Eurographics/ IEEE-VGTC Symposium on Visualization 2010
G. Melançon, T. Munzner, and D. Weiskopf
(Guest Editors)

Volume 29 (2010), Number 3

Scalable Multi-view Registration for Multi-Projector Displays on Vertically Extruded Surfaces

Behzad Sajadi[1] and Aditi Majumder[1]

[1] University of California, Irvine

Abstract

Recent work have shown that it is possible to register multiple projectors on non-planar surfaces using a single uncalibrated camera instead of a calibrated stereo pair when dealing with a special class of non-planar surfaces, vertically extruded surfaces. However, this requires the camera view to contain the entire display surface. This is often an impossible scenario for large displays, especially common in visualization, edutainment, training and simulation applications. In this paper we present a new method that can achieve an accurate geometric registration even when the field-of-view of the uncalibrated camera can cover only a part of the vertically extruded display at a time. We pan and tilt the camera from a single point and employ a multi-view approach to register the projectors on the display. This allows the method to scale easily both in terms of camera resolution and display size. To the best of our knowledge, our method is the first to achieve a scalable multi-view geometric registration of large vertically extruded displays with a single uncalibrated camera. This method can also handle a different situation of having multiple similarly oriented cameras in different locations, if the camera focal length is known.

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

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms; I.4.0 [Image Processing and Computer Vision]: General˚Uimage displays—;

1. Introduction

Tiled multi-projector displays provide life-size high-resolution imagery for many applications like visualization, edutainment, training and simulation. Curved surfaces (e.g. cylinders) are better in creating immersive experiences than planar displays. They are often preferred over a piecewise planar display surface (e.g. CAVE) where the sharp corners reduce the sense of immersion. However, till recently, it was difficult to register multiple projectors on such curved displays automatically and easily. Complex registration techniques used calibrated stereo camera pair to extract the shape of the display to recover the 2D display parametrization [RBWR04, CNG*04, CZGF05, JF07, ZWA*08, JWF*09, ZLB06], critical to wall-paper the image on the display so that it looks acceptable from multiple viewpoints. When single uncalibrated camera was used, physical fiducials were used on the rim of the display to provide the display parametrization [HCS*06, SSC*08]. When avoiding markers and using a single camera, recovering the 2D display parametrization, and hence wall-papering, was not possible. Thus, it was only possible to achieve a registration that is correct from the single viewpoint of the camera and shows perspective distortions when viewed from other viewpoints [YWB02, BMY05].

Recently, it has been shown that when considering a special class of non-planar surfaces – vertically extruded surfaces – it is possible to achieve a wall-papered registration without using markers with a single uncalibrated camera, even with imperfect distorted projectors [SM09], if the aspect ratio of the display is known. However, this method needs the camera to see the entire display in a single view. Thus, it does not scale well for large immersive displays, which are becoming increasingly common as the size of data explodes in an exponential manner.

Main Contribution: This paper presents a new algorithm

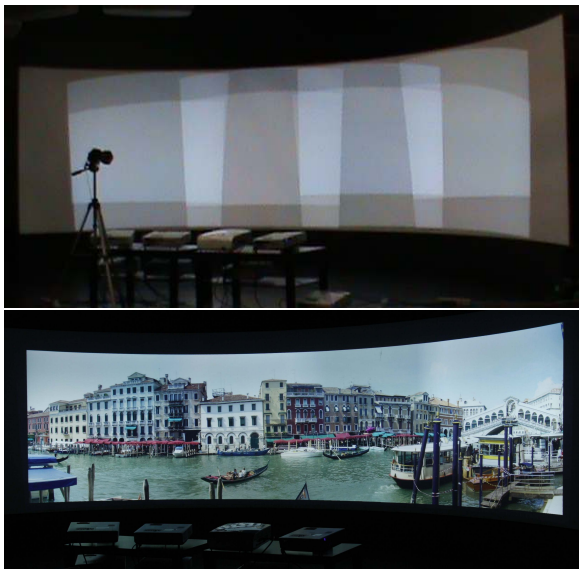

Figure 1: *Top: This shows our 8 projector (2×4 array) cylindrical display setup. The camera used to register the display is also shown in the image. Bottom: The same display registered using 7 camera views. Please zoom in to see the registration quality.*

that uses multiple views from an uncalibrated camera (instead of a single view as in [SM09]) to register multiple projectors on a vertically extruded surface of known aspect ratio. A camera, mounted on a pan-tilt unit(PTU), is panned and titled to capture multiple views of the display. For each camera view, multiple images are captured by projecting patterns from different projectors. We design a novel method that uses these images to recover the camera pose and orientation for all the different views and also the 2D display parametrization. This allows us to register the multiple projectors on the display in a wall-papered manner even when a single view from the camera cannot see the entire display. Wall-papered registration is not correct from any single view point, but provides a natural way for multiple users to look at the display. To the best of our knowledge, this is the first work that can achieve geometric registration on a non-planar display using a single uncalibrated camera, even when multiple views from the camera is used to see only parts of the display at a time. Using multiple views from the same position provides enough constraints to solve the problem unambiguously. However, if the focal length of the camera is known, we still have enough constraints to handle translated views (but not rotated) and hence multiple similarly oriented cameras in different positions.

2. Related Work

Our work is related to a body of literature on geometric registration of multi-projector displays for *non-planar surfaces*. When considering such displays, especially arbitrary ones, using *multiple cameras* is necessary for reconstructing the shape of the non-planar surface. Raskar et al. in [RBY*99] 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 geometric registration. Aliaga et al. in [AX08, Ali08] 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 [RBWR04] 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 stretching of projected pixels during registration. All the above methods achieve a pre-calibration that takes a few minutes.

A complementary set of techniques focus on continuous image registration during display time for dynamic changes in the display shape and movement in projectors. Yang and Welch [YW01] use the projected content (as opposed to special patterns) at display time to automatically estimate the shape of the display surface and account for changes in its shape over time. Using a projector augmented by two stereo cameras, Cotting et al. [CNG*04, CZGF05, JF07] estimate the shape of the display surface and the pose of a single projector continuously over time by embedding imperceptible patterns into projected imagery. Zhou et al. [ZWA*08] achieves the same by tracking displayed image features. Johnson et al. [JWF*09] 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. [ZLB06] present a hybrid technique that compensates for small changes in the display configuration using optical flow, and resorts to active structured light projection when optical flow becomes unreliable.

Our work belongs to the body of literature that tries to avoid the complexity of using multiple cameras for non-planar screens. Brown et al. [YWB02, BMY05] avoid reconstructing the display geometry by registering the multiple projectors with respect to the single point of view of a camera. More recently, [HCS*06, SSC*08] try to avoid this view-dependency in registration for the special case of cylindrical surface by relating its 2D parametrization with the camera image space without reconstructing the 3D display surface. A precisely calibrated physical pattern is pasted along the top and bottom rims 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 sample only the rims of the display,

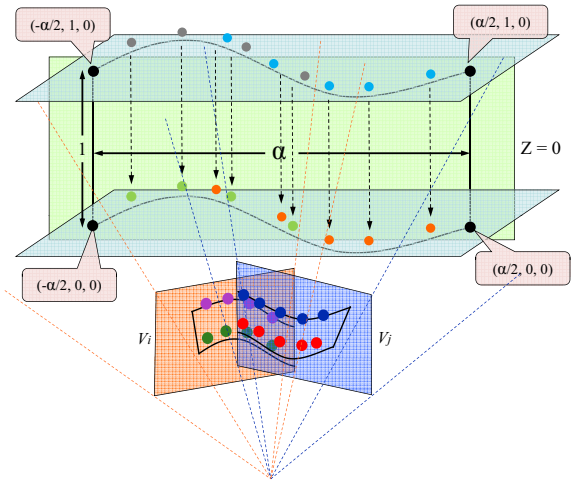

Figure 2: *This figure illustrates the 3D coordinates of a cylindrical display with aspect ratio α and two camera views V_i and V_j. The top and bottom 3D curves lie on the $Y = 0$ and $Y = 1$ planes respectively. The 3D coordinates of the four corners of the display are $(-\frac{\alpha}{2},1,0)$, $(\frac{\alpha}{2},1,0)$, $(\frac{\alpha}{2},0,0)$, and $(-\frac{\alpha}{2},0,0)$ (clockwise from top left). The blue and purple points on the image plane of V_i and V_j are the sampled 2D points on the top boundary. These are back projected to give the cyan and gray 3D points. These are then translated by (0, -1, 0) to give the orange and lime points respectively. These are then reprojected to give the red and green points respectively.*

these methods result in distortions or stretching, especially towards the middle of the display surface. More importantly, since the display surface in not reconstructed in both these methods, registering images from an arbitrary viewpoint (for e.g. in a virtual reality system), is not possible.

Our work draws inspiration from a more recent work which uses a single uncalibrated camera, does not need to use physical fiducials, and can still achieve a registration of multiple projectors [SM09]. Unlike previous work where the registration looks correct from a single view-point, [SM09] achieves a wall-papered registration. This is achieved by assuming that the non-planar surface is not completely arbitrary in shape, but vertically extruded, and the aspect ratio of the planar rectangle formed by the four corners of the surface is known. This covers a range of useful surfaces (like cylinders and CAVES) that are commonly used for immersive displays. In our work we allow multiple panned and tilted camera views from the same position to see parts of the display at a time, in contrast to a single camera view seeing the entire display in [SM09]. Our novel algorithm can recover the pose and orientations of the different camera views and the 2D display parametrization, even from the partial views, resulting in a wall-papered registration.

Our work achieves the same goal as a body of work for planar displays that use multiple cameras. When a planar display is too big for using a single camera to register the projectors [RP04, RvBB*03, BJM07, RGM*03, YGH*01], multiple cameras were used to achieve registration. Chen et al. [CSWL02] 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, [BSM06] presents 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. Our method focuses on the more complex case of non-planar vertically extruded displays. Hence, we cannot handle all types of camera configurations, only multiple panned and titled views from the same camera at the same position. However, if the camera focal length is known, we can handle multiple views from multiple locations, but of the same orientation.

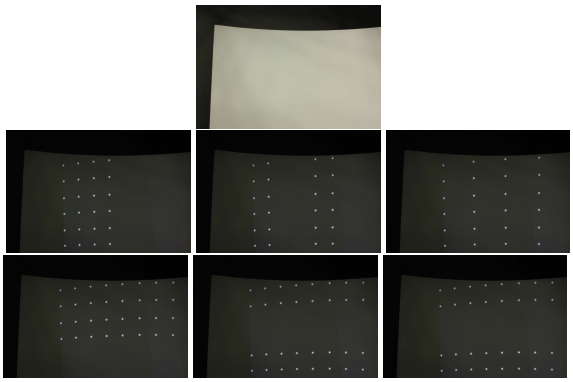

Figure 3: *This shows the images taken from one of the multiple camera views for one of the projectors (all except the top one). We use 20 blobs per projector and hence we need 6 frames to detect the binary coded IDs of the blobs. We have similar images for all projectors that fall within the FOV of this camera view. In addition, we take one picture of the screen with no projectors turned on (top).*

3. Algorithm

Let the display surface, the image planes of the camera and the projector be parametrized by (s,t), (x,y) and (u,v) respectively. We denote the 3D coordinates of the point at (s,t) on the display by $(X(s,t),Y(s,t),Z(s,t))$. Since the display is a vertically extruded surface, the four corners of the display lie on a planar rectangle, whose aspect ratio α is known. We define the world 3D coordinate 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 $Z = 0$. We assume that the top and bottom curves of the vertically extruded surface lie respectively on $Y = 1$ and $Y = 0$ planes. Hence, $Y(s,0) = 0$ and $Y(s,1) = 1$. Further, these two curves are identical except for a translation in the Y direction. Therefore, $\forall s$, $(X(s,0),Z(s,0)) = (X(s,1),Z(s,1))$. This is illustrated in Figure 2.

We assume that the camera is placed on a pan-tilt unit. It is panned and titled (but not translated) to N different poses to capture multiple views of the display, in each of which only

a small part of the display is visible. We assume that zoom of the camera is not changed across these views. This allows us to assume that while the extrinsic parameters of the camera change across the views, the intrinsic parameter remains constant. We assume considerable overlap between adjacent views (more on overlaps in Section 4.1). We assume M projectors in the display. In each camera view, we capture K images from each projector that falls within the field of view of the camera, each image comprising of a set of blobs (a gray spot). In addition, we capture the display with no projector turned on from each camera view. We assume that our camera is a linear device with no lens distortion. However, our projectors can have non-linear distortion. These input images are illustrated in Figure 3.

The goal of wall-papered registration is to define a function from (u,v) projector coordinates to the (s,t) display coordinates. Our method follows three steps to achieve this. First, we *recover the parameters of the N camera views* from the correspondences provided by the captured images (Section 3.1). This involves three steps: (a) we first use a non-linear optimization to recover the common intrinsic parameter of the camera across multiple views (Section 3.1.1); (b) then we use a graph based linear optimization to find the relative extrinsic parameters of each camera view with respect to a reference view (Section 3.1.2); and (c) finally, we use a non-linear optimization to find the pose and orientation of the reference camera with respect to the display which allows us to recover the pose and orientation of all camera views with respect to the display (Section 3.1.3). Next, we use the recovered camera parameters to *reconstruct the 3D shape of the display* and define 2D parametrization of the display surface (Section 3.2). Finally, we define the relationship between the projector and display coordinates to achieve the geometric registration (Section 3.3). Assuming the image parametrization to be identical to the display parametrization, our method automatically achieves a seamless wall-papered geometric registration. Each of the above steps are described in detail in the following sections.

3.1. Multiple-View Camera Property Reconstruction

Most cameras today have the principal center at the center of the image, no skew between the image axes and square pixels. Using these assumptions, as in [SSS06], we express the intrinsic matrix of a camera, K_c, as

$$K_c = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

A large number of image formats like jpg or tiff store EXIF tags for images which provide some of the camera parameters used during the capture. We use this focal length as input to our method in this step. 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. The additional input to this step is the known aspect ratio α of the planar rectangle formed by the four corners of the display. Thus the 3D coordinates of the four corners of the rectangle are given by $(-\frac{\alpha}{2},1,0)$, $(\frac{\alpha}{2},1,0)$, $(\frac{\alpha}{2},0,0)$, and $(-\frac{\alpha}{2},0,0)$ (clockwise from top left).

For the ith camera view, $i \in [1 \ldots N]$, denoted by V_i, the camera calibration matrix that relates the 3D coordinates with the 2D camera image coordinates (x,y) is given by a 3×4 matrix K_cC_i where K_c denotes the intrinsic parameters and C_i denotes the extrinsic parameters including pose and orientation. We assume V_1 to be the reference view and hence the calibration matrix for V_1 is given by K_cC_1. For all other $N-1$ views, since we assume only a rotation from V_1 and no change in the intrinsic properties of the camera,

$$C_i = K_cC_1R_i \quad (2)$$

where R_i denotes the relative rotation of C_i from C_1.

For each projector $j, j \in [1..M]$, we use Q blobs for structured light projection. We project these blobs in a time sequential manner in $K = \lceil log(Q) \rceil$ frames [RBY*99, YGH*01]. When seen by a camera, the presence or absence of a blob in a particular temporal frame helps us to recover the binary coded id of the detected blobs in the camera image (See Figure 3). We use gaussian blobs and fast non-maximum suppression techniques during blob detection to achieve sub-pixel accuracy. It also provides robustness in the face of perspective distortion and vignetting effect of the camera assuring no false-positives. When the same blob is detected from two camera views, it provides a correspondence across these two camera views. Using this technique, for a pair of overlapping camera views V_i and V_j, we can recover a set of correspondences denoted by S_{ij}. Note that $S_{ij} = \emptyset$ when there is no overlap between two camera views. Given these sets of correspondences across all pairs of overlapping camera views, our goal is to recover the intrinsic parameter K_c, the relative rotation R_i of each camera with respect to the reference camera, and the extrinsic parameter for the reference camera view C_1 with respect to the global coordinate system defined by the screen. For recovering K_c and R_i, we use the images with the projectors turned on. However, for recovering C_1 we only need the images where no projector is on. After recovering K_c, R_i for every V_i, and C_1, we can find the pose and orientation C_i of the view V_i using Equation 2. In the following sections we describe each of these steps in detail.

3.1.1. Recovering K_c

Recovering the intrinsic parameter matrix K_c entails recovering the focal length f of the camera (Equation 1) that remains constant across the multiple views. We start with a reasonably good estimate of f from the EXIF tags and then use an angular constraint for pairs of correspondences across multiple views to converge to a more accurate estimate. To explain this process, lets consider the view V_i with center of projection (COP) at O_i (see Figure 4). Let us consider two points $A = (x_i, y_i)$ and $A' = (x_i', y_i')$ in V_i. We define the local camera coordinates of V_i assuming O_i to be at $(0,0,0)$

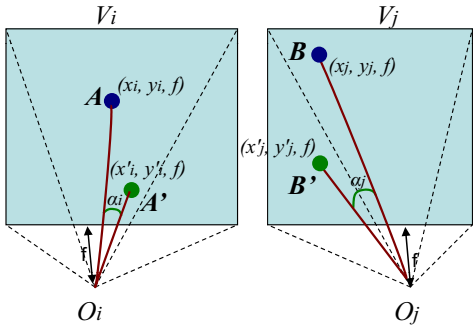

Figure 4: *This figure illustrates the constraint used to recover the focal length and hence the intrinsic parameter matrix of the camera.*

and image coordinate axes and a vector perpendicular to the image plane defining the camera orientation. The 3D coordinates of A and A' in the local coordinate of V_i are given by $A = (x_i, y_i, f)$ and $A' = (x_i', y_i', f)$ respectively and let the angle between O_iA and O_iA' be α_i. Let us consider the corresponding points B and B' respectively in V_j and angle subtended by these two points at the COP, α_j. Since the COP of both V_i and V_j are same and f has not changed across views, the angle $\alpha_i = \alpha_j$. Hence,

$$cos(\alpha_i) = \frac{O_iA \cdot O_iA'}{|O_iA||O_iA'|} = cos(\alpha_j) = \frac{O_jB \cdot O_jB'}{|O_jB||O_jB'|}. \quad (3)$$

Equation 3 results in a non-linear equation of degree 8. It is solved by Newton Raphson method to give f and hence K_c.

Every pair of correspondences across any two camera views can provide an estimate of f. However, some of these correspondences may be more reliable than others, as explained in the following paragraph. Hence, we design a metric to choose a smaller subset of more reliable correspondences and find f for each of those. The final f is estimated as an average of all the focal length estimates resulting from the correspondences.

We choose a subset of reliable pairs of correspondences as follows. The triangle O_iAA' should not be equal to the triangle O_jBB' to avoid ambiguity. This is assured when $|AA'|$ and $|BB'|$ are significantly different. Hence, we first reject a subset of the pairs of correspondences based on this criterion. To reduce noise in the estimated f, $|AA'|$ and $|BB'|$ and the ratio $|AA'|$ to that of $|BB'|$ should be large. Hence, we rank these pairs based on the metric $d = \min(|AA'|, |BB'|) \times \max(\frac{|AA'|}{|BB'|}, \frac{|BB'|}{|AA'|})$. Finally, we choose the pairs whose d is higher than a threshold D to estimate f. Since we begin with a reasonably good estimate of f, even a small set of reliable corresponding pairs (even 5) yields an accurate estimate.

3.1.2. Recovering R_i

In this step, our goal is to recover the rotation matrix R_i of V_i that gives its relative orientation with respect to the reference camera V_1. For this, we first need to find the relative rotation relating any two overlapping camera views V_i and V_j, R_{ij}, using the correspondences in S_{ij}. Since R_{ij} is a homography,

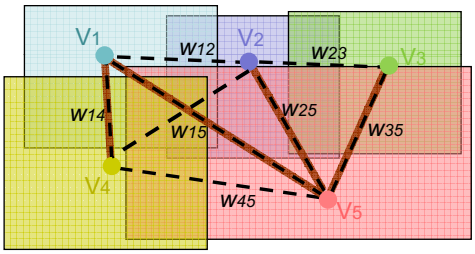

Figure 5: *This figure illustrates the recovery of R_i. Each camera view V_i is a node of the graph. A non-null $S_{i,j}$ forms an edge shown by black dashed lines. Each edge is associated with a weight w_{ij}. To find R_i relating any V_i to the reference camera V_1, we find the single source shortest path from V_1 shown by the red lines. Concatenation of the R_{ij}s on the path from V_1 to V_i in this graph provides R_i.*

we can achieve this by a linear least square optimization, as in [CSWL02, RvBB*03].

If we consider a graph whose nodes are given by the camera views $V_1 \ldots V_n$, then every non-null S_{ij} signifies an edge associated with R_{ij}, as illustrated in Figure 5. Note that multiple paths exist from V_1 to V_i in this graph. Concatenating the matrices R_{ij} along any of these paths would result in an estimate of R_i. However, the reliability of an edge and hence, the accuracy of the estimated R_{ij} associated with the edge, depends on the number and the spatial distribution of the correspondences in S_{ij}. Hence, we associate a weight w_{ij} with every edge to create a weighted graph. We want this weight to represent a measure of the inaccuracy of the R_{ij} estimated from the correspondences between camera view V_i and V_j and design it as follows.

The robustness of the estimate of R_{ij} depends on the relative spatial distribution of the corresponding blobs in the two camera views. Let us consider two pairs of corresponding points, A and A' in V_i which correspond to B and B' in V_j respectively. Robust estimates are assured if both $|AA'|$ and $|BB'|$ are large. Hence, we assign a robustness metric r to a pair of correspondence as the geometric mean of $|AA'|$ and $|BB'|$, i.e. $r = \sqrt{|AA'||BB'|}$. The sum of r over all possible pairs of correspondences between V_i and V_j gives an estimate of the robustness of R_{ij}. Let us denote this sum by L_{ij}. The inaccuracy in the estimate of R_{ij} is hence given by $\frac{1}{L_{ij}}$. To find a good R_i that relates V_i to V_1, we seek to find the shortest path from V_1 to V_i such that the product of the weights is minimized so that the path contains only low weighted edges. Since single source shortest path (SSSP) methods minimize the sum of the weights associated with each edge, we desire to use $ln(\frac{1}{L_{ij}})$ for weights to convert the product to sum. However, in this case w_{ij} can be negative. Negative weights favor longer paths which is more error-prone in practice. This is alleviated by a metric that assigns positive weight to each edge. So, we assign $w_{ij} = 1 + ln(max(L_{ij}) - ln(L_{ij}) = 1 + ln(\frac{\max L_{ij}}{L_{ij}})$. Addition of the constant assure that w_{ij} is always positive. We find the

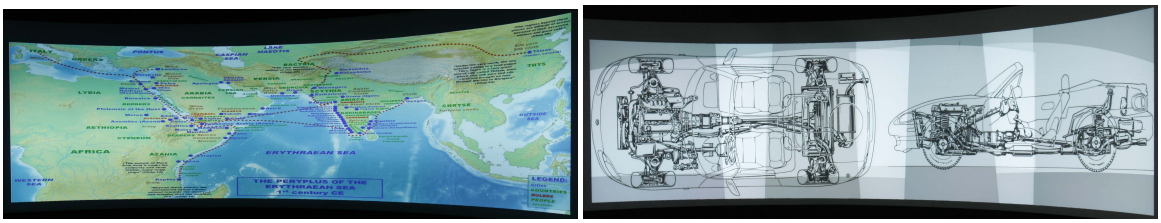

Figure 6: *Top: Our registered 8 projector display registered using 7 camera views (left) and 3 camera views (right). For the right one, we only registered the images geometrically intentionally leaving out the final color calibration step to show the presence of eight projectors. Note the content is very testing for geometric registration since it has text and fine line illustrations. Please zoom in to see the registration quality.*

SSSP from V_1 to all the other nodes V_i, $i \neq 1$. R_i is given by concatenating all the R_{ij}s on the path from V_1 to V_i.

3.1.3. Recovering C_1

At this point, we have recovered the relative orientations of the different camera views with respect to the reference camera V_1, but we are yet to find the pose and orientation of the reference camera with respect to the 3D display, C_1. For this, we extend the method presented in [SM09] to find the pose and orientation of a single camera view with respect to the 3D display to handle multiple camera views. However, in this step we use only the images which do not have any projector turned on.

[SM09] first detects the four corners of the display in the single camera view. Using this and the known aspect ratio of the display it uses a non-linear *corner based optimization* to recover a rough estimate of the 3×4 camera calibration matrix by minimizing the reprojection error of the corners on the camera image plane. Using this rough estimate as the initial guess, it then runs a larger non-linear optimization using the constraint that the top and bottom boundary curves have the same shape in 3D. In this *curve based optimization*, the top and bottom curves are first detected in the image. The top 2D curve is first sampled in the camera image, back-projected in 3D via raycasting using the estimated camera calibration matrix, and intersected with the $Y = 1$ plane to sample the top curve in 3D. These 3D samples are then moved by $(0,-1,0)$ and reprojected back to the camera image plane. When the optimization converges these points should lie on the 2D bottom curve detected in the image plane. Hence, the optimization seeks to minimize the sum of the distances of these reprojected points from the detected 2D bottom curve in the camera image.

The key difference of our scenario from that of [SM09] is that we do not have a single camera view, but several of them. Hence, our corners and boundaries do not appear in a single camera image, but multiple ones. Note that there may be camera views where no boundary features appear. These views will not be considered in this step, but they are important nonetheless to register the projectors (Section 3.3). Now we will describe how we extend the method in [SM09] to handle multiple views. The key idea behind the extension is that both the back projection and the reprojection happens in the camera view in which the boundary feature or curve is detected. So, in the *corner based optimization step*, we detect the multiple images where the corners are detected. The identity of the corners (top left or bottom right and so on) can be easily found by studying the location of the foreground (the display surface) with respect to the background in the image. We reproject the 3D known corners to the respective camera views where they are detected using the matrix $K_c C_1 R_i$. Note that K_c and R_i are already known and we are now trying to estimate C_1. The distance between the reprojected point and the detected corner is the error which is summed across the multiple relevant views and minimized to provide a rough estimate of C_1. In the curve-based optimization stage, first all the images where the top boundary is detected are used. Let one such camera image be V_k. The top boundary is sampled in V_k (blue and purple points in Figure 2) and the corresponding 3D points are found by raycasting using the matrix $K_c C_1 R_k$ and intersecting these rays with the $Y = 1$ plane (cyan and gray points in Figure 2). Next these 3D points are moved by $(0,-1,0)$ to get an estimate of a set of points lying on the bottom boundary curve (orange and lime points in Figure 2). Now, unlike [SM09] which reprojected these points to the single camera view, we reproject these points to *all* the camera views where a bottom boundary curve is detected (red and green points in Figure 2). Each reprojected point will appear within the field-of-view (FOV) of a subset of these camera views. When this subset is not a singleton, it indicates that this part of the boundary was captured by multiple overlapping views. The average of the distance of the reprojected point from the detected 2D bottom boundary across all these views define our error for each point. We sum the errors across all the points sampled on the top boundary curve in V_k to provide a reprojection error E_k. We seek to minimize the sum of E_k across all the camera views where the top boundary curve is detected to find C_1.

3.2. Display Shape Extraction

Once the C_is for all camera views V_i, $i \in [1..N]$, have been recovered, we extract the display shape and a 2D parametrization thereof, by extending the method in [SM09]. For each view V_k containing the top (or bottom) boundary, we sample the detected 2D boundary in V_k, cast rays through these 2D samples using C_k and find their intersections with the $Y = 1$ (or $Y = 0$) plane to sample the 3D top (or bottom) curve.

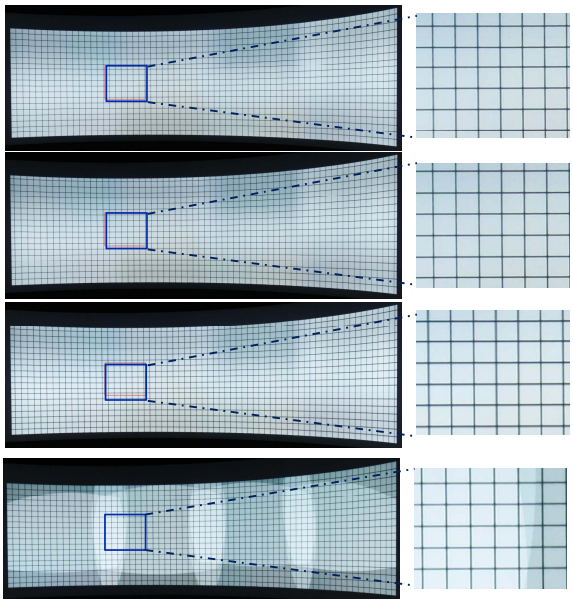

Figure 7: *Here we show that the quality of our multi-view registration (top) is comparable with the quality of registration achieved by a single camera view presented in [SM09] (second). We also show that we do not compromise the accuracy of our multi-view registration when using a low-resolution webcam (third), as opposed to a high-resolution SLR camera (top). Finally, we show the quality of our registration in the presence of severe non-linear distortion (bottom). In this image, we do not apply color calibration to show the distortions, evident from the curved projector boundaries. The zoomed in view on the right shows the quality of registration in four and two projector overlap area. Please zoom in to see the details.*

Then we fit a polynomial through the sampled 3D points (using least square linear optimization) to recover the top (or bottom) 3D curve. Technically, we do not need to restrict the degree of the polynomial. However, in the context of the application of display surfaces, we usually do not face very high degree polynomials. Note that due to several inaccuracies incurred during the earlier steps, these two curves may not be identical in shape. Hence, we average these two to converge to an identical shape for the top and bottom curves of the display in 3D. Next, we seek a 2D parametrization of the display with (s,t). The top and bottom curves on the XZ planes are arc length parameterized using the parameter s. Considering the 3D point (X,Y,Z) on the display surface, $X = X(s,t) = X(s)$ and $Z = Z(s,t) = Z(s)$. Since extrusion is along the Y direction, $Y = Y(s,t) = t$. Using the vertical extrusion assumption we can conclude that X and Z are independent of t and Y is independent of s.

3.3. Multiple Projector Registration

Geometric registration entails defining a function F for each projector that relates the projector coordinates (u,v) directly to display coordinates (s,t). This has to be done via the camera coordinates since this is the sensor that sees both the projector and the display. However, unlike a single camera calibration, there is no single camera coordinate system to define this intermediate coordinate space. Hence, to generate the projector display correspondences, we use the following method. Let the coordinate of a projector blob be (u_q,v_q). Let us assume that this blob is seen by P camera views, $P > 1$. Let the corresponding point in each of these P views be denoted by (x_i,y_i), $i \in [1..P]$. We cast a ray through each of the corresponding points using the respective camera calibration matrix and intersect the ray with the estimated 3D display. Then we find the (s,t) coordinates of this intersection point using the 2D parametrization developed in the previous section. Thus, using the P correspondences we find P display coordinate (s_i,t_i), $i \in [1 \ldots P]$, that correspond to the same blob (u_q,v_q). We take a weighted mean of all these P display coordinates to find an accurate corresponding point (s_q,t_q) for the blob (u_q,v_q). The weight is proportional to the minimum distance of the detected blob from the edges of captured view. Now, as in [SM09], we use such correspondences between the projector and the display coordinates to fit a rational Bezier patch using non-linear least square fitting solved by Levenberg-Marquardt gradient descent optimization technique. The rational Bezier allows us to achieve registration with a sparse set of blobs from each projector, even in the presence of projector distortion as is common in compact setups with short throw lens.

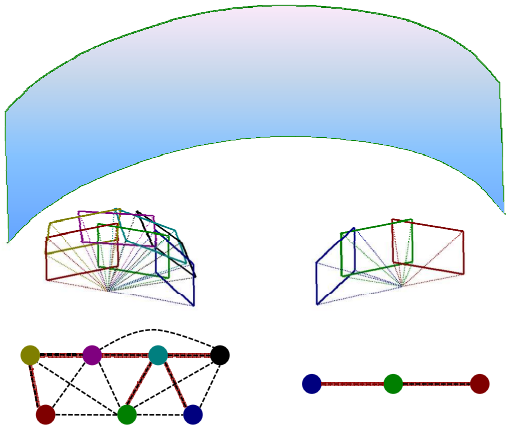

Figure 8: *Here we show the reconstructed camera views for the two multi-view registrations shown in Figure 6 - the seven views and the three views used for registering the left and right images in Figure 6 respectively. We also show the corresponding graphs and the spanning tree used to find the camera pose and orientations.*

4. Results

Figure 6 shows our results on our cylindrical display using eight projectors - 7 Epson 1825p LCD projectors (about $600 each) and 1 Canon SX80 LCD projectors (around $2000). Our display has a radius of about 14 feet and an angle of 90 degrees. We show the results using two camera configurations: (a) three views in a row; (b) and a more

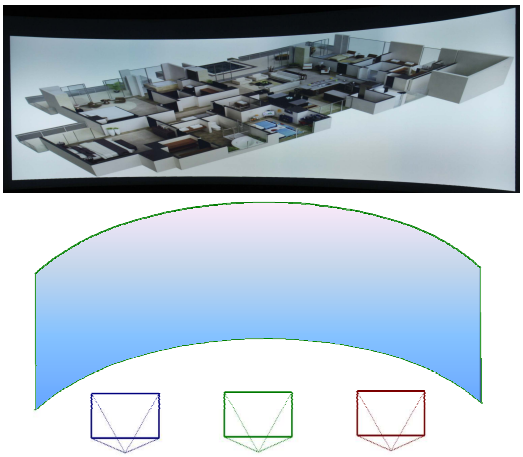

Figure 9: *This shows the applicability of our method when handling multiple cameras (instead of multiple views from same camera). The display on top is registered using three translated camera views with known focal length for the camera. The recovered camera pose and orientation along with the graph are shown in the bottom. Please zoom in to check the quality of the registration.*

complex set of seven views. Figure 8 shows the recovered camera views with respect to the display and the corresponding graphs with the spanning tree used to relate the camera views. The entire calibration of the cameras and projectors took about 5 minutes for these two cases. We find the accuracy of our method indistinguishable from that achieved by a single camera registration in [SM09] (Figure 7). However, using [SM09], we have to put the camera 20 feet away from the display while for our method, we can have the camera as close as 7 feet away. Hence, our method opens up the possibility of using smaller spaces to install such setups. Since we use a rational Bezier function for representing the relationship between the projector and the display coordinates, we need a sparse set of samples for accurate registration. In our demonstrations we have used 6×8 array of 48 blobs per projector, projected in six temporally sequential frames (see Figure 3). This allows us to use a low-resolution camera, even a VGA resolution webcam (640×480) to achieve a high quality registration (Figure 7). Finally, since the final warps are still defined by a rational Bezier, we can handle non-linear distortion in projectors(Figure 7) and use an identical real-time implementation in GPU as in [SM09] to achieve registration at interactive rates (See Video). For color calibration in all our demonstrations, we use the methods presented in [SLMG09].

4.1. Discussion

Accuracy and Scalability: We have used a simulator to analyze the accuracy of our reconstructed camera views and the display shape. Table 1 illustrates the accuracy of our method for a large number camera views and configurations. Large overlaps across adjacent camera views are desired for providing robust correspondences to our algorithm. Hence, we ran our simulation for different percentages of overlap for each of these camera configurations, which are then simulated on a variety of display shapes (by changing the shape of the boundary curves). We show the error in the recovered camera orientation (angular measurement in degrees) and error in the estimation of translation, focal length and the shape of the display curves (percentage deviation from the original values). Note that our method yields accurate results even when the number of camera views are large, even as large as 40. Further, errors are small even for overlaps as small as 5-10% across adjacent camera views. Note that we are discussing overlap across camera views and not projectors. Hence, from a user perspective when using an automated capture process, having large overlaps does not compromise anything – neither display resolution nor display real-estate. We have even used an unrealistically large number of 200 camera views to show the scalability of our method. Also, note that the accuracy is not compromised when the configuration of the camera views are changed.

Camera Non-Linearity and Resolution: We assume the camera to be devoid of any non-linear distortion. However, even if this is not true when using commodity cameras, our method does not result in any pixel misregistration. The camera non-linearity is accounted for by the fitted rational Bezier patches. However, the camera non-linearity affects the accuracy of the recovered 3D shape of the screen and hence, the final result may not be perfectly wall papered. Fortunately, the human visual system can tolerate such minor deviation from wall papering. In case of more severe camera non-linearities one can use standard camera calibration techniques to undistort the captured images [Zha99].

Camera Placement: As in [SM09], some camera positions may lead to ambiguities and hence insufficient number of constraints to recover the camera and display properties. These positions are when the normal to the camera image plane lies on the XZ plane (i.e. is perpendicular to the Y-axis) and should be avoided.

Handling Multiple Cameras: Our camera configuration is constrained by the fact that all the camera views share a common center of projection (COP). When considering camera configurations where the camera is translated from one view to another rather than rotated, or simply when multiple cameras are used, this constraint no longer holds. This results an ambiguity that does not allow us to recover the focal length and the multiple COPs at the same time. However, if the focal length of the camera(s) is known, our method can handle the case of multiple translated views or multiple cameras (Figure 9). We used focal length given by the EXIF tags. While our method (Section 3.1.1) computes the focal length with an error of less than 2.5%, we found empirically that an EXIF tag can have an error of around 8%. However, these errors still do not lead to pixel misregistration, but only some local deviation from wall-papering due to small errors in the display shape extraction.

Robustness: Though our method assumes a camera which is just rotated with no translation. This may not be true when

Table 1: *Percentage errors of the estimated camera and display parameters over a large number of simulations with different number and configuration of camera views and different extruded display shapes.*

Camera Config	% Overlap	Camera Orientation (deg)			Camera Position (%)			Focal Length (%)			Curve Shape (%)		
		Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
1	-	0.494	0.192	0.167	0.432	0.186	0.150	3.82	2.26	0.98	0.547	0.217	0.153
2×4	20	0.521	0.201	0.173	0.481	0.198	0.159	3.12	2.19	0.91	0.573	0.229	0.169
2×4	10	0.593	0.230	0.185	0.570	0.216	0.181	3.58	2.25	0.95	0.624	0.246	0.182
2×4	5	1.024	0.467	0.361	1.012	0.437	0.362	3.86	2.28	1.04	1.169	0.556	0.372
4×5	20	0.539	0.217	0.181	0.502	0.211	0.174	3.10	2.15	0.88	0.590	0.241	0.184
4×5	10	0.625	0.265	0.202	0.619	0.259	0.193	3.48	2.21	0.92	0.664	0.273	0.204
4×5	5	1.573	0.791	0.489	1.581	0.787	0.497	3.83	2.26	0.99	1.669	0.819	0.532
2×10	20	0.551	0.228	0.186	0.542	0.224	0.185	3.11	2.13	0.90	0.611	0.250	0.187
2×10	10	0.731	0.306	0.230	0.729	0.289	0.222	3.50	2.22	0.93	0.755	0.328	0.239
2×10	5	2.173	0.908	0.591	2.189	0.912	0.604	3.80	2.26	0.97	2.492	1.121	0.659
4×10	20	0.559	0.233	0.191	0.554	0.235	0.195	3.07	2.08	0.84	0.634	0.261	0.194
4×10	10	0.762	0.321	0.242	0.759	0.316	0.239	3.41	2.18	0.90	0.778	0.340	0.247
4×10	5	2.529	0.997	0.638	2.690	1.018	0.651	3.78	2.23	0.96	2.731	1.216	0.692
10×20	20	0.575	0.254	0.206	0.578	0.252	0.211	3.03	2.01	0.81	0.651	0.281	0.209
10×20	10	0.879	0.363	0.269	0.873	0.354	0.259	3.37	2.12	0.88	0.893	0.379	0.288
10×20	5	4.128	1.831	1.097	4.326	1.872	1.131	3.73	2.19	0.93	4.696	2.012	1.207

panning and tilting cameras using a PTU. However, we find that our rational Bezier provides a particularly robust framework to handle such small errors and also deviation from extruded surfaces. This is due to the fact that a small deviation from extrusion will lead to an erroneous 2D parametrization of the display surface, but the overlapping pixels from the multiple projectors will still map to the same (s,t). Hence, an imprecision in the extrusion can create small image distortions but will not lead to any misregistration.

Comparison with Prior Relevant Techniques: At a top level, our method is similar to any homography tree based method used for planar multi-projector display registration [CSWL02, RvBB*03] where the projector registrations are defined via transformations through the multiple cameras. This makes the registration sensitive to camera property estimation. Unlike these methods, we use the camera only to recover the display shape and the correspondences between the projectors following which a direct Bezier function between the projectors and display coordinates is used to register the projectors. This makes the registration less sensitive to the errors in the estimation of C_is. Further, the greater complexity of our method results from the fact that we need to recover the exact shape of the display with no planar display to take advantage of. This demands more complex algorithm design as described in Section 3. Finally, our graph based method to recover the relative camera orientations is faster and much simpler to implement than a standard global optimization method, as is in vogue for panoramic image generation techniques [BL07]. We used our method to recover the homographies in panoramic image generation and compared it with an existing global optimization techniques [BL07]. In addition to being at least an order of magnitude faster, the recovered homographies had less than 2% deviation from those recovered by the global optimization method.

Hardware and Computational Overhead: Our method works with commodity projectors and cameras, even inexpensive webcams (Figure 7). An inexpensive pan-tilt unit can be used to change the views manually. Or, the whole process can be completely automated by using a programmable pan-tilt unit. The time taken for the calibration procedure is dependent on the number of projectors and the camera views used, but is usually in the order of minutes. The calibration does not require any special hardware and can be achieved even on a regular laptop. The real-time image correction for each projector uses basic pixel/fragment shader functions which are available even on a low-end GPU.

5. Conclusion

In summary, we present a method that achieves automated geometric registration of multiple projectors on vertically extruded surfaces using multiple partial views of the display from a camera. This allows compact setup of displays by reducing the distance at which the camera needs to be placed. This also allows the display to be registered using a low-resolution inexpensive camera (e.g. a VGA webcam) rather than a high-end SLR camera. Our method is scalable, robust and accurate, resulting in a well registered wall-papered image that can be viewed by multiple users comfortably. It is amenable to real-time implementation to accommodate interactive applications.

Acknowledgements

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

References

[Ali08] ALIAGA D.: Digital inspection: An interactive stage for viewing surface details. *Proc. ACM Symp. on I3D* (2008). 2

[AX08] ALIAGA D., XU Y.: Photogeometric structured light: A self-calibrating and multi-viewpoint framework for accurate 3d modeling. *Proc. of IEEE CVPR* (2008). 2

[BJM07] BHASKER E., JUANG R., MAJUMDER A.: Registration techniques for using imperfect and partially calibrated devices in planar multi-projector displays. *IEEE TVCG 13*, 6 (2007), 1368–1375. 3

[BL07] BROWN M., LOWE D. G.: Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision 74*, 1 (2007), 59–73. 9

[BMY05] BROWN M., MAJUMDER A., YANG R.: Camera based calibration techniques for seamless multi-projector displays. *IEEE TVCG 11*, 2 (March-April 2005). 1, 2

[BSM06] BHASKER E., SINHA P., MAJUMDER A.: Asynchronous distributed calibration for scalable reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1101–1108. 3

[CNG*04] COTTING D., NAEF M., GROSS M., , FUCHS H.: Embedding imperceptible patterns into projected images for simultaneous acquisition and display. *International Symposium on Mixed and Augmented Reality* (2004), 100Ű–109. 1, 2

[CSWL02] CHEN H., SUKTHANKAR R., WALLACE G., LI K.: Scalable alignment of large-format multi-projector displays using camera homography trees. *Proc. of IEEE Vis* (2002). 3, 5, 9

[CZGF05] COTTING D., ZIEGLER R., GROSS M., FUCHS H.: Adaptive instant displays: Continuously calibrated projections using per-pixel light control. *Proc. of Eurographics* (2005), 705–714. 1, 2

[HCS*06] HARVILLE M., CULBERTSON B., SOBEL I., GELB D., FITZHUGH A., TANGUAY D.: Practical methods for geometric and photometric correction of tiled projector displays on curved surfaces. *IEEE PROCAMS* (2006). 1, 2

[JF07] JOHNSON T., FUCHS H.: Real-time projector tracking on complex geometry using ordinary imagery. *IEEE CVPR Workshop on Projector Camera Systems (PROCAMS)* (2007). 1, 2

[JWF*09] JOHNSON T., WELCH G., FUCHS H., FORCE E. L., TOWLES H.: A distributed cooperative framework for continuous multi-projector pose estimation. *IEEE Virtual Reality Conference* (2009), 35–42. 1, 2

[RBWR04] RASKAR R., BAAR J. V., WILLWACHER T., RAO S.: Quadric transfer function for immersive curved screen displays. *Eurographics* (2004). 1, 2

[RBY*99] RASKAR R., BROWN M., YANG R., CHEN W., TOWLES H., SEALES B., FUCHS H.: Multi projector displays using camera based registration. *Proc. of IEEE Vis* (1999). 2, 4

[RGM*03] RAIJ A., GILL G., MAJUMDER A., TOWLES H., FUCHS H.: Pixelflex 2: A comprehensive automatic casually aligned multi-projector display. *IEEE PROCAMS* (2003). 3

[RP04] RAIJ A., POLLEYFEYS M.: Auto-calibration of multi-projector display walls. *Proc. of ICPR* (2004). 3

[RvBB*03] RASKAR R., VAN BAAR J., BEARDSLEY P., WILLWACHER T., RAO S., FORLINES C.: ilamps: Geometrically aware and self-configuring projectors. *ACM TOG 22*, 3 (2003). 3, 5, 9

[SLMG09] SAJADI B., LAZAROV M., MAJUMDER A., GOPI M.: Color seamlessness in multi-projector displays using constrained gamut morphing. *IEEE Transactions of Visualization and Computer Graphics*, 6 (2009), 1317–1326. 8

[SM09] SAJADI B., MAJUMDER A.: Markerless view-independent registration of multiple distorted projectors on vertically extruded surfaces using single uncalibrated camera. *IEEE Transactions of Visualization and Computer Graphics*, 6 (2009), 1307–1316. 1, 2, 3, 6, 7, 8

[SSC*08] SUN W., SOBEL I., CULBERTSON B., GELB D., ROBINSON I.: Calibrating multi-projector cylindrically curved displays for "wallpaper" projection. *IEEE/ACM Workshop on PROCAMS* (2008). 1, 2

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings* (New York, NY, USA, 2006), ACM Press, pp. 835–846. 4

[YGH*01] YANG R., GOTZ D., HENSLEY J., TOWLES H., BROWN M. S.: Pixelflex: A reconfigurable multi-projector display system. *Proc. of IEEE Vis* (2001). 3, 4

[YW01] YANG R., WELCH G.: Automatic projector display surface estimation using every-day imagery. *9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (2001). 2

[YWB02] YANG R., WELCH G., BISHOP G.: Real-time consensus-based scene reconstruction using commodity graphics hardware. *Proceedings of Pacific Graphics* (2002). 1, 2

[Zha99] ZHANG Z.: Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision* (1999). 8

[ZLB06] ZOLLMANN S., LANGLOTZ T., BIMBER O.: 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). 1, 2

[ZWA*08] ZHOU J., WANG L., AKBARZADEH A., , YANG R.: Multi-projector display with continuous self-calibration. *IEEE/ACM Workshop on Projector-Camera Systems (PROCAMS)* (2008). 1, 2