A Flexible Projector-Camera System for Multi-Planar Displays

Mark Ashdown[1] mark@ashdown.name
Matthew Flagg[2] mflagg@cc.gatech.edu
Rahul Sukthankar[3,4] rahuls@cs.cmu.edu
James M. Rehg[2] rehg@cc.gatech.edu

[1] University of Cambridge Computer Lab; J. J. Thomson Avenue; Cambridge CB3 0FD; U.K.
[2] College of Computing; Georgia Institute of Technology; Atlanta, GA; U.S.A.
[3] Intel Research Pittsburgh; 416 S. Craig St. Suite 300; Pittsburgh, PA; U.S.A.
[4] The Robotics Institute; Carnegie Mellon University; Pittsburgh, PA; U.S.A.

Abstract

We present a novel multi-planar display system based on an uncalibrated projector-camera pair. Our system exploits the juxtaposition of planar surfaces in a room to create *ad-hoc* visualization and display capabilities. In an office setting, for example, a desk pushed against a wall provides two perpendicular surfaces that can simultaneously display elevation and plan views of an architectural model. In contrast to previous room-level projector-camera systems, our method is based on a flexible calibration procedure that requires a minimum amount of information for the geometry of the multi-planar surface scenario. A number of display configurations can be created on any available planar surfaces using a single commodity projector and camera. The key to our calibration approach is an efficient technique for simultaneously localizing multiple planes and a robust planar metric rectification method which can tolerate a restricted camera field-of-view and requires no special calibration objects. We demonstrate the robustness of our calibration method using real and synthetic images and present several applications of our display system.

1. Introduction

Recent advances in commodity high-resolution ultra-portable projectors have stimulated the development of a variety of novel projected displays such as large multi-projector walls [6], steerable projected displays [13], immersive environments [4, 15], intelligent presentation systems [14,17] and remote-collaboration tools [18]. The effort involved in manually calibrating multiple projectors to each other and aligning projected displays to physical surfaces has motivated research in projector-camera systems, where techniques adopted from multi-view geometry are applied to collections of projectors and cameras. A survey of computer vision research in this area can be found in [16].

We present a novel multi-planar display system based on an uncalibrated projector-camera pair. Our system exploits the juxtaposition of planar surfaces in a room to create *ad-hoc* visualization and display capabilities. In an office setting, for example, a desk pushed against a wall provides two perpendicular surfaces that can simultaneously display elevation and plan views of a map (see Figure 1). In contrast to previous room-level projector-camera systems, our method uses the minimum amount of information for the geometry of the multi-planar scenario. The key to our calibration approach is an efficient technique for simultaneously localizing multiple planes and a robust planar metric rectification method which can tolerate a restricted camera field-of-view. We demonstrate the robustness of our calibration method using real and synthetic images and present several applications of our display system.

The remainder of this paper is structured as follows. Section 2 discusses related work. Sections 3, 4, 5 and 6 describe and evaluate our calibration algorithms. Section 7 summarizes our contributions and proposes some promising directions for future research.

2. Related Work

Given a calibrated projector-camera pair, conventional structured light ranging can be used to reconstruct a complete 3-D model of an arbitrary display environment [15]. For an uncalibrated projector-camera pair, a variety of calibration techniques is available for the case where the display surface is a single plane. These methods generally consist of two stages: (1) estimation of homographies relating the projector and camera to the display surface, (2) followed by metric rectification of the planar homography. For example, in [17], planar homographies are estimated from projected points and metric rectification is achieved using the constraint that the display surface contains a rectangle of known aspect ratio whose boundaries are visible.

In [14], the projector and camera are rigidly attached in the manner of a stereo rig and homographies obtained from two different positions of the rig relative to a planar

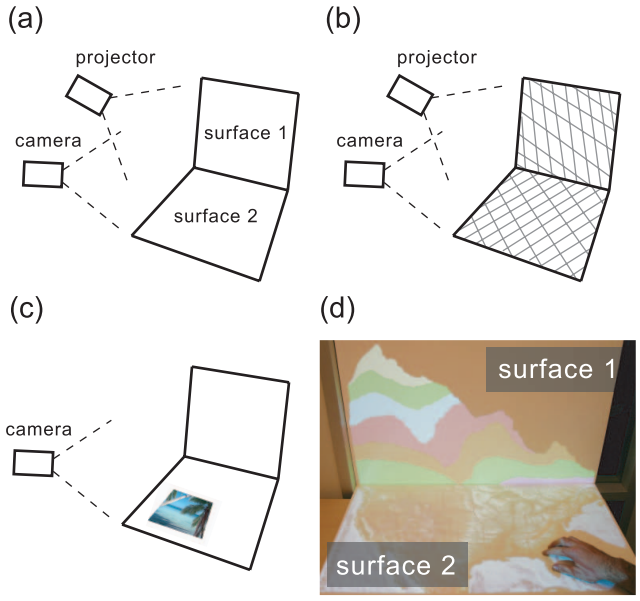

Figure 1: Calibrating a projector-camera system to a multiple surfaces. (a) The projector and camera are placed arbitrarily and the 3D geometry of the display surfaces is unknown. (b) A variant of structured light is used to find the surfaces and obtain homographies from projector to camera. (c) An everyday object is placed on the surfaces to get homographies from camera to surface. (d) The final mappings are used to create a display that spans the surfaces.

surface are combined to estimate the 3-D transformation between the projector and camera, and their intrinsic parameters. In the Everywhere Display system [13], a projector with a motorized mirror is calibrated using a camera and a known 3-D model of the display surfaces. In recent work, Okatani and Deguchi [12] address the problem of recovering the surface-to-camera homography in a system with intrinsically-calibrated projectors at unknown poses, a feature-less planar projection surface and stationary camera.

The primary difference between our work and the earlier research is that we address the autocalibration of *multiple* planar surfaces which can be viewed simultaneously by a single uncalibrated projector-camera pair. Our calibration approach builds on standard techniques for computing planar homographies [10, 17] and performing metric rectification [11]. In order to create a continuous display surface, it is necessary to map content consistently across the edges that are formed at the intersections of planes. As a result of these edge constraints, it is not possible to treat the calibration problem as a decoupled set of planar calibrations. Multi-planar scenes could in principle be addressed in the context of a more general 3-D autocalibration approach, for example based on recovering the fundamental matrix between the camera and projector followed by a Euclidean upgrade [9]. However the display system only requires homographies between planes to achieve the desired rendering effect. Moreover, accurate localization of specific geometric features, such as the edge that defines the intersection of two display surfaces, are critical for good performance. For these reasons, we have adopted an approach based on the simultaneous recovery of multiple planar homographies.

An extreme example of a multi-planar display system is an immersive environment, such as the CAVE [4] or the recent Blue-C system [7], in which every wall surface is a projected display. By contrast, our work focuses on office and home environments, where 2D displays have been found to be most effective [3]. Our goal is to develop a portable system that could be deployed quickly in a standard office environment. As a consequence, we have developed a highly automated calibration approach which computes only the information that is required to achieve the desired result. Another difference is that, unlike single-user 3D environments that require real-time head tracking, our display is view-independent and supports simultaneous interaction for multiple users.

Recent work on iLamps [5] includes a conformal mapping approach to projecting onto non-planar surfaces. The method can handle curved and piece-wise planar surfaces. However, since the approach is not based on explicitly identifying seams between surfaces, unwanted distortions can be introduced at inter-surface boundaries.

3. Multi-Planar Calibration

Our goal is to automatically detect, segment and calibrate a piece-wise planar scene into a set of connected surfaces that can form the basis for a multi-planar display. In the most general formulation of this problem we are given multiple projector-camera pairs in unknown locations and our goal is to form a large connected display with minimal input from the user.

Figure 1(a) illustrates the simplest multi-planar scenario, where two planes meeting at a common edge form a continuous display surface. One consequence of the need for a *continuous* display is that the planes cannot be calibrated independently. Coordinate frames must be aligned along an edge and the overall scale factors must be constrained so that content can be continuously mapped between displays without unwanted distortion. Figure 3 illustrates this goal in the context of a photo-browser application.

Figure 2 (top) shows three multi-planar surface configurations of increasing complexity: a simple two-plane geometry (*e.g.*, a desk surface placed against a wall); a common three-plane corner (*e.g.*, a table pushed into the corner of a room); and a five-plane, four-corner configuration (*e.g.*, the inside of an open box, or the floor, ceiling and three walls of a room). In each case, the goal of our system is to deter-

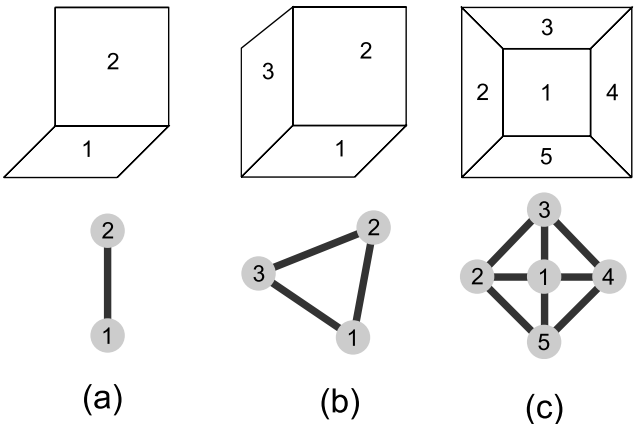

Figure 2: Top: Three multi-planar configurations: (a) desk surface placed against a wall, (b) table pushed into corner of a room, (c) inside of an open box. Bottom: Graphical representation of each configuration where projector-to-surface homographies are represented as nodes and constraints between homographies are edges.

mine the homography that maps the projector to each surface. Obviously, these homographies are not independent; points on the common edge between two planes should map to the same point in the projector, using either homography.

We can visualize the relationship between the various mappings in the form of a *homography graph*, where homographies are represented by nodes and constraints between homographies are denoted by arcs, as shown in Figure 2 (bottom). In projector-space, this graph partitions the frame buffer into polygons, where each polygon corresponds to a display surface and common edges between polygons map to the edges where the surfaces meet in the real world. By thinking of the projector as an imaging device rather than a projection device, one can imagine the configuration of the various surfaces in the projector's frame buffer.

Since a homography is completely determined by four point correspondences, one could specify each surface's homography by reading off the coordinates of the surface's corners from the image captured by the projector. From this, we can appreciate that the homography for the far surface in the five-planar display is completely constrained by the homographies of its four adjacent surfaces. Similarly, it is important to ensure that a closed loop on this homography graph returns a point to its original starting position. This paper shows that initial solutions for the different homographies can be obtained independently, but a refinement step is necessary to ensure that the mappings correctly obey the constraints expressed in the homography graph.

We assume that camera and projector optics can be modeled as projectivities [9], therefore we require a homography ${}^{i}_{\mathrm{P}}\mathrm{H}_{\mathrm{S}}$ from the projector to a Euclidian frame on each

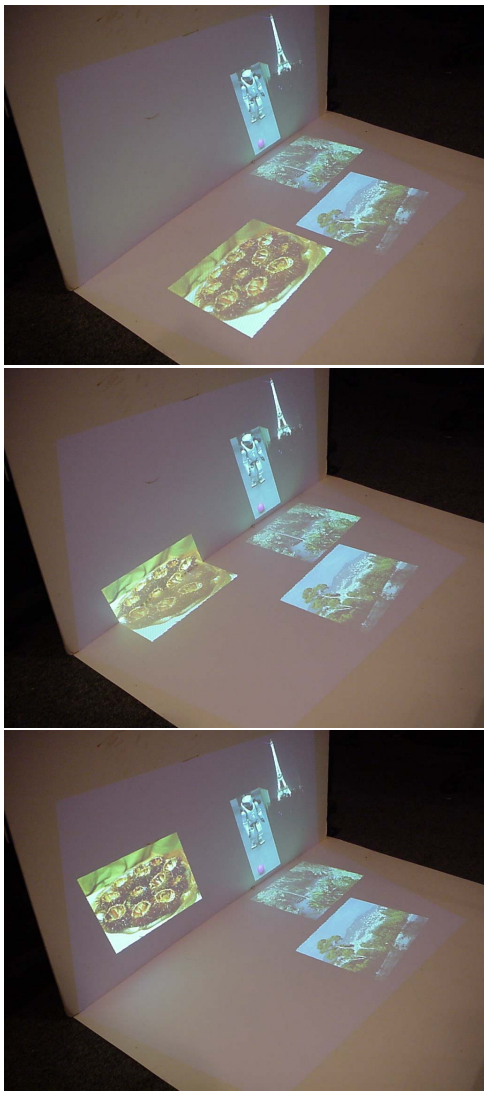

Figure 3: A photo is dragged from one plane to another in a photo-browser application that spans two surfaces.

surface i, where the frame is aligned to the physical layout. A homography from projector to surface cannot be observed directly. Thus, as in [17], we decompose the mapping for each surface into two homographies: one from projector to camera (through the surface), and another from camera to surface.

Calibration consists of three stages. The first stage (Section 4) is the identification of planar surfaces and recovery of homographies ${}^{i}_{\mathrm{P}}\mathrm{H}_{\mathrm{C}}$ from projector to camera through each surface i. This step uses uncalibrated structured light to segment planar surfaces and a robust algorithm for estimating homographies from line correspondences. The second stage (Section 5) performs metric rectification and alignment to obtain the homographies ${}^{i}_{\mathrm{C}}\mathrm{H}_{\mathrm{S}}$ from the camera

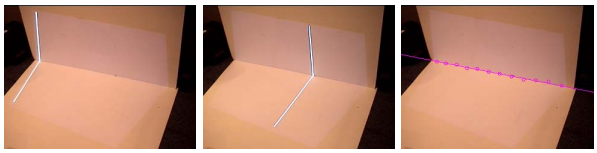

Figure 4: Uncalibrated structured light is used to identify boundaries between adjacent planar surfaces in the camera image.

to each surface i. We use a rectification method based on orthogonal line pairs. Our results highlight the importance of normalization for good error properties. The third stage (Section 6) adjusts the homographies ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}$ from projector to each surface i to reflect the contact constraints imposed by the intersection of pairs of surfaces. We describe an iterative refinement algorithm to enforce the contact constraint.

4. Planar Surface Identification

We describe a line-based method for segmenting planar surfaces and estimating homographies. The first step is to identify planar surfaces using an uncalibrated variant of structured light. The projector displays a series of horizontal and vertical lines that are observed by the camera. A line that crosses multiple surfaces appears as several line sections (Figure 4). The detection of kinks in the image of the projected lines can be exploited to segment the camera image into regions corresponding to distinct planar surfaces. Each kink divides the projected line into two sections. We fit a line to each section, intersect these lines and connect the intersections to determine the precise boundary (in the camera image) between each pair of adjacent surfaces. Using this boundary, we can process the sets of line sections on the two surfaces separately. These line sections are used to recover planar homographies from projector to camera through the respective surfaces.[1]

A standard method for estimating the homography between two planes is to apply the Direct Linear Transform (DLT) to a set of point correspondences [9]. For each pair of corresponding homogeneous points $\mathbf{x}_i$ and $\mathbf{x}'_i$, the homography $\mathtt{H}$ should satisfy $\mathtt{H}\mathbf{x}_i = \mathbf{x}'_i$.[2] Each point correspondence contributes two constraint equations. The set of constraints for $n \geq 4$ point correspondences can be combined into a single null-space problem of size $2n \times 9$ and solved using the singular value decomposition (SVD).

Since points and lines are duals in 2D projective geometry [9] it is straightfoward to derive a DLT formulation for line correspondences. A line satisfying the equation $ax + by + c = 0$ can be represented by the vector $\mathbf{l} = (a, b, c)^\top = k(\cos\theta, \sin\theta, -\rho)^\top$ for any $k \neq 0$, where $(\cos\theta, \sin\theta)$ define the normal vector to the line and ρ gives the signed distance from the origin. The line, $\mathbf{l}'$, resulting from a transformation through $\mathtt{H}$ is given by $\mathbf{l}' = \mathtt{H}^{-\top}\mathbf{l}$. Each line correspondence supplies three constraint equations: $\mathbf{l}'_i \times \mathtt{H}^{-\top}\mathbf{l}_i = \mathbf{0}$. Given n pairs of lines, a least squared estimate for $\mathtt{H}$ can be obtained by solving a null-space problem of size $3n \times 9$.

There are two advantages to using lines instead of points in our calibration framework. The first is that we have already established the value of line projections for segmenting surfaces. Given a set of line sections for a segmented surface, the homographies can be computed directly without performing any further projections. The second advantage is that it often easier to detect projected lines, since projected points must be kept small to avoid perspective distortion.

For each surface i, we calculate a homography ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{C}}$ from the projector (through surface i) to the camera using lines fitted to pixels in the camera image, and the known positions of those lines in the projector. A key issue in the line-based homography estimation step is achieving robustness to outliers. Outliers can arise from two sources. First, in conditions of high ambient lighting, detecting the projected lines in images can be a noisy process. Second, projected lines may fall across surfaces in the scene other than the planes of interest. The result in this case is short line segments in the periphery of the camera image which are inconsistent with the desired homography. A least squares estimation method which is robust to these outliers is required for accurate calibration.

Robustness has been addressed for point correspondences in [9, 10]. We have developed a robust algorithm for line correspondences that is based on an algorithm for calculating the fundamental matrix for epipolar geometry [19]. Our algorithm takes a set of n line correspondences $\mathcal{L} = \{(\mathbf{l}_j, \mathbf{l}'_j) : 1 \leq j \leq n\}$, and an upper bound ϵ on the fraction of outliers (*e.g.*, 10%), and returns a homography. It consists of two main steps. First, we employ DLT to compute homographies using several random minimal subsets of line correspondences and record the subset that achieves the lowest error for the remaining correspondences. Second, we classify each line correspondence as an inlier or an outlier depending on its error based on the recorded minimal subset. We then apply DLT to compute the final homography using only the inliers. This algorithm is detailed in a technical report [1]. We have tested this algorithm extensively, both in simulation and under controlled laboratory conditions. While traditional closed-form methods are unusable in our setup due to noise, the robust line-homography algorithm reliably recovers the projector-camera mappings for each surface.

[1] Note that for the line sections to be distinguishable, the camera and projector must be suitably displaced from one another: if they were at the same location, for instance, projected lines would always appear straight, irrespective of the shape of the surface.

[2] The 2D point (x, y) is represented by $\mathbf{x} = (xw, yw, w)^\top$, $w \neq 0$.

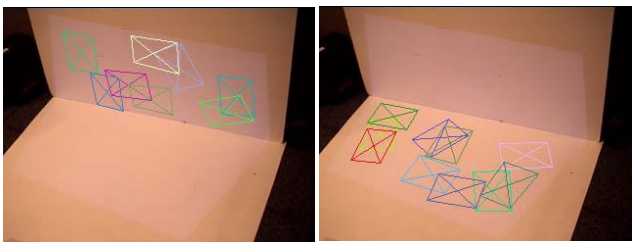

Figure 5: Orthogonal line pairs extracted from randomly-placed postcards are used to achieve metric rectification. This image shows the detected positions of a postcard as it was moved over the surface.

5. Metric Rectification

Once planar surfaces have been identified and their projector-camera homographies have been recovered, the next step is to recover the homographies ${}^{i}_{C}\mathtt{H}_{S}$ from the camera to each surface i. We decompose ${}^{i}_{C}\mathtt{H}_{S}$ into two parts: a metric-rectifying homography that maps the image of each surface into an arbitrary Euclidean frame, and a similarity transform that aligns this frame to the physical surface. To obtain the former, the camera observes a set of (arbitrarily-placed) right angles on each surface. These are obtained by imaging everyday objects such as postcards.[3]

Given a rectifying homography, the final step is to determine the similarity transform. We constrain the origin of each surface's coordinate frame to a point on a boundary, we constrain its x-axis to lie along that boundary, and select a scaling that is consistent across all surfaces. Thus, while the size of a projected object on the multi-surface display may be chosen arbitrarily, it should not change as it is moved between surfaces.

Liebowitz and Zisserman [11] describe a method for metric rectification of the image of a planar surface given five or more images of line pairs that are orthogonal on the surface. For instance, for our display system, the user could simply scatter a few rectangular objects (such as postcards) in the scene.

The conic dual to the circular points $\mathtt{C}^{*}_{\infty}$ on a Euclidian surface is

$$\mathtt{C}^{*}_{\infty} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

For perpendicular lines $\mathbf{l}$ and $\mathbf{m}$, $\mathbf{l}^{\top}\mathtt{C}^{*}_{\infty}\mathbf{m} = 0$. When these lines are imaged through a perspective transform $\mathtt{H}$, the conic dual to the circular points becomes $\mathtt{C}^{*'}_{\infty} = \mathtt{H}\mathtt{C}^{*}_{\infty}\mathtt{H}^{\top}$.

The goal is to recover $\mathtt{H}$ (up to a similarity transform), since ${}^{i}_{C}\mathtt{H}_{S} = \mathtt{H}^{-1}$. To do this, we first find $\mathtt{C}^{*'}_{\infty}$ for the given camera image. The images of the perpendicular lines (see Figure 5) satisfy $\mathbf{l}'^{\top}\mathtt{C}^{*'}_{\infty}\mathbf{m}' = 0$, so each such pair imposes a linear constraint on the elements of $\mathtt{C}^{*'}_{\infty}$. The constraints can be formed into a null-space problem of the form $\mathtt{A}\mathbf{c}=\mathbf{0}$, where each constraint gives one row of $\mathtt{A}$ and $\mathbf{c}$ is a six-parameter vector representation of $\mathtt{C}^{*'}_{\infty}$. The SVD provides a closed-form solution for $\mathtt{C}^{*'}_{\infty}$ given five or more constraints [9]. This closed-form formulation, which we will call CF, is analogous to the DLT algorithm, and also minimizes an algebraic rather than a geometric error. The value being minimized, $\sum(\mathbf{l}'^{\top}\mathtt{C}^{*'}_{\infty}\mathbf{m}')^2$, is related to the square of the cosine of the angle between the rectified lines.

[3] One could also use images of circular objects, such as CDs, to perform metric rectification. Although the basic techniques are similar, the image processing is more difficult since one must fit conics to perspectively-warped circles.

5.1. Recovery of Rectifying Homography

To obtain an estimate for the homography $\mathtt{H}$ we factor $\mathtt{C}^{*'}_{\infty}$ into the form $\mathtt{H}\mathtt{C}^{*}_{\infty}\mathtt{H}^{\top}$. Taking the SVD we get $\mathtt{U}\mathtt{D}\mathtt{V}^{\top}$, where $\mathtt{D}$ is a diagonal matrix. If the input data are well-conditioned, $\mathtt{D}_{33}$ should be small compared to $\mathtt{D}_{11}$ and $\mathtt{D}_{22}$. Thus, we assume $\mathtt{D}_{33}$=0 and express $\mathtt{D}$ in the form $\mathtt{D} = \mathtt{B}\mathtt{C}^{*}_{\infty}\mathtt{B}^{\top}$, where

$$\mathtt{B} = \mathtt{B}^{\top} = \begin{bmatrix} \pm\sqrt{\mathtt{D}_{11}} & 0 & 0 \\ 0 & \pm\sqrt{\mathtt{D}_{22}} & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

This gives us the desired decomposition for our estimate of $\mathtt{C}^{*'}_{\infty}$ as

$$\mathtt{U}(\mathtt{B}\mathtt{C}^{*}_{\infty}\mathtt{B}^{\top})\mathtt{V}^{\top} = (\mathtt{U}\mathtt{B})\mathtt{C}^{*}_{\infty}(\mathtt{U}\mathtt{B})^{\top} = \mathtt{H}\mathtt{C}^{*}_{\infty}\mathtt{H}^{\top} ,$$

where $\mathtt{H}=\mathtt{U}\mathtt{B}$. The choices for the signs of the diagonal elements in $\mathtt{B}$ create four possibilities for $\mathtt{H}$, but only two need to be considered: one where the signs are equal and one where they are different. Both versions of $\mathtt{H}$ will correctly rectify angles and length ratios in the camera image, but one will reverse orientation in our region of interest. We determine which is the correct one by combining each with a similarity to get the full homography from camera to surface as described in section 3, then testing those homographies with a point that is known to be on a particular side of the boundary line; only one of them will map it to the correct side.

5.2. Robust Estimation of Conic Dual

The SVD estimation of $\mathtt{C}^{*'}_{\infty}$ described above can suffer from normalization and robustness problems in practice. As in the case of the 8-point algorithm [8], we have found that centering and sphering the line data dramatically improves the conditioning of the problem and is key for correct metric rectification. Centering the line data consists of shifting the origin of the image plane to the homogeneous point $\mathbf{x}^{*}$ that minimizes the squared distances to the set of lines: $E(\mathbf{x}) = \sum_i(\mathbf{x}^{\top}\mathbf{l}_i)^2$. The unique solution is obtained by solving a

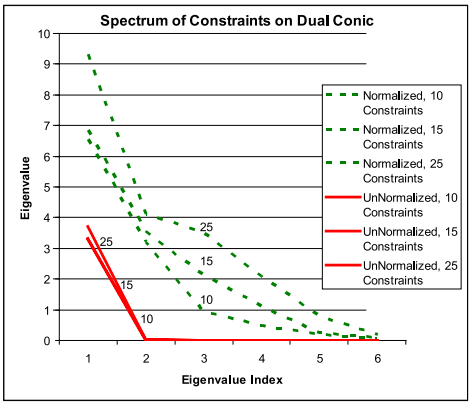

Figure 6: Experimental evidence for the importance of normalization in estimating the dual conic. This shows the eigenvalue distribution as a function of the number of constraints before and after centering and sphering line data.

symmetric 2×2 linear system. The RMS value of ρ after centering is given by $\hat{\rho} = \sqrt{E(\mathbf{x}^*)}$. The image coordinates are then scaled so that $\hat{\rho} = \sqrt{2}$. We find empirically that normalization significantly improves the spectrum of the A matrix used in the SVD.

We use two rules of thumb to assess the conditioning of our constraints: (a) inspection of spectrum of dual-conic constraint matrix for uniqueness of null-space; (b) the ratio of the maximum to minimum value for each constraint imposed by an orthogonal line pair. Figure 6 shows the distribution of eigenvalues before and after normalization. Normalization clearly separates the smallest eigenvalues from the rest. Without normalization, the null-space is far from unique. Also, as constraints are added, this separation becomes more evident. In addition to inspecting the spectrum, we noted the ratio of minimum to maximum values drops from 6 orders of magnitude before normalization to 1 order of magnitude following normalization for an experiment involving 27 detected line pairs from postcards.

As with the robust line homography algorithm (Section 4), our algorithm for robust metric rectification classifies each input line pair as an inlier or outlier, then computes final rectifying homographies from only the inliers. It takes a set of n line pairs $\mathcal{P} = \{(\mathbf{l}_j, \mathbf{m}_j) : 1 \leq j \leq n\}$ in the camera frame that are known to be orthogonal on the surface, and an upper bound ϵ on the fraction of outliers. It returns two homographies, one of which is the correct one. The correct homography is identified as described above. The proportion ϵ and the minimum number of line pairs s required to compute a rectifying homography (s=5 in this case) yield the number of random minimal subsets required as for our line homography algorithm. For P=0.9999 and ϵ=0.1, we require m=11 random minimal subsets. The robust algorithm proceeds by employing CF on these subsets

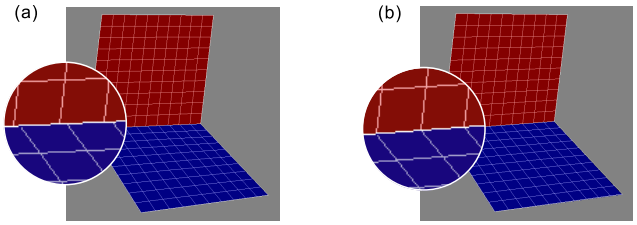

Figure 7: (a) Mappings can be inconsistent along inter-surface boundaries. (b) Our iterative refinement algorithm (Section 6.1) addresses this problem.

of line pairs and records the subset that achieves the lowest error for the remaining $n - s$ line pairs once they have been transformed using the computed rectifying homography. The error metric is

$$\text{median}_{\mathcal{P}} \ \ \text{d}(\mathtt{H}^{\top}\mathbf{l}_j, \mathtt{H}^{\top}\mathbf{m}_j) \ ,$$

where $\text{d}(.)$ is the cosine of the angle between two lines, and H is the homography derived from the current random minimal subset.

Using the homography from the recorded minimal subset, we classify each line pair in $\mathcal{P}$ as an inlier or outlier. As cosines, the error values for the pairs are bounded by the range [0,1], so we simply deem the $\lceil n\epsilon \rceil$ pairs with the highest error to be outliers. The final rectifying homography is then computed using CF with only the inliers.

To test our metric rectification algorithm, we generated homographies by randomly positioning a virtual camera above a virtual surface as for the line homography. Random pairs of perpendicular lines on the surface were transformed with the homography, then the resulting lines in (ρ, θ) were perturbed with zero-mean Gaussian noise, $N(0, \sigma^2)$. Gross outliers were added to the line pairs in various proportions. As outliers were added CF's error quickly became unacceptable, but that of our closed-form method remained level with up to 25% outliers.

6. Constraints Between Homographies

As discussed in Section 3, independently calculating the homographies ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}$ from projector to each surface i does not guarantee that they will be consistent along the boundaries between surfaces (Figure 7). We now present an iterative algorithm, motivated by the camera tree homography refinement technique in [2], that enforces the inter-surface constraints.

6.1. Algorithm

To simplify the following discussion, let us consider a two-surface setup. Let ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}} = {}^{1}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}^{-1}$ and ${}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}} = {}^{2}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}^{-1}$ be the homographies mapping each of the two surfaces to the projector. Ideally those two homographies will transform points

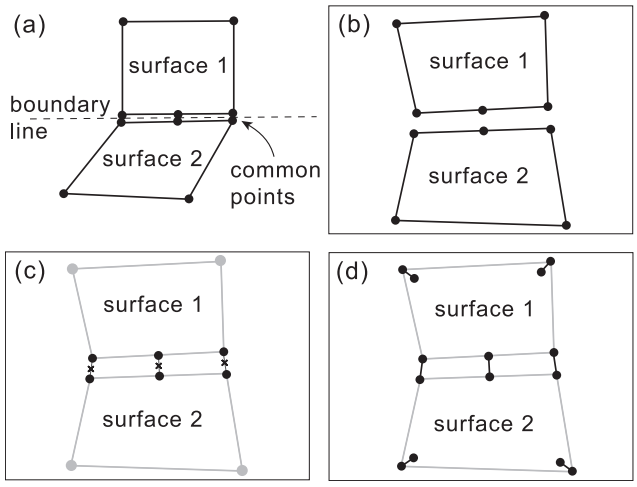

Figure 8: (a) Five points are defined on each surface. (b) Corresponding points are created in the projector. (c) Each boundary point pair is replaced by its midpoint. (d) The error is the sum of distances between transformed points.

on the boundary line identically; if $\mathbf{x}$ is a point on the boundary, we would like ${}^{1}_{S}\mathtt{H}_{P}\mathbf{x} = {}^{2}_{S}\mathtt{H}_{P}\mathbf{x}$. This will typically not be the case due to image processing error, so we apply this constraint by refining the homographies with an iterative algorithm (Figure 8).

We start by generating five points on each surface (Figure 8a), and generate initial correspondences in the projector using the current values of ${}^{1}_{S}\mathtt{H}_{P}$ and ${}^{2}_{S}\mathtt{H}_{P}$ (Figure 8b). The locations of the points on the two surfaces are fixed throughout the algorithm, but their corresponding points in the projector move as ${}^{1}_{S}\mathtt{H}_{P}$ and ${}^{2}_{S}\mathtt{H}_{P}$ are adjusted. At least three common points are needed on the shared boundary; this is to ensure that length ratios for the two homographies agree along the boundary.

The goal of the iterative algorithm is to refine ${}^{1}_{S}\mathtt{H}_{P}$ and ${}^{2}_{S}\mathtt{H}_{P}$ until each common point on the boundary $\mathbf{x}$ maps to the same point in the projector (*i.e.*, ${}^{1}_{S}\mathtt{H}_{P}\mathbf{x} = {}^{2}_{S}\mathtt{H}_{P}\mathbf{x}$) regardless of which homography is used. Figure 8c illustrates the iterative process. Each common point on the boundary transforms to two distinct points in the projector, and we create a new point in the projector by taking the midpoint of these. We set these new points as the correspondences for the boundary points on both surfaces. These new points, along with the (unchanged) correspondences of the non-boundary points are used to recompute ${}^{1}_{S}\mathtt{H}_{P}$ and ${}^{2}_{S}\mathtt{H}_{P}$ for the next iteration, using DLT. We terminate the refinement process when the improvement in the error falls below a threshold. The error (Figure 8d) is measured as the sum of the separations of the boundary points plus the sum of distances between desired and transformed locations of the non-boundary points (all measured in the projector frame).

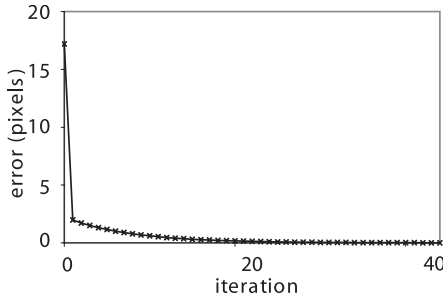

Figure 9: Using our homography refinement algorithm, the sum of errors between transformed points decreases rapidly with successive iterations. This ensures that homographies on adjacent surfaces become consistent along their common boundary.

6.2. Evaluation

Figure 9 shows a trace of error from a run of our iterative refinement algorithm. The combined error drops rapidly with the number of iterations; within 20 iterations, the inconsistency between the homographies along the shared boundary is imperceptible. If the initial estimates for the homographies were poor, this refinement technique could converge to a bad solution. In practice, we find that our robust algorithms generate initial estimates for ${}^{i}_{P}\mathtt{H}_{S}$ that are sufficiently accurate.

This algorithm works with any number of surfaces and inter-surface constraints. All of the homographies can be refined during each iteration of the algorithm, and each set of corresponding points at a constraint can be replaced by the centroid.

7. Conclusions and Future Work

We have presented robust calibration algorithms for aligning a camera-projector system to multiple planar surfaces. Our algorithm for calculating homographies from line correspondences works even when a significant fraction of the data consists of image processing outliers. This enables us to deploy the uncalibrated structured light system in difficult imaging situations. Our robust metric rectification algorithm employs everyday objects, such as postcards, to accurately determine the camera-surface homography for each of the multi-planar surfaces. Our iterative homography refinement technique converges in only a few iterations and significantly reduces inconsistencies along boundaries of multi-surface displays. None of the techniques presented in this paper require significant computation, and all can be deployed in practical camera-projector systems (requiring under a second of processing time).

We have implemented an interactive application for a multi-surface display that enables users to manipulate im-

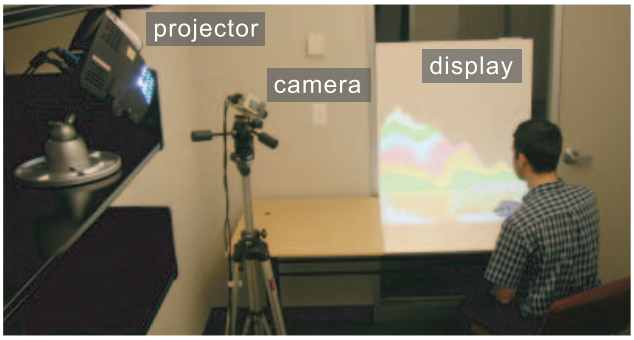

Figure 10: Using a visualization on a multi-surface display in an office setting. The projector and camera can be placed out of the way.

ages on each surface, and move them between surfaces (Figure 3). Multi-surface displays could also be used to simultaneously present different visualizations of a dataset. For example, a geologist could manipulate an overhead map on a horizontal surface while examining a vertical cross-section through the dataset on the other surface (Figures 1(d) and 10). Similarly, an architect or CAD tool user could simultaneously manipulate plan and elevation views on different surfaces. A three-surface display projected into a corner could be an affordable alternative to expensive immersive volumetric visualization displays, enabling users to examine 3D structure by clipping or projecting the data on to each surface. All of these displays can be created using a single commodity projector and camera, on any available surfaces in the user's environment.

In future work, we will investigate bundle adjustment to estimate projector-surface homographies and surface boundary points while enforcing edge constraints. Our system may also be extended to use multiple projectors in a straight-forward manner. When multi-planar surface-spanning projectors are positioned in overlapping configurations, recovered plane junctions and orthogonal line pairs may be shared for imposing constraints while simultaneously estimating projector-surface homographies.

Acknowledgments

We would like to thank Keith Packard and Matthew Mullin for their help and comments.

References

[1] M. Ashdown and R. Sukthankar. Robust calibration of camera-projector system for multi-planar displays. Technical Report HPL-2003-24, HP Labs, 2003. January.

[2] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable alignment of large-format multi-projector displays using camera homography trees. In *Proceedings of Visualization*, 2002.

[3] A. Cockburn and B. McKenzie. Evaluating the effectiveness of spatial memory in 2D and 3D physical virtual environments. In *Proceedings of CHI*, 2002.

[4] C. Cruz-Neira, D. Sandlin, and T. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *Proceedings of SIGGRAPH*, 1993.

[5] R. R. et al. iLamps: Geometrically aware and self-configuring projectors. In *Proceedings of SIGGRAPH*, 2003.

[6] T. Funkhouser and K. Li. Large format displays. *Computer Graphics and Applications*, 20(4), 2000. (guest editor introduction to special issue).

[7] M. Gross et al. Blue-C: A spatially immersive display and 3D video portal for telepresence. In *Proceedings of SIGGRAPH*, 2003.

[8] R. Hartley. In defense of the 8-point algorithm. *IEEE Transactions on PAMI*, 19(6), 1997.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] K. Kanatani, N. Ohta, and Y. Kanazawa. Optimal homography computation with a reliability measure. *IEICE Transactions on Information and Systems*, E83-D(7), 2000.

[11] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of Computer Vision and Pattern Recognition*, 1998.

[12] T. Okatani and K. Deguchi. Autocalibration of a projector-screen-camera system: Theory and algorithm for screen-to-camera homography estimation. In *Proceedings of International Conference on Computer Vision*, 2003.

[13] C. Pinhanez. The Everywhere display. In *Proceedings of Ubiquitous Computing*, 2001.

[14] R. Raskar and P. Beardsley. A self-correcting projector. In *Proceedings of Computer Vision and Pattern Recognition*, 2001.

[15] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of SIGGRAPH*, 1998.

[16] R. Sukthankar, T.-J. Cham, C. Pinhanez, and J. Rehg, editors. *IEEE International Workshop on Projector-Camera Systems (PROCAMS)*, 2003.

[17] R. Sukthankar, R. Stockton, and M. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In *Proceedings of International Conference on Computer Vision*, 2001.

[18] N. Takao, J. Shi, and S. Baker. Tele-graffiti. Technical Report CMU-RI-TR-02-10, Carnegie Mellon University, March 2002.

[19] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers, 1996.