar projection-based displays. First, it allows fast recalibration in the face of projector, display or camera movements and even change in display shape. Second, this opens up, for the first time, the possibility of allowing multiple projectors to overlap on the corners of the CAVE—a popular immersive VR display system. Finally, this opens up the possibility of easily deploying multiprojector displays on aesthetic novel shapes for edutainment and digital signage applications.

Index Terms—Tiled displays, autocalibration, geometric registration, cylindrical displays, CAVES.

1 INTRODUCTION

VERTICALLY extruded surfaces (e.g., cylinders and CAVEs) are commonly used in virtual reality (VR) systems. VR systems are common for a 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 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 of the display surface via structured light or special fiducials on the display surface. Finally, a fast and robust automatic registration is essential to allow recalibration in the face of change in display, projector and camera parameters (e.g., shape, pose, and orientation).

Registering images from multiple projectors on nonplanar 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 [1], [2], [3], [4], [5], [6], [7], [8], these are complex and time-consuming 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 nonplanar screens. Brown and Seales [9] 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 multiprojector display autocalibration 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. Harville et al. [7] and Sun et al. [8] 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 “wallpapers” the imagery on the display, lack of a 3D display reconstruction does not allow registration from arbitrary viewpoints. Sajadi and Majumder [10] address vertically extruded surface like ours, but can only handle smooth surfaces and not sharp edges and corners, as is common in CAVES.

1.1 Main Contributions

In this paper, we present a new method to register images from multiple projectors on a piecewise smooth vertically extruded display using a single uncalibrated camera and without using any fiducials. Contrary to multiframe structured light patterns, we avoid using direct point correspondences between the display surface and the observing camera. This is achieved by imposing three priors: 1) the

• The authors are with the Department of Computer Science, University of California, Irvine, Bren Hall, Irvine, CA 92697-3435.
E-mail: bsajadi@uci.edu, majumder@ics.uci.edu.

Manuscript received 15 Aug. 2010; revised 22 Nov. 2010; accepted 13 Jan. 2011; published online 3 Feb. 2011.

Recommended for acceptance by K. Kiyokawa, G. Klinker, and B. Lok.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCGSI-2010-08-0182.

Digital Object Identifier no. 10.1109/TVCG.2011.33.

display is a vertically extruded surface; 2) the aspect ratio of the planar rectangle formed by the four corners of the display is known; and 3) the boundary of the display is visible and segmentable. Our method has three steps:

- We recover the camera pose and orientation and the 3D display geometry using only a single image of the display surface (Section 3.1).
- Next, we autocalibrate (i.e., find the intrinsic and extrinsic parameters) each projector from a single pattern captured by the camera. This is done using a deterministic, fast, and robust algorithm (Section 3.2).
- Once autocalibrated, we achieve geometric registration via a ray-casting method (Section 3.3).

Unlike many existing methods that use a single camera to register multiple projectors on a nonplanar display [9], [7], [8], 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 previous methods which achieve registration without calibrating the projectors, we achieve a complete autocalibration of the projectors. This results in a simple and accurate method to compute direct point correspondences between the display and the projectors. Hence, for static display surfaces, once the 3D display geometry is recovered, our autocalibration method can be used to quickly change the projector pose and orientation to create displays of different scale, resolution, and aspect ratio. Fast convergence of the method also allows for changing the display shape and position and the camera position.

Our method handles piecewise smooth surfaces with sharp edges. This opens up the possibility of using our algorithm in a CAVE like setup. Since smooth surfaces are a subset of piecewise smooth surfaces, we can handle cylindrical VR systems and even novel smooth shapes integrated with the physical world for entertainment, and digital sign applications (Fig. 8). Finally, the image correction required to register the images can be achieved in real time on GPUs making our method suitable for real-time VR applications.

Organization. We present a survey of related work in Section 2. We present our autocalibration 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 6. Finally, we conclude with the future work in Section 7.

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*, Raji and Pollefeys [11] and Raskar et al. [12] describe techniques to automatically calibrate multiple projectors on planar display surfaces. PixelFlex [13], [14] provides multiprojector 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. [15] achieve the same in presence of projector nonlinearities (i.e., radial and tangential distortions) using rational Bezier patches.

Chen et al. [16] 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, Bhasker et al. [17] 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 nonplanar surface. Raskar et al. in [1] 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 and Xu in [18] and [19] 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 [20] use a stereo camera pair to reconstruct special nonplanar 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 precalibration, sometimes in a few minutes. A complementary set of techniques focus on continuous image registration during the display time for change in the display shape and movement of the projectors. Yang and Welch [21] use the projected content (as opposed to special patterns) at the display time to automatically estimate the shape of the display and account for the changes in its shape over time. Using a projector augmented by two stereo cameras, Cotting et al. [2], [3], [4] estimate the shape of the display and the pose of a single projector continuously over time by embedding imperceptible calibration patterns into the projected imagery. Zhou et al. [5] achieve the same by tracking the displayed image features. Johnson et al. [6] show that multiple such units can be used in a distributed framework to achieve continuous geometric calibration in a multiprojector setup. Zollman et al. [22] present a hybrid technique that compensates for small changes in the display configuration using optical flow, and resorts to active structured light projection when the optical flow becomes unreliable.

To avoid the complexity of using multiple cameras when using nonplanar screens, Brown et al. [23], [9] register multiple projectors with respect to the single point of view of a calibrating camera. More recently, [7], [8] 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 display 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

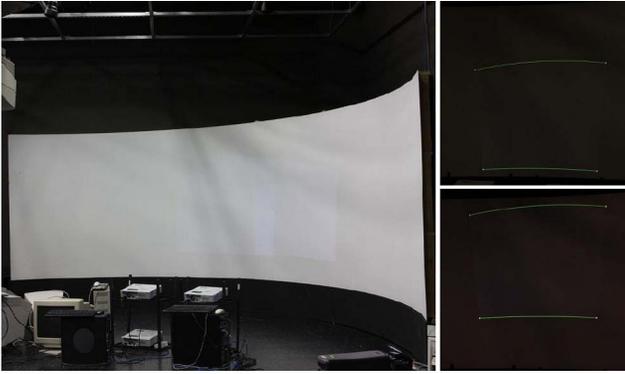


Fig. 1. Left: The image I_0 used for estimating camera and display properties. Right: The zoomed in view of I_1 and I_2 for one of our setups, where the projector 1 and 2 are projecting the single pattern used for autocalibration.

camera space resulting in a “wallpapered” registration. However, since it is not possible to have fiducials at a high spatial density on a display and the fiducials only sample the rims of the display, these methods result in distortions or stretching, especially toward the middle of the display surface. More importantly, since the display surface is not reconstructed in both these methods, registering images from an arbitrary viewpoint, as is required in a virtual reality system, is not possible.

In terms of goals achieved, our work is close to [11] that achieves a similar goal of autocalibration of projectors for planar screens, but our method is entirely different, handling a particular class of nonplanar surfaces—piecewise smooth vertically extruded surfaces. In particular, unlike [11] where the projector autocalibration results from an involved optimization process, our projector autocalibration 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, unlike [11], we do not make restrictive assumptions like square projector pixels and identical vertical shift for all projectors.

Our work belongs to a body of literature that assumes some priors on the display shape to constraint the problem of estimating the display shape from a single uncalibrated camera. Sajadi and Majumder [10] assume the knowledge of a smooth vertically extruded surface and the aspect ratio of the rectangle formed by its corner and can estimate the properties of the calibrating camera (focal length, pose, and orientation) and 3D shape of the display surface. This is then used to find the mapping between projector coordinates and the 3D display coordinates to achieve a geometric registration. The optimization method used in this method to recover the camera parameters is similar to our method. However, we improve on their optimization method to achieve higher speed and convergence rate. Further, this method cannot handle surfaces with sharp corners. Finally, it does not achieve a complete autocalibration of the projectors. Hence, it cannot achieve a fast recalibration when projectors are moved. Sajadi and Majumder [24] use a single uncalibrated camera and achieve a complete autocalibration of the projectors. However, this method is slow to converge and hence cannot allow changes in the display shape and position and in the camera position. In this work, we introduce a novel dimension reduction technique to achieve fast convergence and robustness. This enables fast recalibration in the face of changes in the parameters of all

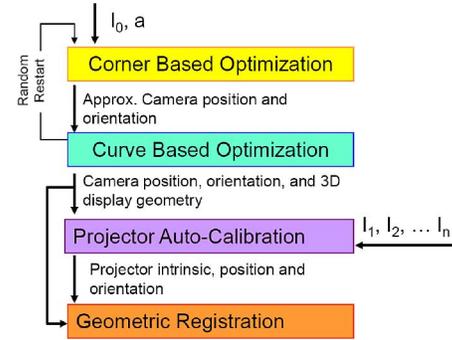


Fig. 2. The pipeline of our algorithm.

of display, camera, and projectors. However, unlike [10], we cannot handle nonlinear projector distortions.

3 AUTOCALIBRATION 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 a point at (s, t) on the display by $(X(s, t), Y(s, t), Z(s, t))$. Using the fact that the display is vertically extruded, we impose the following constraints on it. The four corners of the display lie on a planar rectangle with known aspect ratio, a . We define the 3D world 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 surface 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 shown in Fig. 3.

We make the following practical assumptions to simplify the problem: 1) Projectors are considered dual of a pin-hole camera; 2) Our camera and the projectors are linear devices with no radial distortion; 3) The camera intrinsic parameters are known, but not its pose and orientation.

For an n projector system, our autocalibration 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 (Fig. 1).

Our algorithm consists of three steps shown in Fig. 2:

1. *Camera and Display Property Estimation.* With I_0 as input we estimate the camera and display properties using a nonlinear optimization (Section 3.1).
2. *Projector Calibration.* Using the recovered camera and display properties and the image I_i , we find the intrinsic and extrinsic parameters of projector I_i , thus autocalibrating 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. *Geometric Registration.* We use the recovered projector properties to register the imagery seamlessly on the display (Section 3.3). We present two ways to register the images. 1) The first kind of registration provides seamless images that look correct for an

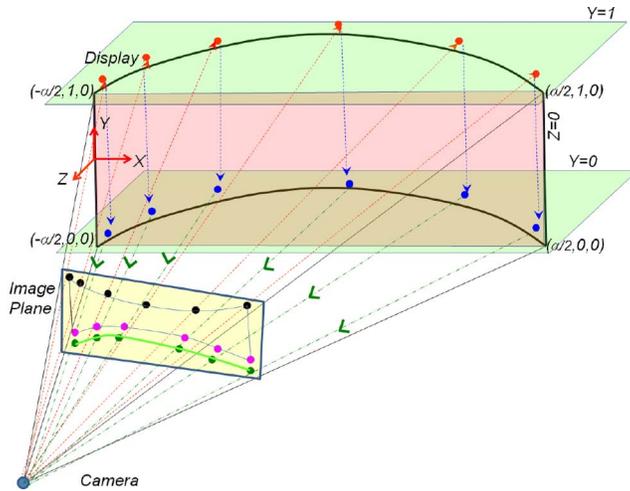


Fig. 3. A curve is fitted through the sampled points on the 2D bottom curve (purple line). The sampled points on the 2D top curve in the camera (black) are reprojected in 3D to estimate the 3D top curve (red), and translated down to estimate of the 3D bottom curve (blue), and finally projected on the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

arbitrary viewpoint, suitable for VR applications with head-tracked single user. 2) The second type of registration wallpapers the image seamlessly on the display. Though not correct from any viewpoint, wallpapered display is suitable for multiuser applications like large scale visualization.

3.1 Camera and Display Property Estimation

The input to this step is the image I_0 , the 3×3 intrinsic parameter matrix of the camera, and the aspect ratio a . The output is an estimation of the 3×4 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 direct point 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 adapt the method introduced by Sajadi and Majumder [10]. This method uses 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*. We improve the method introduced by Sajadi and Majumder [10] in Section 3.1.3 to achieve higher speed and convergence rate. The recovered extrinsic camera parameters are then used to *estimate the 3D display geometry*.

3.1.1 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 $(R|RT)$.

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 [25], we use these assumptions to initialize the intrinsic parameter matrix of a camera, K , as

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

We also 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. Therefore the total number of unknowns is seven which includes the focal length, three rotation parameters, and three translation parameters.

Given the 3D world coordinate system, the 3D locations of the four corners of the 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 seven unknowns by minimizing the reprojection error, E_r , of the 3D corners from the detected corners in the image (i.e., sum of the distances of the projection of these corners on the image plane from the camera captured positions).

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 $(0, 0, a)$.

3.1.2 Accurate Estimation of Camera Parameters

The 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 piecewise smooth vertically extruded surface (Fig. 3). We seek to minimize the weighted combined error, $w_r E_r + w_c E_c$ where w_r and w_c comprise the weights.

Let C_T and C_B be the 3D top and bottom curves of the display. 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 $S_1^T, S_2^T, \dots, S_N^T$ and $S_1^B, S_2^B, \dots, S_N^B$, respectively. First, we fit a curve C_B^B through $S_1^B, S_2^B, \dots, S_N^B$. Next, we reproject

$S_1^T, S_2^T, \dots, S_N^T$ to the 3D coordinate system. To reproject the k th 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 display surface 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 $Q_1^B, Q_2^B, \dots, Q_N^B$ would lie exactly on C'_B . So, E_c , the reprojection error of the 3D top and bottom curves of the vertically extruded surface, is defined as the sum of the distances between $Q_1^B, Q_2^B, \dots, Q_N^B$ and C'_B in a least squares sense.

This optimization can be solved by standard gradient descent methods. To assure convergence, one will need to 1) apply a preconditioning to the variables so that the range of the values that can be assigned to them is normalized; and 2) use decaying step size. However, since the optimization conducts a search in a 7-dimensional space, this leads to the following problems:

- Converges for some of the cases only after multiple random restarts.
- Requires a relatively accurate estimate of the focal length of the camera.
- Converges to a local minima when the initial position is far from the answer or when the surface deviates considerably from being an extruded surface.

Hence, in the next section we propose a novel dimension reduction technique that allows the optimization technique to operate on a lower dimensional space. This strengthens our method by making it faster and more robust. Further, it removes the requirement of providing the camera focal length estimate.

3.1.3 Dimension Reduction of the Optimization Space

If the unknowns are assumed to be the rotation angle, COP, and focal length, the reprojection error is a nonlinear equation involving seven unknowns demanding a nonlinear optimization that should operate on a 7-dimensional search space. Instead, we consider the camera calibration matrix M and its 12 elements. Hence,

$$M = \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \end{pmatrix}. \quad (3)$$

If we consider any 3D point $V = (X, Y, Z, 1)^T$ whose 2D position in the camera image plane is known to be (u, v) , then we can write the following:

$$(uw, vw, w)^T = MV. \quad (4)$$

This can be written as

$$xw = M_1V, \quad yw = M_2V, \quad w = M_3V. \quad (5)$$

Consequently,

$$xM_3V - M_1V = 0, \quad yM_3V - M_2V = 0. \quad (6)$$

TABLE 1

Mean, Max, and Standard Deviation of the Time Required for Convergence before and after Dimension Reduction

Method	Max	Mean	Std
Before Dimension Reduction	337	172	103
After Dimension Reduction	20.3	6.48	2.85

All of the values are in seconds. The experiments are performed on several display shapes in simulation and the time is reported only for display reconstruction and does not include the projector calibration time.

Therefore, each point correspondence provides two equations. Since we know the corresponding 2D projections for the four corners of the display, we get eight equations. However, this results in an under constrained linear system since we have eight equations and twelve unknown values.

The dimension reduction is achieved by first fixing the last row of M , M_3 . Also since we know that the first three elements of M_3 form a unit vector, we only need to fix M_{31} , M_{32} , and M_{34} and choose M_{33} such that $|(M_{31}; M_{32}; M_{33})| = 1$. Eight unknowns remain after fixing the four elements of M_3 and the system can be solved using linear least squares optimization.

We know that

$$M = \begin{pmatrix} fR_1 & fR_1T \\ fR_2 & fR_2T \\ R_3 & R_3T \end{pmatrix}. \quad (7)$$

Since R is an orthonormal rotation matrix the elements of M_1 and M_2 should conform to the following constraints:

$$E_{size} = |(M_{11}, M_{12}, M_{13})| - |(M_{21}, M_{22}, M_{23})| = 0, \quad (8)$$

$$E_{ang1} = (M_{11}, M_{12}, M_{13}) \cdot (M_{21}, M_{22}, M_{23}) = 0, \quad (9)$$

$$E_{ang2} = (M_{11}, M_{12}, M_{13}) \cdot (M_{31}, M_{32}, M_{33}) = 0. \quad (10)$$

In the new nonlinear optimization system, we define the error function as deviation of matrix M from the above constraints. We define the new error function $E_{lin} = E_{size}^2 + E_{ang1}^2 + E_{ang2}^2$ to search for the correct M_3 which conforms with the above equations. We seek to minimize this error by choosing M_3 and use linear least squares optimization to find M_1 and M_2 . Since we use the nonlinear optimization only for finding M_3 , the search takes place in a 3-dimensional space instead of a 7-dimensional space. We initialize the process with $M_3 = (0, 0, 1, a)$. This is similar to our initialization in Section 3.1.1.

The next issue is to include the constraints due to the similarity of the top and bottom curves in this optimization. For this we add E_c , the curve reprojection error, to E_{lin} . Therefore, the final error function will be $w_{lin}E_{lin} + w_cE_c$.

This new approach has several advantages compared to the optimization described in Sections 3.1.1 and 3.1.2.

- *Guaranteed convergence.* Unlike the nonlinear optimization in 7-dimensional space, this new method guarantees convergence even if the initialization is far from the real solution. More importantly, it does not converge to a local minima.
- *Efficiency.* Since the number of variables is reduced to three the optimization tends to converge much faster, removing the need for a two phase optimization anymore. Table 1 shows the convergence time before and after dimension reduction. Please note that

TABLE 2

Error Values in the Estimated Parameters over a Large Number of Simulations with Different Configurations of the Devices and the 3D Display before and after Dimension Reduction

Parameter	Before Dim. Reduction			After Dim. Reduction		
	Max	Mean	Std	Max	Mean	Std
Cam Orientation	0.494	0.192	0.167	0.322	0.149	0.098
Cam Position	0.432	0.186	0.150	0.327	0.124	0.103
Cam Focal Len.	3.82	2.26	0.981	2.23	1.89	0.827
Extrusion Curve	0.547	0.217	0.153	0.390	0.168	0.174
Proj. Position	0.313	0.115	0.972	0.298	0.104	0.881
Proj. Orientation	0.131	0.052	0.050	0.126	0.043	0.048
Proj. Focal Len.	0.295	0.105	0.895	0.278	0.093	0.814
Proj. Offset	1.251	0.486	0.452	1.052	0.428	0.395

The values are in degrees for the camera and projector orientation and percentage for the rest of quantities.

without applying the dimension reduction, the optimization sometimes converges to a local minima.

- *No Camera Calibration.* Since we do not need an estimate of the focal length anymore, we can use entirely uncalibrated cameras. For some of the cameras especially webcams an estimate of the focal length might not be available.
- *Higher accuracy.* Finally, this results in greater accuracy, even when the surface deviates from a perfect vertically extruded surface (Table 2).

3.1.4 Estimation of the Display Geometry

After convergence of the nonlinear optimization, we use the estimated camera calibration parameters to reproject $S_1^T, S_2^T, \dots, S_N^T$ and $S_1^B, S_2^B, \dots, S_N^B$ 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 images 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 (Fig. 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 autocalibration 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

$$K_p = \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (11)$$

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.

3.2.1 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. 1) First, we find a line that is parallel to X_p and passes through O . 2) Second, we find the position of O on X_p . 3) 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, 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 , make the equal angles with X_p (Fig. 4). We first

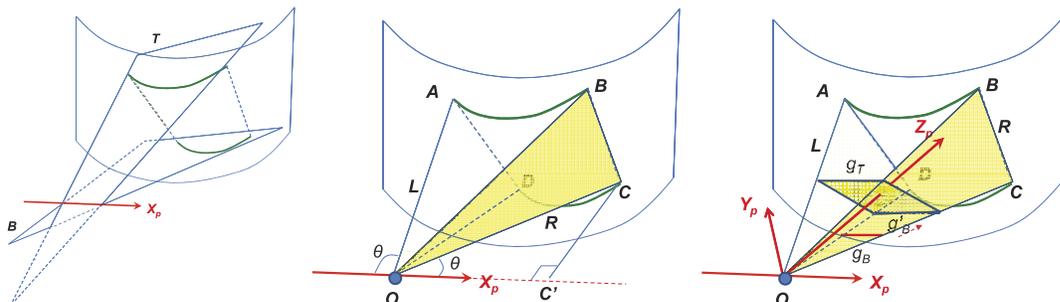


Fig. 4. Illustrations demonstrating the methods to find X_p (left), enter of the camera O (middle), and Y_p and Z_p (right).



Fig. 5. Registering from arbitrary viewpoint. Left: Viewpoint is on the left of the correction viewpoint. Middle: Viewpoint is the same as the correction viewpoint. Right: Viewpoint is on the right of the correction viewpoint. Please note the perspective distortion as the viewpoint deviates from the correction viewpoint.

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. 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 (Fig. 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 ODC . 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 (Fig. 4). Z_p is the normal to this plane and $Y_p = Z_p \times X_p$.

3.2.2 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 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 VR 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 multiviewing purposes (e.g., in museums, airports, and other public places). For such scenarios, the images from the projectors are pasted or seamlessly “wallpapered” on the display surface. In this section, we describe both these registrations.

3.3.1 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 then be used 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. The system needs to know the position of the viewer to correct the imagery for that viewpoint. The user will experience perspective distortion whenever the correction viewpoint is different from the user viewpoint as shown in Fig. 5.

3.3.2 Wallpapered Registration

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

Geometric registration involves finding a function that relates the projector coordinates (x, y) to the display coordinates (s, t) . Assuming the image to be wallpapered to have the same parametrization as of the display, we first cast a ray through each pixel (x, y) using the autocalibrated projector coordinates and find the 3D coordinates of its intersection with the display surface. Then, we find the corresponding (s, t) values and 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 the top and bottom boundaries of the surface are smooth curves, we implicitly assume a smooth surface.

However, the algorithm remains unchanged even if we have a piecewise smooth surface 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 the projectors on such surfaces, most CAVE setups use multiple projectors on each planar face, but do not allow overlap of projectors across different faces. This does not allow blending regions for color calibration [26], [27] and also makes it difficult to achieve automatic geometric calibration across the planar faces. Our method removes this restriction by allowing the projectors to overlap across the edges of the planar surfaces (Fig. 8).

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 [28]. 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 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 FAST RECALIBRATION

The position of the projectors, camera, and display can change easily in any flexible multiprojector setup. Even the shape of the display can change when using flexible materials. It is desirable to quickly recalibrate the display in such a scenario with minimal projection of patterns. Our method is particularly suited to achieve such fast recalibration. In this section, we describe how our method can be used for fast recalibration in the face of several such common changes in a multiprojector display setup.

During the content display we frequently capture images of the display. We call such images the *feedback image*. A change in the display shape or camera position can be detected by comparing the boundaries of the screen in a feedback image and the initial screen image used for calibration. Next, using the known geometry of the display and pose of the camera, we reproject the projected image on the camera plane to predict the captured image. We call this image the *predicted image*. We use simple image processing techniques to detect similar features (like corners and boundary curves) in both the feedback and predicted images and measure their deviation from each other. Whenever this deviation is more than a certain threshold we run a recalibration. We consider three scenarios for recalibration:

- Movement of one or more projectors.
- Movement of the camera.
- Movement or change in shape of the display.

In any one of these scenarios a partial recalibration is sufficient and is explained in the following sections. These

partial recalibration processes can be achieved with close to interactive rate. However, if multiple changes happen simultaneously, a full recalibration is required.

5.1 Recovering from Projector Movement

If the boundaries of the screen in the feedback image match the boundaries of the screen in the initial calibration image but the features do not match in the area of one or more projectors then only the pose of those projectors have changed. To recover from this the system projects the aforementioned pattern only from the moved projectors and recalibrates them using the autocalibration method in Section 3.2. Following this ray shooting is used from the recalibrated projectors to the known surface of the screen to find the projector to display correspondences.

5.2 Recovering from Change of the Camera Position

If the boundaries of the screen do not match the expected shape, either the camera has moved or the display shape/position has changed. In order to recover from this, first the system reruns the camera parameter estimation and screen geometric reconstruction which require only an image of the screen and take a few seconds to complete. Following this, if the recovered shape of the display is similar to the previously recovered shape then only the camera position has changed. Since a change in the camera position does not effect the registration of the display, nothing needs to be done unless a consequent change in the projector position is detected as in Section 5.1.

5.3 Recovering from Change of the Display Shape and Position

If the recovered shape of the display is found to be different but the camera position is still the same, a new coordinate system is first redefined based on the different shape and position of the display. This new coordinate system can be related to the old coordinate system by the rotation and translation which also relate the newly recovered camera coordinate system to the old camera coordinate system. Using this, the system can recover the position and orientation of the projectors in the new coordinate system without projecting any patterns. Since the projectors are precisely calibrated, their correspondences with the display coordinate system can be easily recovered to register them on this modified display.

6 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 algorithm takes about 6 minutes before dimension reduction and about 1 minute after that. The dimension reduction improves the time for recovering the camera properties from 5 minutes to 6-20 seconds. The display shape

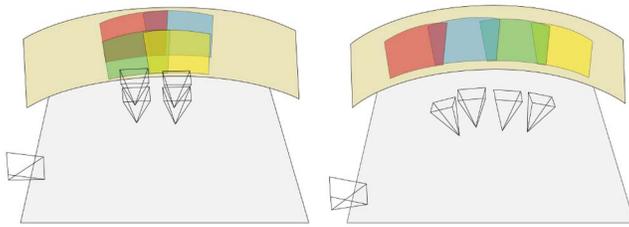


Fig. 6. The estimated setup (camera, 3D display and projectors) using our algorithm for the 2×2 array (left) and 1×4 array (right) of four projectors.

reconstruction takes about 20 seconds and the autocalibration of each projector takes about 10 seconds.

One needs to play with the weights w_r , w_c , and w_{lin} . Empirically we found that the weights of $w_r = 3$, $w_c = 1$, and $w_{lin} = 1$ work reasonably well for most scenarios.

Figs. 8 and 6 provide a visualization of the estimated camera, display, and projector locations and orientations in 3D using our algorithm for the two different setups. Fig. 7 shows the error between the reconstructed top and bottom curves of the display. They coincide demonstrating the accuracy of our method.

The most coveted feature of our method is its robust geometric registration. The accuracy of our method is demonstrated by a robust geometric registration. Empirically, we have seen a maximum misregistration of less than a pixel. Figs. 9, 10, and 11 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 [29], [16] and the registration with respect to a single viewpoint of the calibrating

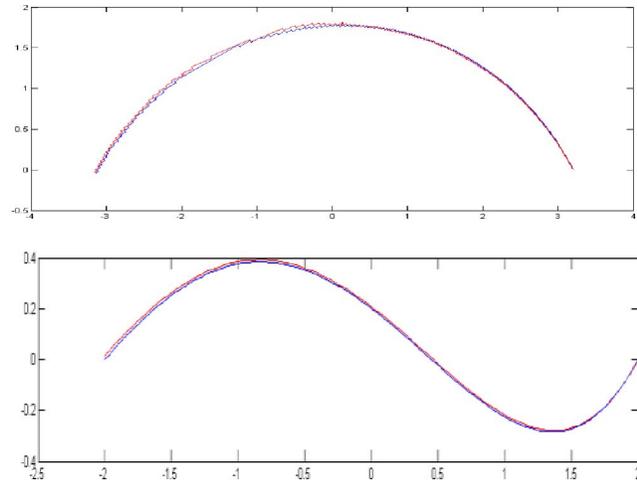


Fig. 7. Top: In the real system, red and blue are the estimated 3D top and bottom curves of the display. Bottom: In simulation, the blue curve is the original display surface curve and the red curve is the estimation provided by our method.

camera [9]. In addition to the obvious misregistrations, the shape of the display shows that wallpapering 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 [1], [14]. Photometric seams can be further removed by using [26].

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 as in [10]. This was supported by five poles to which the styrene sheet was attached (Fig. 12). The shape of the profile curve of this extruded display can be changed by simply changing the

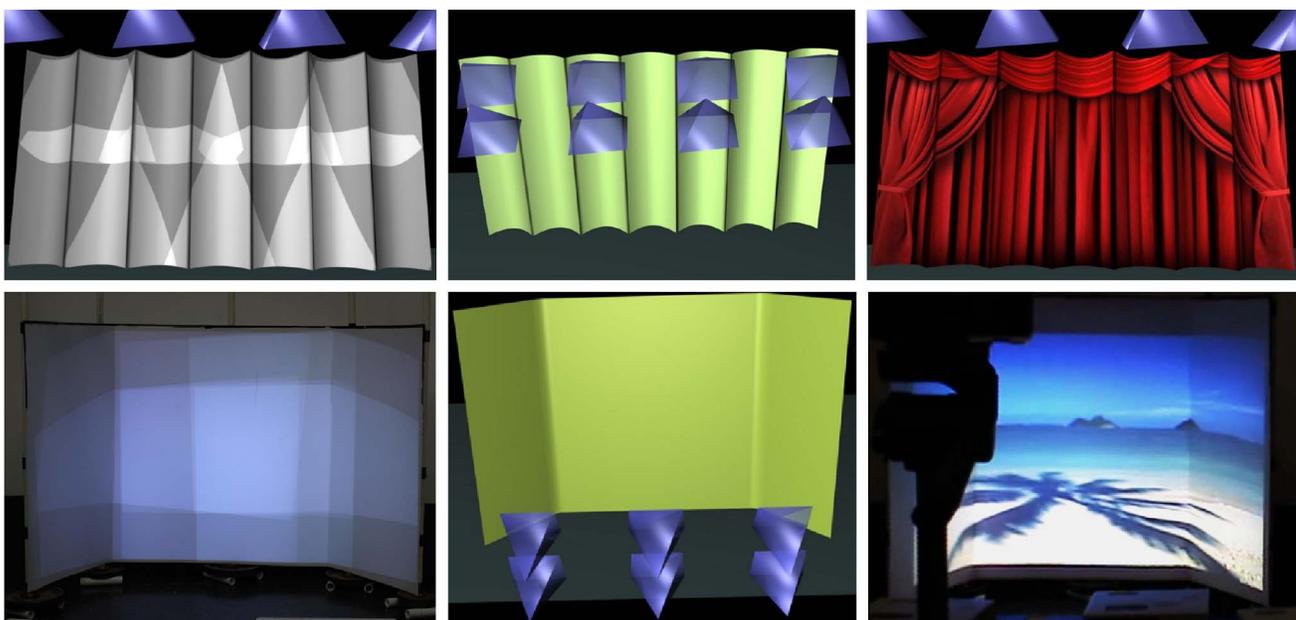


Fig. 8. Top: A simulated extruded piecewise smooth display with an array of 2×4 projectors with all the projectors projecting white (left), reconstruction of the display and projectors (middle), and with a registered image (right). Bottom: A real rudimentary CAVE with an array of 2×3 projectors. Coverage area of the projectors (left), reconstruction of the display and projectors (middle), and a registered image on the display (right) are shown in the figure.

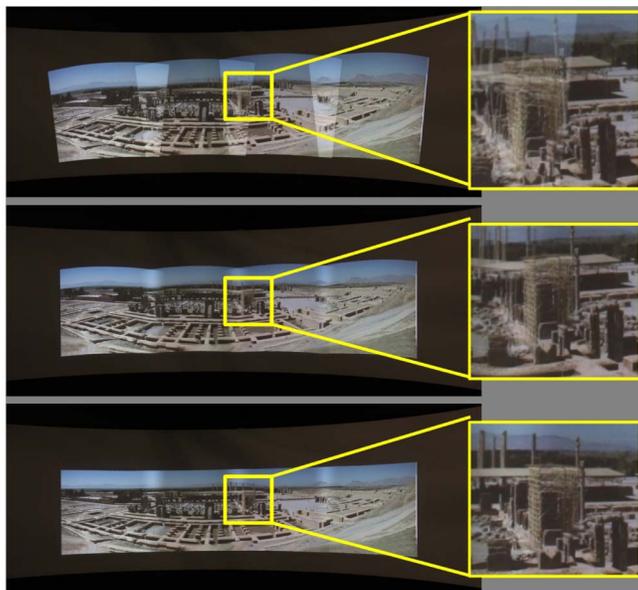


Fig. 9. These images show our geometric registration for the panoramic configuration. The images of Persepolis from top to bottom show a naive method, homography, and our method. Please zoom-in to see the details.

position of the poles. Fig. 12 illustrates the accuracy of our autocalibration on such a display.

We also demonstrate the extension of our algorithm to CAVEs. 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. Fig. 8 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 subpixel accuracy—but see 2-3 pixels misregistration, as shown in the video. However, in simulation, we can achieve the same subpixel accuracy as for our smooth extruded surface. We also tested our method on several novel piecewise smooth vertically extruded surfaces in simulation, one of which is illustrated in Fig. 8. In these general cases also our method shows subpixel accuracy. As shown in Fig. 8, such surfaces can be used for several edutainment and digital signage applications.

When a projector is moved after autocalibration, 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. Also a change in the camera position or the display shape can be recovered in 20 seconds.

6.1 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 2. 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 divided by the original value. We've shown the accuracy before and after dimension reduction. Note that even though dimension reduction is only for screen

reconstruction, the later steps use the information from this step and therefore get affected by its higher accuracy.

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 2. 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 tiled displays on piecewise smooth vertically extruded surfaces 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. The accuracy of reconstructing the display curves are evaluated over extrusions of different shapes. One example is shown in Fig. 7.

6.2 Discussion

In this section, we discuss some of the issues pertinent to the implementation of our method.

6.2.1 Nonperfect Extruded Surfaces

Most screens designed for commercial purposes are quite rigid infrastructure as is the screen we used for this work. We studied the effect of small deviation from extruded surface on the estimation of camera parameters and geometric registration in simulation. We performed several experiments on smooth and piecewise smooth surfaces and also applied our dimension reduction technique in the simulation. The deviation is simulated using the same metric as is used to measure the accuracy of estimating the curves. The results in Fig. 13 show that the surface need not be perfectly extruded. 4 to 6 percent deviation from thereof results in less than 1 to 2 pixels misregistration.

6.2.2 Nonlinear Projector Distortions

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 than 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 that the second and third dimensions of the space which encompass all the points are very small and therefore the points are along a line. This is a degenerate case since a plane can not be fit robustly to such points. A high fourth eigenvalue indicates that all the three dimensions of the space which encompass the points are large and therefore the points do not lie on a plane due to the presence of radial distortion in the

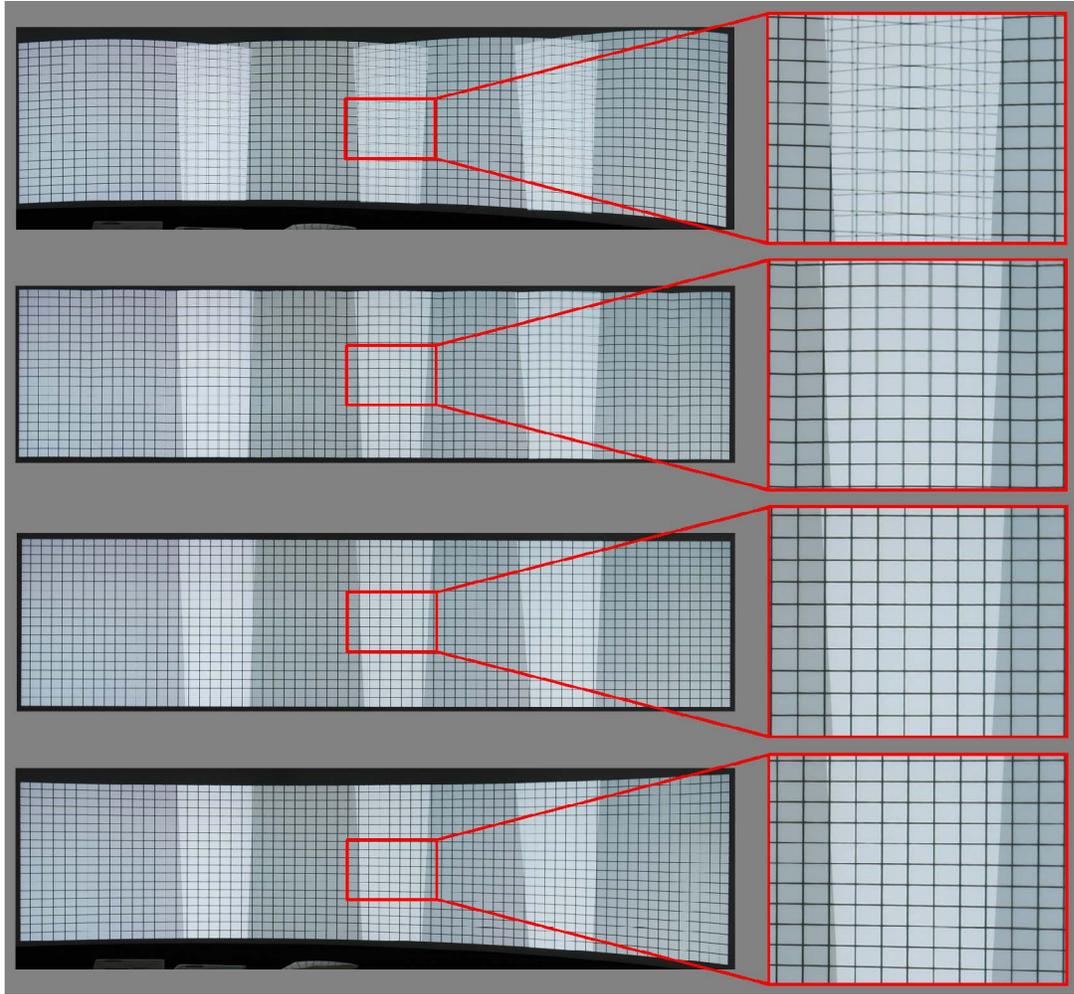


Fig. 10. These images show the accuracy of our geometric registration for the panoramic configuration. The images, from top to bottom, show homography, piecewise linear method with a sparse set of correspondences, piecewise linear method with a dense set of correspondences, and our method. Please note that the piecewise linear images are only correct from a single viewpoint and show perspective distortion from any other viewpoint.

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 their third and fourth 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 fourth 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.

6.2.3 Convergence

Before dimension reduction, our nonlinear optimization was often slow. More importantly, it sometimes converged to a local minima, especially when the initialization was far away from the solution. Convergence rate of our method is greatly improved after the dimension reduction of the search space from seven to three. Empirically, our optimization has never converged to a wrong solution after dimension reduction. We performed an extensive simulation and analyzed the convergence rate (Fig. 14) and time

(Fig. 15) before and after dimension reduction with respect to the distance of the initial guess and the correct answer. Following the dimension reduction, convergence is always achieved. Also, note an order of magnitude decrease in convergence time. To compute the distance between the correct answer and the one given by the optimization, we've used Euclidean distance. This distance is in a 7-dimensional space before dimension reduction and in a 3-dimensional space after that. However, we normalized the range of values for all of the variables to have fair comparison.

7 CONCLUSION

In summary, we have presented the first work to autocalibrate projectors on vertically extruded surfaces without using display to camera correspondences. Our projector autocalibration 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 autocalibration 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.

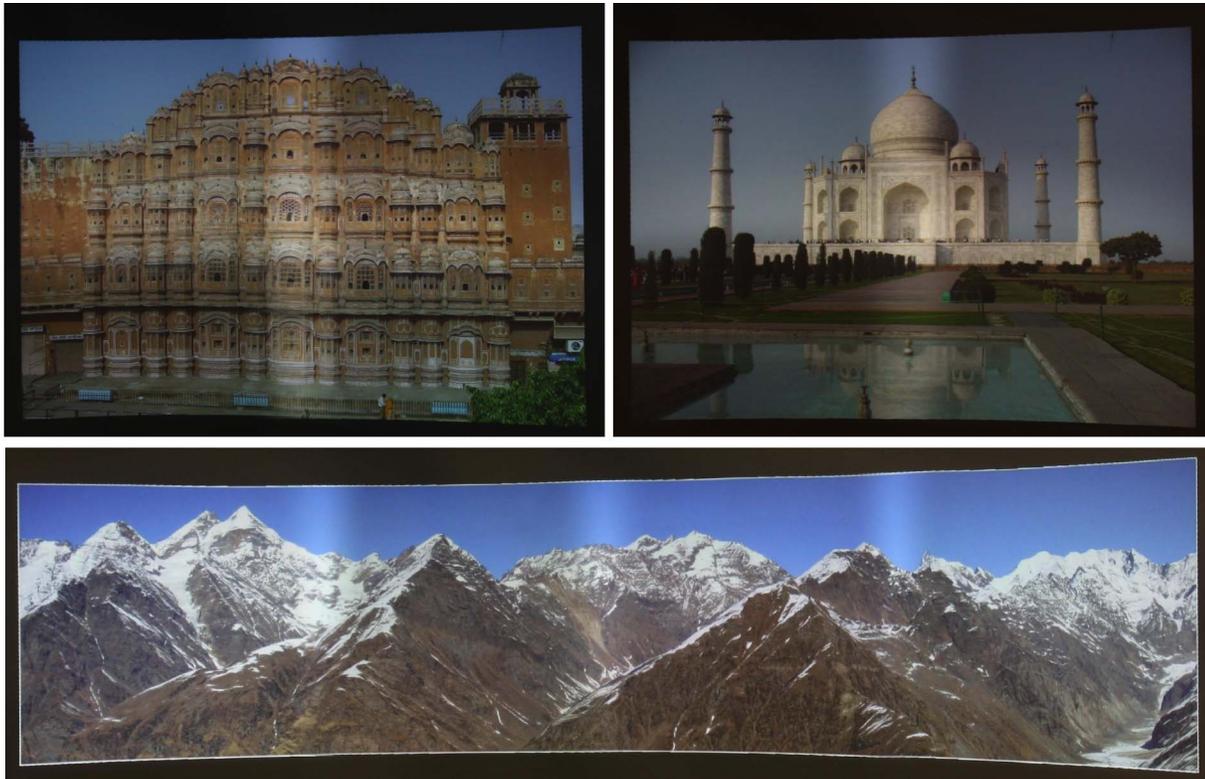


Fig. 11. Top: Geometric registration on our 2×2 projector display using our algorithm. Bottom: A registered and wall papered panorama of the Himalayas in the panoramic setup. Please zoom in to see the quality of registration.

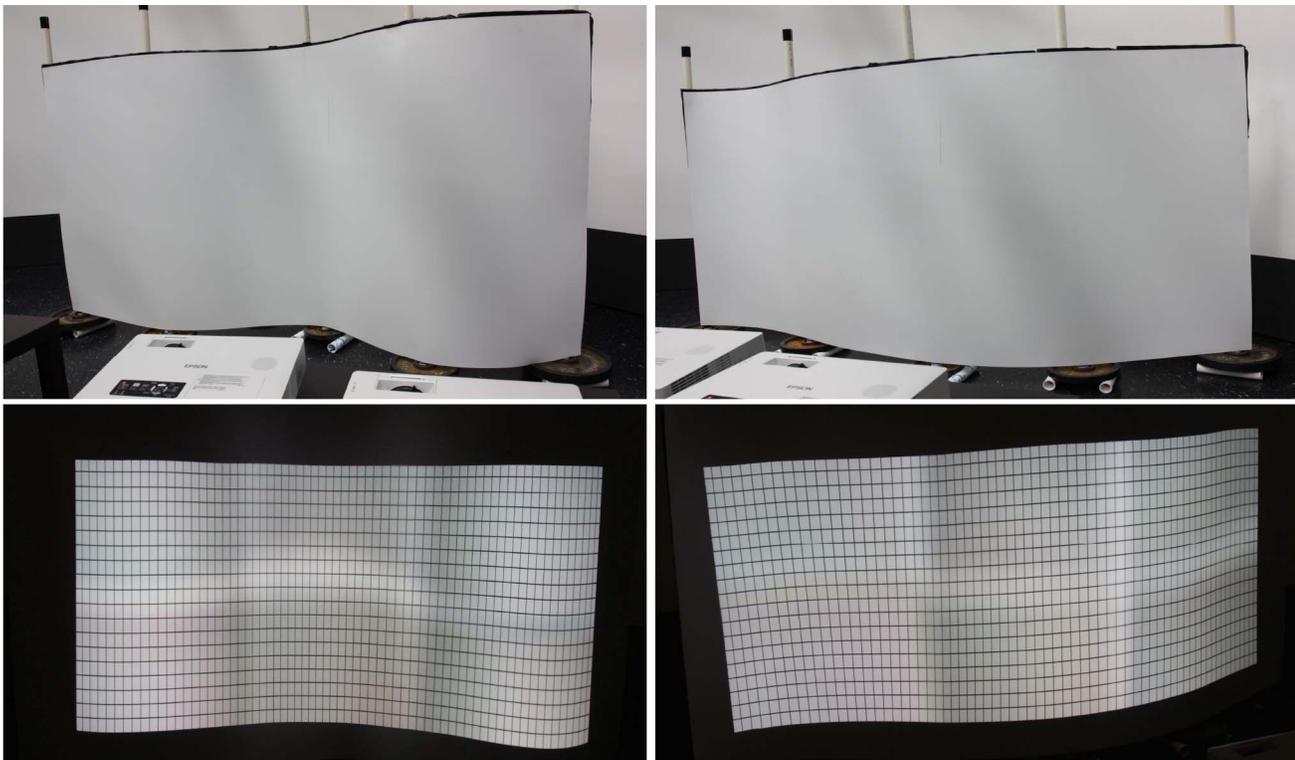


Fig. 12. On the left: Our flexible screen. On the right: Geometric registration on our flexible screen with two different shapes. Please zoom in to see the quality of registration.

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

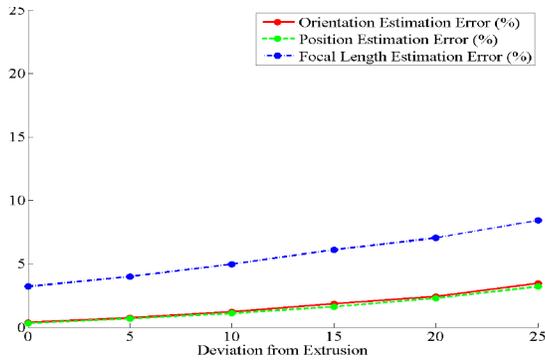


Fig. 13. The plot shows the pixel misregistration as the surface deviates from being an extruded surface. This shows that if we assume a roughly tiled configuration and can tolerate 1 or 2 pixel misregistration, we can allow 4 and 6 percent deviation from an extruded surface, respectively.

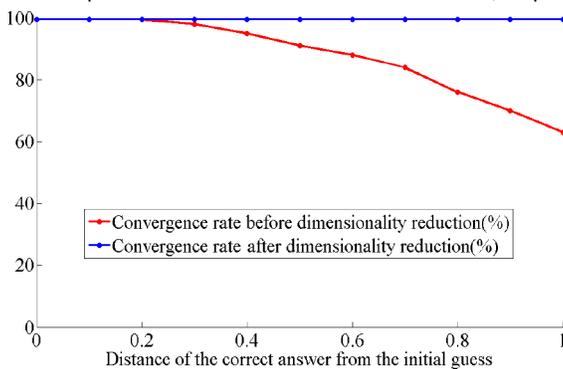


Fig. 14. Comparison of the convergence rate of the optimization with respect to the distance of the initial guess from the correct answer before and after dimension reduction.

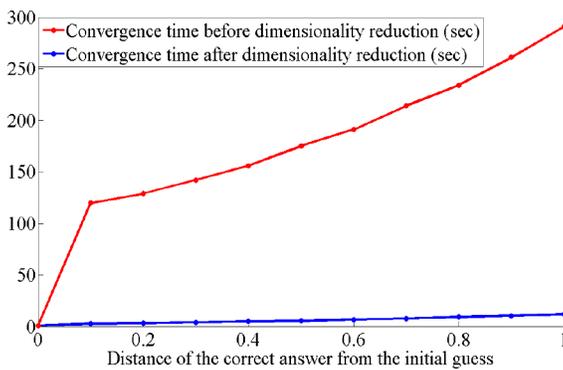


Fig. 15. Comparison of the convergence time of the optimization with respect to the distance of the initial guess from the correct answer before and after dimension reduction. This also includes convergence time of the simulations which converged to a local minima.

surfaces. Reasonable deviation from perfectly extruded surfaces will allow lower precision in the screen manufacturing, making these displays more affordable.

ACKNOWLEDGMENTS

We would like to acknowledge our funding agencies the US National Science Foundation (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.S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs, "Multi-Projector Displays Using Camera-Based Registration," *Proc. Conf. Visualization*, pp. 161-522, 1999.
- [2] D. Cotting, M. Naef, M. Gross, and H. Fuchs, "Embedding Imperceptible Patterns into Projected Images for Simultaneous Acquisition and Display," *Proc. Int'l Symp. Mixed and Augmented Reality*, pp. 100-109, 2004.
- [3] D. Cotting, R. Ziegler, M. Gross, and H. Fuchs, "Adaptive Instant Displays: Continuously Calibrated Projections Using Per-Pixel Light Control," *Proc. Eurographics*, pp. 705-714, 2005.
- [4] T. Johnson and H. Fuchs, "Real-Time Projector Tracking on Complex Geometry Using Ordinary Imagery," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1-8, 2007.
- [5] J. Zhou, L. Wang, A. Akbarzadeh, and R. Yang, "Multi-Projector Display with Continuous Self-Calibration," *Proc. Fifth ACM/IEEE Int'l Workshop Projector-Camera Systems*, pp. 3:1-3:7, 2008.
- [6] T. Johnson, G. Welch, H. Fuchs, E. la Force, and H. Towles, "A Distributed Cooperative Framework for Continuous Multi-Projector Pose Estimation," *Proc. IEEE Virtual Reality Conf. (VR '09)*, pp. 35-42, 2009.
- [7] M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, and D. Tanguay, "Practical Methods for Geometric and Photometric Correction of Tiled Projector," *Proc. Conf. Computer Vision and Pattern Recognition Workshop (CVPRW '06)*, p. 5, 2006.
- [8] W. Sun, I. Sobel, B. Culbertson, D. Gelb, and I. Robinson, "Calibrating Multi-Projector Cylindrically Curved Displays for "Wallpaper" Projection," *Proc. Fifth ACM/IEEE Int'l Workshop Projector-Camera Systems (PROCAMS)* pp. 1:1-1:8, 2008.
- [9] M.S. Brown and W.B. Seales, "A Practical and Flexible Tiled Display System," *Proc. 10th Pacific Conf. Computer Graphics and Applications*, pp. 194-203, 2002.
- [10] B. Sajadi and A. Majumder, "Markerless View-Independent Registration of Multiple Distorted Projectors on Extruded Surfaces Using an Uncalibrated Camera," *IEEE Trans. Visualization and Computer Graphics (Visualization)*, vol. 15, no. 6, pp. 1307-1316, Nov./Dec. 2009.
- [11] A. Rajj and M. Pollefeys, "Auto-Calibration of Multi-Projector Display Walls," *Proc. 17th Int'l Conf. Pattern Recognition (ICPR '04)*, pp. 14-17, 2004.
- [12] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines, "Ilamps: Geometrically Aware and Self-Configuring Projectors," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 809-818, 2003.
- [13] A. Rajj, G. Gill, A. Majumder, H. Towles, and H. Fuchs, "Pixelflex 2: A Comprehensive Automatic Casually Aligned Multi-Projector Display," *Proc. IEEE Int'l Workshop Projector-Camera Systems (PROCAMS)*, 2003.
- [14] R. Yang, D. Gotz, J. Hensley, H. Towles, and M.S. Brown, "Pixelflex: A Reconfigurable Multi-Projector Display System," *Proc. Visualization*, pp. 167-174, 2001.
- [15] E. Bhasker, R. Juang, and A. Majumder, "Registration Techniques for Using Imperfect and Partially Calibrated Devices in Planar Multi-Projector Displays," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1368-1375, Nov./Dec. 2007.
- [16] H. Chen, R. Sukthankar, G. Wallace, and K. Li, "Scalable Alignment of Large-Format Multi-Projector Displays Using Camera Homography Trees," *Proc. IEEE Visualization (VIS '02)*, pp. 339-346, 2002.
- [17] E.S. Bhasker, P. Sinha, and A. Majumder, "Asynchronous Distributed Calibration for Scalable and Reconfigurable Multi-Projector Displays," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1101-1108, Sept. 2006.
- [18] D.G. Aliaga and Y. Xu, "Photogeometric Structured Light: A Self-Calibrating and Multi-Viewpoint Framework for Accurate 3D Modeling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1-8, 2008.
- [19] D.G. Aliaga, "Digital Inspection: An Interactive Stage for Viewing Surface Details," *Proc. ACM Symp. Interactive 3D Graphics*, pp. 53-60, 2008.
- [20] R. Raskar, J.V. Baar, T. Willwacher, and S. Rao, "Quadric Transfer Function for Immersive Curved Screen Displays," *Proc. Eurographics*, 2004.
- [21] R. Yang and G. Welch, "Automatic Projector Display Surface Estimation Using Every-Day Imagery," *Proc. Ninth Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2001.

- [22] S. Zollmann, T. Langlotz, and O. Bimber, "Passive-Active Geometric Calibration for View-Dependent Projections onto Arbitrary Surfaces," *Proc. Workshop Virtual and Augmented Reality (AR/VR)*, 2006.
- [23] M. Brown, A. Majumder, and R. Yang, "Camera-Based Calibration Techniques for Seamless Multiprojector Displays," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 2, pp. 193-206, Mar./Apr. 2005.
- [24] B. Sajadi and A. Majumder, "Auto-Calibration of Cylindrical Multi-Projector Systems," *Proc. IEEE Virtual Reality (VR) Conf.*, pp. 155-162, 2010.
- [25] N. Snavely, S.M. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in Third," *Proc. ACM SIGGRAPH*, pp. 835-846, 2006.
- [26] A. Majumder and R. Stevens, "Perceptual Photometric Seamlessness in Projection-Based Tiled Displays," *ACM Trans. Graphics*, vol. 24, pp. 118-139, Jan. 2005.
- [27] A. Majumder and R. Stevens, "Color Nonuniformity in Projection-Based Displays: Analysis and Solutions," *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 2, pp. 177-188, Mar./Apr. 2004.
- [28] G. Ferrari-Trecate and M. Muselli, "A New Learning Method for Piecewise Linear Regression," *Lecture Notes in Computer Science*, J. D. Dorransoro, eds., vol. 2415, pp. 444-449, 2002.
- [29] M. Ashdown, M. Flagg, R. Sukthankar, and J.M. Rehg, "A Flexible Projector-Camera System for Multi-Planar Displays," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition (CVPR '04)*, pp. 165-172, 2003.



Behzad Sajadi received the bachelor's degree from Sharif University of Technology in 2006. He is working toward the PhD degree at the Department of Computer Science in University of California, Irvine, since 2007. His main research area is Computer Graphics and Visualization with particular interest in multiprojector displays and computational projectors and cameras. He has published several works on geometric and photometric registration of multiprojector 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. He is a student member of the IEEE.



Aditi Majumder received the PhD degree from Department of Computer Science, University of North Carolina, at Chapel Hill, in 2003. She is an associate professor at the Department of Computer Science in University of California, Irvine. Her research area is computer graphics and vision, image processing with primary focus on multiprojector displays. Her research aims to make multiprojector 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 coauthor of the book "Practical Multi-Projector Display Design." She was the program and general cochair 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 multiprojector 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 US National Science Foundation (NSF) CAREER award in 2009 for Ubiquitous Displays Via a Distributed Framework. She is a member of the IEEE.

► **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**