Robust Calibration of Camera-Projector System for Multi-Planar Displays

Mark Ashdown[1], Rahul Sukthankar[2,3]
Cambridge Research Laboratory
HP Laboratories Cambridge
HPL-2003-24
January 30th , 2003*

E-mail: mark@ashdown.name, rahuls@cs.cum.edu

camera, projector, computer vision, homography

We present a robust calibration method for aligning a camera-projector system to multiple planar surfaces. Unlike prior work, we do not recover the 3D scene geometry, nor do we assume knowledge of projector or camera position.

We recover the mapping between the projector and each surface in three stages. In the first stage, we recover planar homographies between the projector and the camera through each surface using an uncalibrated variant of structured light. In the second stage, we express the homographies from the camera to each display surface as the composition of a metric rectification and a similarity transform. Our metric rectification algorithm uses several images of a rectangular object. In the third stage, we obtain the homographies between the projector and each surface by combining the results of the previous two stages. Inconsistencies appear along the boundaries between adjacent surfaces; we eliminate them through a process of iterative refinement.

Standard techniques for recovering homographies from line correspondences and performing metric rectification are very sensitive to image processing outliers. We present robust algorithms for both tasks, and confirm that accuracy is maintained in the presence of outliers, both in simulation and on our interactive application that spans a table and adjacent wall.

Our calibration method enables users to quickly set up multi-planar displays as they are needed, using any available projector and camera. These displays could be applied to visualization tasks in medical imaging, architecture and geographic information systems.

* Internal Accession Date Only Approved for External Publication
[1] University of Cambridge Computer Lab, J.J. Thomson Avenue, Cambridge CB3 0FD; UK
[2] HP Labs; One Cambridge Center, Cambridge, MA 02142
[3] The Robotics Institute; Carnegie Mellon University, Pittsburgh, PA 15213

Robust Calibration of Camera-Projector System for Multi-Planar Displays

Mark Ashdown[1]
mark@ashdown.name

Rahul Sukthankar[2,3]
rahuls@cs.cmu.edu

[1] University of Cambridge Computer Lab; J. J. Thomson Avenue; Cambridge CB3 0FD; U.K.
[2] HP Labs (CRL); One Cambridge Center; Cambridge, MA 02142; U.S.A.
[3] The Robotics Institute; Carnegie Mellon University; Pittsburgh, PA 15213; U.S.A.
http://www.cs.cmu.edu/~rahuls/Research/Projector/

Abstract

We present a robust calibration method for aligning a camera-projector system to multiple planar surfaces. Unlike prior work, we do not recover the 3D scene geometry, nor do we assume knowledge of projector or camera position.

We recover the mapping between the projector and each surface in three stages. In the first stage, we recover planar homographies between the projector and the camera through each surface using an uncalibrated variant of structured light. In the second stage, we express the homographies from the camera to each display surface as the composition of a metric rectification and a similarity transform. Our metric rectification algorithm uses several images of a rectangular object. In the third stage, we obtain the homographies between the projector and each surface by combining the results of the previous two stages. Inconsistencies appear along the boundaries between adjacent surfaces; we eliminate them through a process of iterative refinement.

Standard techniques for recovering homographies from line correspondences and performing metric rectification are very sensitive to image processing outliers. We present robust algorithms for both tasks, and confirm that accuracy is maintained in the presence of outliers, both in simulation and on our interactive application that spans a table and adjacent wall.

Our calibration method enables users to quickly set up multi-planar displays as they are needed, using any available projector and camera. These displays could be applied to visualization tasks in medical imaging, architecture and geographic information systems.

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 [2], steerable projected displays [6], immersive environments [1, 8], intelligent presentation systems [7, 9] and remote-collaboration tools [10]. The effort involved in manually calibrating multiple projectors to each other and aligning projected displays to physical surfaces has motivated research in camera-projector systems, where techniques adopted from multi-view computer vision are applied to collections of projectors and cameras.

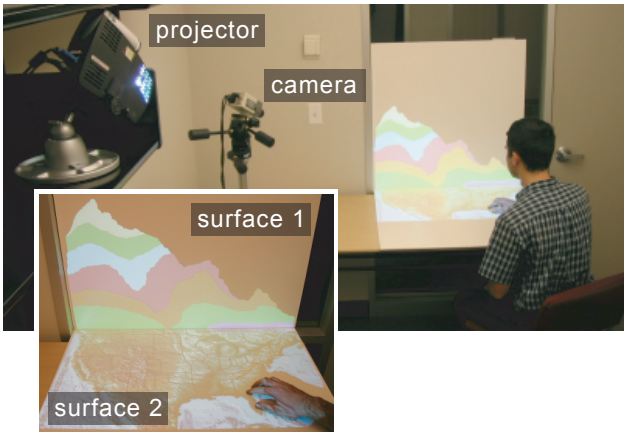

Figure 1: A two-surface camera-projector system running a topographic visualization. The camera and projector are mounted in unknown locations.

This paper describes a calibration method for creating novel projected displays where users can visualize data on multiple planar surfaces (Figure 1). Unlike prior work, we do not recover the 3D scene geometry, nor do we assume knowledge of projector or camera position. This enables users to quickly set up multi-surface displays as they are needed, using any available projector and camera.

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

The calibration of multi-surface displays using a camera-projector system is explored in the Office of the Future [8], where a complete 3D model of the display surfaces is extracted using structured light from a projector. That paper assumes that the intrinsic and extrinsic parameters of the camera and projector are known. We do not need to recover 3D scene geometry and the locations of our camera and projector are unknown.

Automatic calibration of a camera-projector system is presented in [9]. While that paper does not assume knowledge of projector and camera position, it assumes that the display surface consists of a single rectangle of known aspect ratio (whose boundaries are visible to the camera), and that the mapping between projector and surface can be described by a single planar homography. Our method applies to multiple surfaces, and we present more general algorithms for recovering the mappings.

The self-calibrating projector [7] automatically aligns its image to any (single) vertical planar surface using orientation information obtained from a tilt sensor. Although the global position of the components is not known, the system requires that the projector and camera be rigidly attached to form a calibrated stereo pair.

The use of homographies to describe transforms in multi-view geometry has become ubiquitous in computer vision (Hartley and Zisserman [3] present a good introduction to the subject). There has been much recent activity in the area of recovering 2D planar homographies, particularly from noisy point correspondences [4]. Treating the projector as a camera enables many results from the field to be applied directly to our problem. One of our algorithms for metric rectification is derived from unstratified rectification [5].

The idea of projecting displays onto everyday objects is becoming more popular [6, 8]. An extreme example of such a system is the CAVE [1], where the user is immersed inside a multi-surface projected display. A serious drawback of many multi-surface displays is that they are designed for a single user, whose head is tracked in real-time (since the projected scene and surface geometry do not match). While our calibration methods could be applied to those displays, we focus on displays where the 3D surface geometry agrees with the projected visualizations. These displays can be used by multiple users since all users agree on the orientations of the projected objects.

3. Multi-Planar Calibration

The goal is to produce a projected display on several flat surfaces, without requiring any knowledge of the 3D layout. A projector and a camera, mounted in unknown locations, are used to calibrate and create the display (Figure 2) with minimal input from the user.

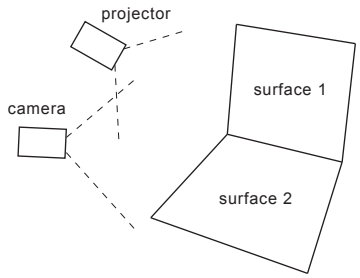

Figure 2: The 3D geometry of the display surfaces is unknown. The projector and camera are arbitrarily placed so that their fields of view contain the surfaces. The goal is to recover a mapping from the projector to each surface.

We assume that camera and projector optics can be modeled as projectivities [3], therefore we require a homography ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}$ from the projector to a Euclidian frame on each surface i. Since these cannot be observed directly, for each surface we decompose the mapping into two homographies: from projector to camera (through the surface), and from camera to surface. This generalizes an existing method for calibrating a projector to a single surface [9].

Calibration consists of three stages. The first stage (Section 4) employs an uncalibrated variant of structured light to identify the planar surfaces, and to recover mappings ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{C}}$ between projector and camera through each surface i. 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 3). We exploit this to segment the camera image into regions corresponding to the planar surfaces. We fit a line to each section, intersect these lines and connect the intersections[1] to determine the precise boundary (in the camera image) between each pair of adjacent surfaces. The same line sections are then used to recover planar homographies from projector to camera through the respective surfaces. 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.

In the second stage (Section 5), we recover homographies, ${}^{i}_{\mathrm{C}}\mathtt{H}_{\mathrm{S}}$ from the camera to each surface i. Structured light cannot be used for this stage since the relative positions of the projector and camera are unknown. We decompose the mapping into two parts: a metric-rectifying homography that maps the image of each surface into an arbitrary

[1] Simply connecting the raw intersections would give jagged boundaries; we model the boundaries as straight lines, fitted with orthogonal regression.

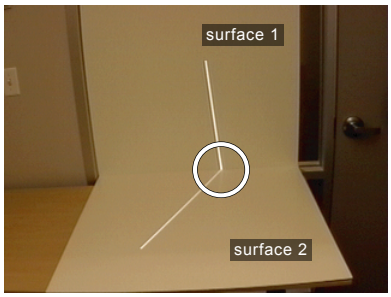

Figure 3: Uncalibrated structured light is used to identify boundaries between adjacent planar surfaces in the camera image.

Euclidean frame, and a similarity transform that aligns this frame to the surface. We explore two methods for metric rectification: (1) observing a set of line pairs known to be perpendicular on each surface [3, 5]; (2) observing a rectangle of known aspect ratio on each surface. To recover 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, the size of a projected object may be chosen arbitrarily, but should not change as it is moved between surfaces.

The final stage (Section 6) combines the homographies calculated in the two previous stages:

$$ {}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}} \quad = \quad {}^{i}_{\mathrm{C}}\mathtt{H}_{\mathrm{S}} \; {}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{C}} \; , $$

for each surface i. Since the homographies ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}$ for the different surfaces were calculated independently, there is no guarantee that they will be consistent along inter-surface boundaries. We apply an iterative refinement technique to ensure continuity across boundaries.

4. Homography from Line Correspondences

The standard method of calculating a homography between two planes is to obtain corresponding homogeneous points $\mathbf{x}_i$ and $\mathbf{x}'_i$. A homography $\mathtt{H}$ that maps one set to the other will satisfy[2]

$$ \mathtt{H}\mathbf{x}_i = \mathbf{x}'_i \; . $$

By seeking the $\mathtt{H}$ for which $\mathbf{x}'_i \times \mathtt{H}\mathbf{x}_i = \mathbf{0}$, a closed-form least-squares solution is reached. Since the three rows of this constraint are not linearly independent, the third row of each constraint can safely be discarded. Therefore, $n \geq 4$ point correspondences can be combined into a single null-space problem of size $2n \times 9$ and solved using singular value decomposition (SVD). This method of recovering homographies minimizes algebraic rather than geometric error, and is commonly known as the Direct Linear Transform (DLT).

Lines and points are duals in 2D projective geometry [3]. A line described by the equation $ax + bx + c = 0$ can be represented by the 3-element vector $\mathbf{l} = (a, b, c)^\top$, where we can express the line in a canonical form with the constraint $\|\mathbf{l}\| = 1$. The line, $\mathbf{l}'$, resulting from a transformation through $\mathtt{H}$ is given by $\mathbf{l}' = \mathtt{H}^{-\top}\mathbf{l}$. A variant of the DLT algorithm can be used to calculate homographies from line correspondences. This closed-form solution attempts to satisfy $\mathbf{l}'_i \times \mathtt{H}^{-\top}\mathbf{l}_i = \mathbf{0}$ in a least-squares sense. However, since any element of the line vector can be zero, all three rows of each constraint should be included (resulting in a null-space problem of size $3n \times 9$).

We present a new algorithm for calculation of homographies from line correspondences in Section 4.1, and evaluate it in Section 4.2.

4.1. Algorithm

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. The lines in the camera are subject to error, and occasionally the image processing techniques used to find them produce a totally incorrect result. Least-squares minimization is very sensitive to outliers, so we require an algorithm to calculate a homography from line correspondences that is robust to such outliers. Homography calculation from noisy point correspondences has been addressed [3, 4], but here we present an algorithm using line correspondences that is based on an algorithm for calculating the fundamental matrix for epipolar geometry [11].

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.

The size s of each minimal subset is 4 since this many line correspondences are required to compute a homography.[3] If we require a probability P that at least one of the

[2] In homogeneous coordinates, the 2D point (x, y) is represented by the vector $\mathbf{x} = (xw, yw, w)^\top$, $\forall w \neq 0$. Homogeneous vectors and homographies are therefore invariant to non-zero scaling factors. Throughout the paper this scaling is implicit, and $\mathbf{x} = \mathbf{y}$ should be interpreted as $\mathbf{x} = k\mathbf{y}$, for some $k \neq 0$.

[3] It is important to ensure that the subset is not degenerate (i.e., no three lines should be concurrent).

subsets does not contain any outliers, the number of subsets m that must be used is given by

$$m = \left\lceil \frac{\log(1-P)}{\log[1-(1-\epsilon)^s]} \right\rceil .$$

Typical values are P=0.9999 and ϵ=0.1, giving the number of samples m=9. The error for each homography based on s line correspondences is the median M of the errors between the desired and actual locations for the remaining $n-s$ correspondences not used in the homography calculation:

$$M = \text{median}_{\mathcal{L}} \ \mathrm{d}(\mathbf{l}'_j, \mathtt{H}\mathbf{l}_j) \ ,$$

where the error d measures the closeness of one line to another, and H is the homography derived from the current random minimal subset. We use $d(\mathbf{m}_1, \mathbf{m}_2) = \frac{\|\mathbf{m}_1 \times \mathbf{m}_2\|}{\|\mathbf{m}_1\| \, \|\mathbf{m}_2\|}$. The median error for the recorded minimal subset is used to calculate an estimate of the standard deviation $\hat{\sigma}$ of the errors [11]:

$$\hat{\sigma} = 1.4826 \left(1 + \frac{5}{n-p}\right) M,$$

where M is the value of the median error. The final homography ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{C}}$ is calculated from the inliers using DLT. We choose the inliers to be those line correspondences for which $\mathrm{d}(\mathbf{l}'_i, \mathtt{H}\mathbf{l}_i) < 2.5\,\hat{\sigma}$, where H is the homography calculated from the recorded subset.

4.2. Evaluation

To test the algorithm above we synthesized homographies by randomly positioning a virtual projector and camera above a virtual surface to simulate the hardware (Figure 4). The surface is defined to be a 1×1 unit square, and a random point X on the surface is selected to be the "look point" for the camera. The camera C is positioned at a fixed distance from X, with a random azimuth, elevation[4] and roll angle. The imaged lines are expressed in (ρ, θ) form[5] and perturbed with zero-mean Gaussian noise, $N(0, \sigma^2)$.

We explored several noise levels in the simulated experiments discussed below. For ρ, σ_ρ ranged up to 1% of the screen width, in increments of 0.2%. For θ, σ_θ ranged up to 0.01π in increments of 0.002π. Figure 5 shows the results (averaged over 10,000 runs) from two noise levels: level 1 (σ_ρ=0.2, σ_θ=0.002π) and level 4 (σ_ρ=0.8, σ_θ=0.008π).

The error in the homography calculated from the lines is measured by transforming ten random points and computing the mean geometric distance between their expected and actual locations (expressed as a proportion of the camera image width). Gross outliers were added to the line correspondences in incremental proportions from 0 to 5%, by replacing some lines with others drawn between two random points in the camera. The results show that while the closed-form solution becomes useless in the presence of outliers, our robust method successfully identifies the outliers and removes them. Additional experiments (not detailed here) show that the error curve remains flat for fractions of outliers up to 20%.

[4] To produce realistic images, elevation is greater than 20°.

[5] This is consistent with the Hough transform used in our physical setup.

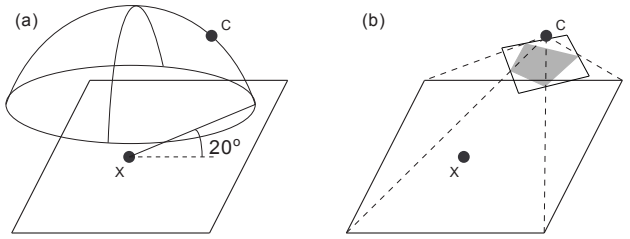

Figure 4: (a) In simulation the virtual camera is placed on on a hemisphere above a random point on the surface, with an elevation of at least 20 degrees. (b) The camera's field of view is created to include the whole surface.

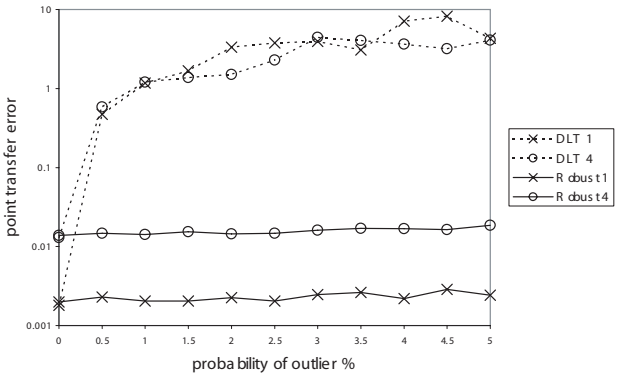

Figure 5: Comparison in the point transfer error between DLT and our robust line homography algorithm for two noise levels. The error is plotted on a log scale. DLT degrades even with a very small fraction of outliers. See text for details.

5. Metric Rectification

As discussed in Section 3, the mapping ${}^{i}_{\mathrm{C}}\mathtt{H}_{\mathrm{S}}$, from camera to surface i, can be decomposed into a metric-rectifying homography and a similarity transform. The method for determining the latter was discussed previously. We now present two algorithms for calculating the former.

5.1. Algorithm 1

Liebowitz and Zisserman [5] 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 (like 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 ${}^i_{\mathrm{C}}\mathtt{H}_{\mathrm{S}} = \mathtt{H}^{-1}$. To do this, we first find $\mathtt{C}_\infty^{*'}$ for the given camera image. The images of the perpendicular lines 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, and five or more such constraints enable a closed-form solution that gives us an estimate of $\mathtt{C}_\infty^{*'}$ [3]. This formulation is analogous to the DLT algorithm, and 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.

The next step is to factor our estimate of $\mathtt{C}_\infty^{*'}$ into the form $\mathtt{H}\mathtt{C}_\infty^* \mathtt{H}^\top$. We decompose our estimate using singular value decomposition into the form $\mathtt{U}\mathtt{D}\mathtt{V}^\top$, 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 $\mathtt{D}_{22}$, so 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 $\mathtt{B}$ might seem to 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.

This closed-form algorithm for metric rectification (termed CF) is attractive because it does not require any specialized calibration target: any set of perpendicular lines on the surface can be used. In practice, we obtain line pairs by taking a few images of a rectangular object, such as a postcard, on each surface. However, CF shares DLT's inherent susceptibility to gross outliers in the line pairs, motivating the development of a robust variant.

As with the robust line homography algorithm (Section 4.1), 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 $\mathcal{P} = \{(\mathbf{l}_j, \mathbf{m}_j) : 1 \le j \le n\}$ 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 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

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

where d gives the cosine of the angle between two lines, and $\mathtt{H}$ is the rectifying 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.

5.2. Evaluation

To test Algorithm 1 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, $N(0, \sigma^2)$. As in Section 4.2, we considered a number of noise levels. For ρ, σ_ρ ranged up to 0.1% of the screen width, in increments of 0.02%. For θ, σ_θ ranged up to 0.001π in increments of 0.0002π. Figure 6 shows results (averaged over 1000 runs) from two noise levels: level 1 (σ_ρ=$0.02, \sigma_\theta$=0.0002π) and level 4 (σ_ρ=$0.08, \sigma_\theta$=0.0008π).

Gross outliers were added to the line pairs in various proportions. To evaluate the algorithm, we generated a set of random orthogonal line pairs on the surface, and mapped them through the known imaging homography. These imaged line pairs were then transformed using the rectifying

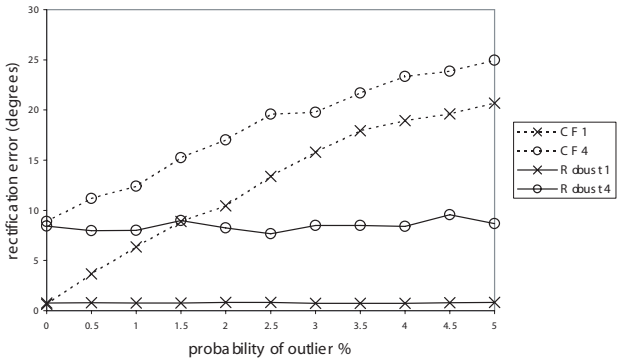

Figure 6: Comparison in the rectification angle error between CF and our robust metric rectification algorithm for two noise levels. CF degrades rapidly as outliers are added. See text for details.

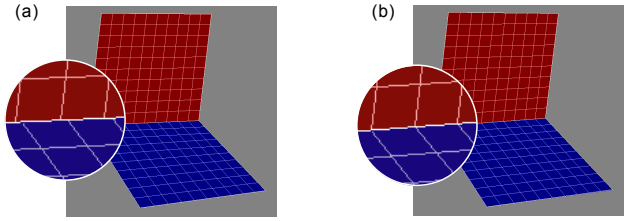

Figure 7: (a) Mappings can be inconsistent along inter-surface boundaries. (b) Our iterative refinement algorithm (Section 6.1) addresses this problem.

homography. The rectification error is the average deviation of rectified angles from the desired $90°$. Our results show that when no outliers are present our robust algorithm has an accuracy similar to that of CF, but as outliers are added CF's error quickly becomes unacceptable, while that of the robust method remains level. The robust method effectively identifies and removes the outliers. Additional experiments (not detailed here) show that the error curve remains flat for fractions of outliers up to 25%.

5.3. Algorithm 2

An alternate approach to metric rectification relies on imaging a calibration object, for example a rectangle with a known aspect ratio, a. Such a rectangle can be defined by the four points: $\{(0,0),(a,0),(a,1),(0,1)\}$. The corresponding points in the camera frame are given by the image coordinates of the corners. Using DLT, we can recover a rectifying mapping, which can be composed with a similarity transform to obtain ${}^{i}_{\mathrm{C}}\mathtt{H}_{\mathrm{S}}$. In practice, we can use an object with more feature points (e.g., a checkerboard) and replace the simple DLT with its more robust RANSAC variant [3]. The primary disadvantage of this algorithm is that it requires the user to supply an artificial target with a precise calibration pattern.

5.4. Discussion

We evaluated both Algorithm 1 and 2 in simulation and on our camera-projector display system. We found that the simpler algorithm performed better (at the expense of user inconvenience). Our robust Algorithm 1 successfully removes all of the outliers, but the inherent sensitivity in the method used to obtain $\mathtt{C}^{*'}_{\infty}$ necessitates very accurate image processing estimates of observed lines. The standard deviation of the line estimates in our current camera-projector setup is too high for Algorithm 1. We are exploring extensions to Algorithm 1 (such as better normalization) to make the technique more practical; our display system currently uses Algorithm 2.

6. Constraints Between Homographies

As discussed in Section 3, the homographies ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}$ from projector to each surface i are calculated independently, and thus there is no guarantee that they will be consistent along boundaries between surfaces (Figure 7). We now present an algorithm for iteratively refining these homographies.

6.1. Algorithm

To simplify the following discussion, let us consider a two-surface setup. Let ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}} = {}^{1}_{\mathrm{P}}\mathtt{H}^{-1}_{\mathrm{S}}$ and ${}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}} = {}^{2}_{\mathrm{P}}\mathtt{H}^{-1}_{\mathrm{S}}$ be the homographies mapping each of the two surfaces to the projector. Ideally those two homographies will transform points on the boundary line identically; if $\mathbf{x}$ is a point on the boundary, we would like ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}\mathbf{x} = {}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}\mathbf{x}$. 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 ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ and ${}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ (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 ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ and ${}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ 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 ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ and ${}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ until each common point on the boundary $\mathbf{x}$ maps to the same point in the projector (i.e., ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}\mathbf{x} = {}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}\mathbf{x}$) 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 correspon-

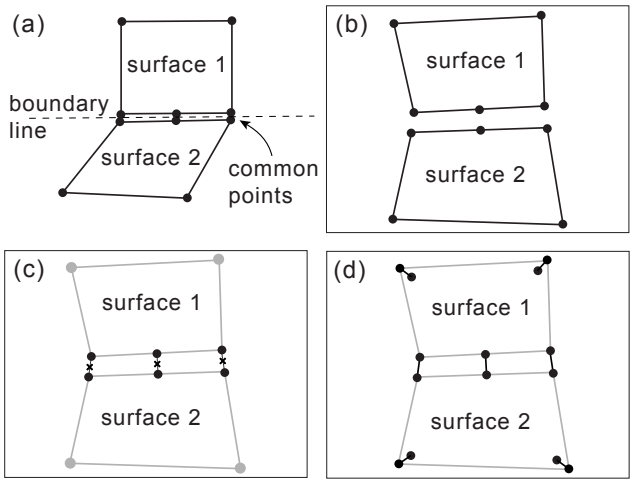

Figure 8: (a) Five points are defined on each surface. (b) Corresponding points are created in the projector. (c) Each boundary point pair is replaced by its midpoint. (d) The error is the sum of distances between transformed points.

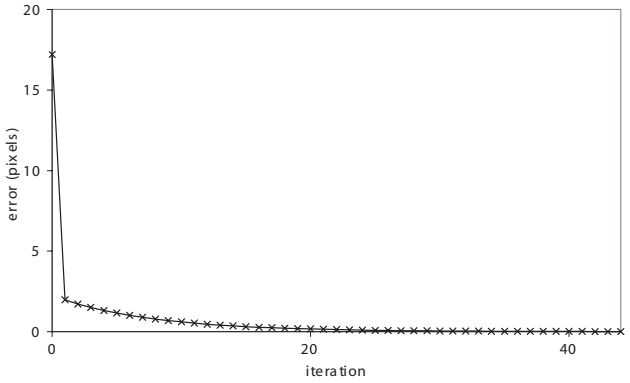

Figure 9: Using our homography refinement algorithm, the sum of errors between transformed points decreases rapidly with successive iterations. This ensures that homographies on adjacent surfaces become consistent along their common boundary.

dences for the boundary points on both surfaces. These new points, along with the (unchanged) correspondences of the non-boundary points are used to recompute ${}^{1}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ and ${}^{2}_{\mathrm{S}}\mathtt{H}_{\mathrm{P}}$ 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).

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 are sufficiently inaccurate, this refinement technique can converge to a bad solution. However, in practice, we find that our robust algorithms generate sufficiently good estimates for ${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{S}}$ that refinement improves the final calibration of our display system.

7. Conclusion

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 conditions where the DLT algorithm fails catastrophically (Figure 10a). For metric rectification, Algorithm 1 successfully removes outliers, but is limited by the inherent instability of the conic-dual method (Figure 10b). We are pursuing methods for improving its performance. We currently employ the simpler, but less natural, Algorithm 2 (Figure 10c). 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). Our aim is to calibrate multi-surface display systems without the use of specialized targets, simply by imaging everyday objects (such as sheets of paper) in the scene.

We have implemented an interactive application for a multi-surface display that enables users to manipulate images on each surface, and move them between surfaces (Figure 10c). 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 (Figure 1). 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.

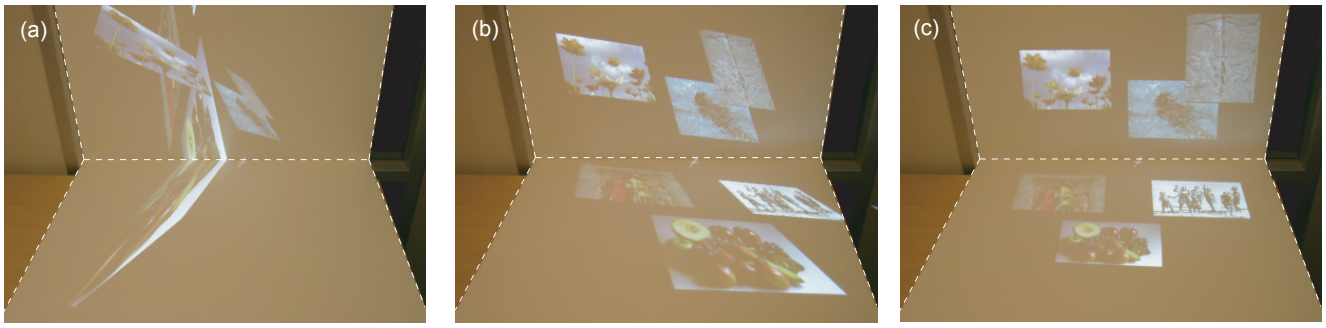

Figure 10: Multi-surface display aligned using three calibration techniques: (a) The standard calibration (${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{C}}$ using DLT, ${}^{i}_{\mathrm{C}}\mathtt{H}_{\mathrm{S}}$ using CF) is unusable due to image processing outliers – the projected images are not even close to being rectangular; (b) The robust algorithm (${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{C}}$ using robust algorithm, ${}^{i}_{\mathrm{C}}\mathtt{H}_{\mathrm{S}}$ using Algorithm 1) removes outliers, but the metric rectification from Algorithm 1 is insufficiently accurate; (c) Our current implementation (${}^{i}_{\mathrm{P}}\mathtt{H}_{\mathrm{C}}$ using robust algorithm, ${}^{i}_{\mathrm{C}}\mathtt{H}_{\mathrm{S}}$ using Algorithm 2) correctly calibrates the display to both surfaces – the images are now rectangular and aligned to the surfaces.

Acknowledgments

This project was made possible by the HP Labs summer internship program through which Mark Ashdown was able to work at CRL. We would like to thank Keith Packard and Matthew Mullin for their help and comments.

References

[1] 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.

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

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

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

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

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

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

[8] 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.

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

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

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