Affine Calibration from Dynamic Scenes

Russell A. Manning   Charles R. Dyer

Department of Computer Sciences
University of Wisconsin
Madison, Wisconsin 53706

Technical Report #1417
March 2000

Abstract

In [9] the authors introduced a linear algorithm for determining the affine calibration between two camera views of a dynamic scene. In this paper, we expand upon the algorithm and investigate its performance experimentally. The algorithm computes affine calibration directly from the fundamental matrices associated with various moving objects in the scene, as well as from the fundamental matrix for the static background if the cameras are at different locations. A minimum of two fundamental matrices are required, but any number of additional fundamental matrices can be incorporated into the linear system to improve computational stability. The technique is demonstrated on both real and synthetic data.

This work is sponsored by the National Science Foundation under Grant No. IIS-9530985 and by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-1-0138.

1 Introduction

Most research into camera calibration and scene reconstruction has focused on static scenes, or scenes without motion. Algorithms developed for static scenes can also be applied to dynamic scenes that contain rigid-body objects in motion by treating each rigid object individually. However, when a dynamic scene contains several moving objects, the movement of the objects relative to each other becomes a new source of information about the cameras and the scene. To utilize this extra information, new algorithms specifically designed for dynamic scenes must be developed.

In this paper, we explore a linear algorithm that utilizes the relative motion of objects in a dynamic scene to determine the *affine calibration* between two cameras viewing the scene. That is, the algorithm finds the homography induced by the plane at infinity between two views of the scene. Among other things, knowledge of affine calibration can be used for affine scene reconstruction and as an intermediate step in metric self-calibration. The basic algorithm was first introduced by the authors in [9] and is expanded herein to allow for more than two moving objects. Extensive experimental results are also presented.

Our algorithm finds affine calibration directly from the fundamental matrices associated with moving objects. At least two fundamental matrices are required, but additional ones can be incorporated naturally into the linear system, providing greater numerical stability. If the two cameras have different optical centers, then the stationary background elements of the scene give rise to the standard fundamental matrix, which can also be incorporated into the linear system.

Although two views of a moving rigid-body object will usually give rise to a fundamental matrix, the matrix can only be used by our algorithm if the object's motion meets certain conditions. The simplest form of these conditions is that the object must undergo a rigid translational motion. However, since only two views of the scene are actually used by our algorithm, this basic condition can be generalized. First, notice that the two views must be captured at different times for the dynamic nature of the scene to be relevant. Consequently, there is a missing interval of time between when the views are captured. The object can undergo *any* motion during this missing interval as long as the total change in the object and its location is equivalent to a single, rigid translational motion.

The term *object* has a specific meaning in this paper, defined by the general condition given above: An object is a group of particles in a scene for which there exists a fixed vector $\mathbf{u} \in \Re^3$ such that each particle's total motion during the missing time interval is equal to $\mathbf{u}$. Throughout the paper, objects will be assigned numbers and the notation $\mathbf{u}^i$ will represent the motion vector for object i.

The problem of finding the affine calibration between two views has been widely studied and is of great use in machine vision. For example, once the affine calibration has been recovered, affine scene reconstruction is immediately possible (e.g., by triangulation, or see [5]). Among other things, affine reconstruction can be used for affine model-based object recognition, tracking, augmented reality, feature transfer, and novel view generation in image-based rendering. Finding affine calibration is also an essential intermediate step in the stratified approach to metric self-calibration [1, 17, 4, 8, 13, 5]. For instance, if three views of a scene are available that have all been captured by the same camera with unvarying internal parameters and if the affine calibration can be recovered for each pair of views, then the metric calibration of the camera can be immediately determined [12, 11]. In the realm of pure image-based rendering, Manning and Dyer [10] have shown how affine calibration can be used to directly generate *linear* interpolation sequences of translational dynamic scenes without the need for scene reconstruction.

Various techniques for finding the affine calibration between pairs of views have been published. Several authors [16, 1] have used the fact that if two views are captured by a fixed camera undergoing a rigid translational motion, then the infinity homography between the views is known to be the identity matrix. Faugeras [5] describes a different approach to affine calibration that also involves pure translational motion. Some techniques [3, 2] have been developed for the restricted case of planar camera motion, that is, for when the camera's internal parameters do not change and the camera only undergoes translations and rotations that are parallel to a fixed plane. None of these techniques are directly related to dynamic scenes, and they all place restrictions on camera motion; our technique places restrictions on object motion but not camera motion.

The most direct method for finding affine calibration is to identify four conjugate directions (i.e., points on the plane at infinity) that are not all coplanar; like all planar homographies, the infinity homography is completely determined by its behavior on four points [5]. Pollefeys demonstrated that affine calibration between two views taken by the same camera can be determined from just two conjugate directions if the modulus constraint is utilized [12]. Since one conjugate direction can be determined from the motion of each moving object in the scene, these techniques might be applicable when two or more moving objects are present. However, the technique presented in this paper is usable even when only one moving object is present (because the static background can provide the second necessary fundamental matrix).

The technique presented by Zisserman et al. [17] and later expanded upon by Horaud et al. [8] applies, in general, to a different class of problems than our technique and uses a completely different mathematical approach. Zisserman's algorithm is for a stereo rig viewing a static scene from two different locations and is mathematically based upon projective reconstruction of conjugate points. In contrast, our technique works directly from fundamental matrices without any need for reconstruction; thus additional errors introduced during projective reconstruction (e.g., errors introduced through triangulation) are avoided. Furthermore, in our technique it is not strictly necessary to identify conjugate points at all if the fundamental matrices can be determined by some other means. For example, Stein [14] presents a direct method for finding the trilinear tensor between three views using optical flow; the required fundamental matrices could be determined from such a trilinear tensor. While our technique could be applied to the stereo rig problem for static scenes if the rig undergoes a rigid translation (see Section 5.2), it is not possible in general to apply Zisserman's technique to the dynamic scenes considered here.

Finally it should be mentioned that, although virtually no previous work has been done on utilizing dynamic scene information for calibration, Stein [15] has presented a method for finding the weak calibration between two widely-separated views by using statistics acquired from a dynamic scene over an extended period of time. His technique is unrelated to the present work and will not be discussed further.

2 Notation and preliminary concepts

Assume two camera views are captured at times $t = 0$ and $t = 1$ using pinhole cameras, which are denoted *camera* A and *camera* B, respectively. In this paper, a *fixed-camera formulation* is used, meaning the two cameras are treated as if they are at the same location and the world is moving around them; this is accomplished by subtracting the displacement $\mathbf{e}$ between the two cameras from the motion vectors $\mathbf{v}^i$ of all objects in the scene. In the reformulated scene, object i moves by $\mathbf{u}^i = \mathbf{v}^i - \mathbf{e}$ and what had been the stationary background becomes an object that moves by $-\mathbf{e}$. Under the fixed-camera formulation, the camera matrices are just 3×3 and each camera is equivalent to a basis for $\Re^3$. The basis induced by camera A will be called basis A, and so on. Note that, although we choose to reinterpret the cameras as sharing the same optical center, in actuality the cameras can be at different locations and can be completely different internally and externally.

The quantity $\mathbf{e}$ used above is called the *epipole*. A position or a direction in space, such as $\mathbf{e}$, exists independently of which basis is used to measure it; when necessary, we will use a subscript letter to denote a particular basis. For instance, $\mathbf{e}_A$ is $\mathbf{e}$ measured in basis A. If cameras A and B are at different locations in the original scene, then $\mathbf{e}$ is nonzero and there exists a fundamental matrix $\mathbf{F}$ for the cameras which has the following representation [6]:

$$\mathbf{F} = [\mathbf{e}_B]_\times \mathbf{H}^\infty_{AB} \tag{1}$$

where $[\cdot]_\times$ denotes the cross product matrix. When the two cameras share the same optical center, the fundamental matrix is $\mathbf{0}$ and has no meaning. However, for each moving object i in the scene, we can define a new kind of fundamental matrix. If, after switching to the fixed-camera formulation, object i is moving in direction $\mathbf{u}^i$, then the fundamental matrix *for the object* is:

$$\mathbf{F}^i = [\mathbf{u}^i_B]_\times \mathbf{H}^\infty_{AB} \tag{2}$$

The epipoles of $\mathbf{F}^i$ are the vanishing points of object i as viewed from the two cameras, and the epipolar lines trace out trajectories for points on object i.

Notice that, under the fixed-camera formulation, the stationary background in the original scene becomes just another moving object (provided $\mathbf{e}$ is nonzero). Hence by using the fixed-camera formulation, we are able to create a single mathematical theory that applies to pairs of cameras at different locations as well as to pairs of cameras that share the same optical center (e.g., two views from a single camera that is undergoing a zoom or rotating around its own optical center).

3 Motion-based affine calibration

We now show how affine calibration can be computed directly from the motion of two scene objects. Let the two objects be indexed by the set $\{0, 1\}$ and consider Eq. 2. Observe that $\mathbf{H}^\infty_{AB}$ is a rank three invertible matrix, but $[\mathbf{u}^i_B]_\times$ is rank two, and consequently $\mathbf{F}^i$ is also rank two. Because of the rank deficiency in $[\mathbf{u}^i_B]_\times$, the following arises: Let $S_i = \{\mathbf{M} \in \Re^{3\times 3} : \mathbf{F}^i = [\mathbf{u}^i_B]_\times \mathbf{M}\}$. Then S_i is a 4-dimensional vector space over the real numbers. A basis for S_i is given by the matrices $\mathbf{p}^i_0$, $\mathbf{p}^i_1$, $\mathbf{p}^i_2$, and $\mathbf{p}^i_3$, where

$$\mathbf{p}^i_0 = \mathbf{H}^\infty_{AB}, \quad \mathbf{p}^i_1 = [\mathbf{u}^i_B, 0, 0], \quad \mathbf{p}^i_2 = [0, \mathbf{u}^i_B, 0], \quad \mathbf{p}^i_3 = [0, 0, \mathbf{u}^i_B]$$

Because $\mathbf{H}^\infty_{AB}$ is in the basis of both S_0 and S_1, and because $\mathbf{u}^0$ and $\mathbf{u}^1$ are not parallel, $S_0 \cap S_1 =< \mathbf{H}^\infty_{AB} >$, where $< \cdot >$ denotes the subspace generated by a set of vectors. Since we only need to find $\mathbf{H}^\infty_{AB}$ up to a scalar, we only need to find one nonzero element in the intersection of S_0 and S_1. This is accomplished by first finding *any* two matrices $\mathbf{p}^i_4$ such that

$$\mathbf{F}^i = [\mathbf{u}^i_B]_\times \mathbf{p}^i_4. \tag{3}$$

Next, notice that S_i is spanned by $\mathbf{p}^i_1$, $\mathbf{p}^i_2$, $\mathbf{p}^i_3$, and $\mathbf{p}^i_4$ (because if $\mathbf{p}^i_4$ is in $< \mathbf{p}^i_1, \mathbf{p}^i_2, \mathbf{p}^i_3 >$, then $[\mathbf{u}^i_B]_\times \mathbf{p}^i_4 = 0$). Consequently, there exist scalars $k_1, \ldots, k_8$ such that

$$\mathbf{H}^\infty_{AB} = -k_1\mathbf{p}^0_1 - k_2\mathbf{p}^0_2 - k_3\mathbf{p}^0_3 - k_4\mathbf{p}^0_4 = k_5\mathbf{p}^1_1 + k_6\mathbf{p}^1_2 + k_7\mathbf{p}^1_3 + k_8\mathbf{p}^1_4 \tag{4}$$

The second equality means that

$$[\mathbf{p}^0_1\ \mathbf{p}^0_2\ \mathbf{p}^0_3\ \mathbf{p}^0_4\ \mathbf{p}^1_1\ \mathbf{p}^1_2\ \mathbf{p}^1_3\ \mathbf{p}^1_4][k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_7\ k_8]^\top = \mathbf{0} \tag{5}$$

Here we treat the matrices $\mathbf{p}^i_j$ as column vectors in $\Re^9$. The above can be solved using standard techniques from linear algebra (e.g., singular value decomposition to find the eigenvector of eigenvalue 0). Once the k_i's are found, we can find $\mathbf{H}^\infty_{AB}$ (up to a scalar) using Eq. 4.

Formally, we must show that the left-most matrix in Eq. 5 has rank 7. The rank is less than 8 since Eq. 4 has a solution. The vectors $\mathbf{p}^0_1$, $\mathbf{p}^0_2$, $\mathbf{p}^0_3$, $\mathbf{p}^1_1$, $\mathbf{p}^1_2$, $\mathbf{p}^1_3$ clearly form a linearly independent set because $\mathbf{u}^0$ and $\mathbf{u}^1$ are not parallel. If $\mathbf{p}^1_4 = h_1\mathbf{p}^0_1 + h_2\mathbf{p}^0_2 + h_3\mathbf{p}^0_3 + h_4\mathbf{p}^1_1 + h_5\mathbf{p}^1_2 + h_6\mathbf{p}^1_3$ for some scalars h_i, then by Eq. 3,

$\mathbf{F}^1 = [h_1\mathbf{u}^2, h_2\mathbf{u}^2, h_3\mathbf{u}^2]$ where $\mathbf{u}^2 = \mathbf{u}^1 \otimes \mathbf{u}^0$. This is a contradiction since $\mathbf{F}^1$ has rank 2, not rank 1. Thus 7 of the column vectors are linearly independent.

Because of the reliance on the linear independence of the column vectors in Eq. 5, it is crucial that $\mathbf{u}^0$ and $\mathbf{u}^1$ be linearly independent; the algorithm becomes unstable as the two objects move in more parallel directions.

Notice that the right-hand equality in Eq. 4 represents a new constraint similar in nature to the epipolar constraint, the trilinear constraint, or the modulus constraint. Such a constraint could be incorporated, for instance, into a nonlinear algorithm for finding the fundamental matrices of objects moving in non-parallel directions. It could also be used in conjunction with the modulus constraint for finding $\mathbf{H}^{\infty}_{AB}$.

3.1 Generalizing to multiple objects

If more than two moving objects are present in the scene, then the mathematics presented above can be generalized to incorporate each object's fundamental matrix simultaneously into one large, linear system.

Let the objects be numbered 0 to $n-1$. Let $\mathbf{P}(i)$ denote the matrix:

$$[\mathbf{p}^i_1 \; \mathbf{p}^i_2 \; \mathbf{p}^i_3 \; \mathbf{p}^i_4] \tag{6}$$

and let $\mathbf{0}_{9\times4}$ denote the 9×4 matrix filled with all 0's. We construct a matrix $\mathbf{M}$ by the following method:

Start with $\mathbf{M}$ *empty. For each* $i \in \{0, \ldots, n\text{-}2\}$ *and* $j \in \{i\text{+}1, \ldots, n\text{-}1\}$ *such that* $\mathbf{u}^i$ *and* $\mathbf{u}^j$ *are not parallel, enlarge the matrix* $\mathbf{M}$ *by appending the following matrix to its bottom:*

$$[\, \underbrace{\mathbf{0}_{9\times4}, \ldots, \mathbf{0}_{9\times4}}_{i-1}, \mathbf{P}(i), \underbrace{\mathbf{0}_{9\times4}, \ldots, \mathbf{0}_{9\times4}}_{j-i-1}, -\mathbf{P}(j), \underbrace{\mathbf{0}_{9\times4}, \ldots, \mathbf{0}_{9\times4}}_{n-j} \,] \tag{7}$$

Once $\mathbf{M}$ has been constructed, the following system is solved (e.g., by singular value decomposition):

$$\mathbf{M}\; [k_1 k_2 \ldots k_{4n}]^\top = \mathbf{0} \tag{8}$$

Affine calibration can now be determined from the following, which holds for every $i \in \{0, \ldots, n-1\}$:

$$\mathbf{H}^{\infty}_{AB} = k_{4i+1}\mathbf{p}^i_1 + k_{4i+2}\mathbf{p}^i_2 + k_{4i+3}\mathbf{p}^i_3 + k_{4i+4}\mathbf{p}^i_4 \tag{9}$$

4 Experiments with synthetic data

Extensive experiments with synthetic data were conducted to test the ideas of this paper. In this section, we summarize the experimental method and present the results.

4.1 Overview of the experimental procedure

The general pattern for each trial run was as follows: Two or more objects were generated, a camera was created that viewed the objects, each object was then moved by a random amount, and a second camera was created that viewed the objects in their new positions. If the second camera could not be created after a reasonable number of tries, the whole process was started over. When both cameras had been successfully generated, noise was added to the projected points on each image plane and then the equations presented earlier were used to recover the affine calibration between the cameras.

For different trials, the overall scale of each object was magnified or reduced, the distance the objects moved was scaled by different amounts, and the amount of noise was varied. The error in the recovered $\mathbf{H}^{\infty}_{AB}$ was measured using the following error metric:

Error Metric: Treating the matrices as vectors in $\Re^9$, with vectors $\mathbf{p}$ and $\mathbf{q}$ corresponding to the calculated $\mathbf{H}_{AB}^{\infty}$ and the true $\mathbf{H}_{AB}^{\infty}$, the error was calculated as:

$$1 - \frac{|\mathbf{p} \cdot \mathbf{q}|}{\|\mathbf{p}\|\|\mathbf{q}\|}$$

Note that this quantity is $1 - |\cos(\theta)|$, where θ is the angle between the vectors. Also note that when the matrices are equal the error is 0.

Each object consisted of up to 100 points selected randomly in a unit sphere such that the density of points was uniform throughout the sphere. The internal parameters of the cameras were randomly generated within ranges that are realistic for actual cameras: The principal point was chosen to be roughly within the middle third of the image, the skew between the x and y axes was in the range $[-10°, +10°]$, and the unit x and y distances were within 10% of each other. The retina of each camera was fixed at 640×480 pixels; this fact is crucial for interpreting the results that follow since measurements (e.g., noise added) will often be given in pixels.

Within the framework outlined above, three different scenarios were created to simulate different conditions under which the algorithms of this paper might be used in practice:

Scenario I: At time $t = 0$, the objects are near each other in space; by time $t = 1$ the objects have moved (each in an arbitrary direction) and are viewed by camera B, which is near camera A in space.

Scenario II: Objects are located arbitrarily within a circle and are only allowed to move parallel to the plane of the circle. Cameras are positioned randomly along the outside of the circle at a higher elevation than the objects.

Scenario III: Objects are located within a sphere of radius five units, and cameras are located in a larger, co-centered sphere but outside the sphere of the objects. Objects can move in any direction.

The positioning of cameras A and B close together in Scenario I simulates a hand-held camera, where the camera might not travel very far between views compared to how far the objects travel. Scenario II simulates a "parking lot" where vehicles drive along the flat surface of the lot and surveillance cameras are positioned on buildings around the lot. Scenario III tests the algorithm under fully general conditions. Note that, when only two objects are used (as was usually the case), each scenario corresponds to the motion of vehicles over level terrain because two vectors are always mutually parallel to some plane.

One final detail should be noted: Noise was added to the projected points on the retinas in such a way that no outliers were created. Specifically, if the trial run called for an average of ν pixels of noise, then uniform noise with a radius of 2ν pixels was added to each point; thus no conjugate point was more than 2ν pixels from its true position on the retina. It is assumed that, in practice, outliers would be removed by earlier steps in processing. Because of the lack of outliers, accurate fundamental matrices could be found by a normalized linear method [7].

4.2 Results

The following table shows how calibration error was related to the number of conjugate points and to the average amount of noise added to each conjugate point. Error values have been multiplied by 100.

CALIBRATION ERROR (10^{-2})					
	average noise added per point				
	5.003 pixels σ=0.261	2.500 pixels σ=0.130	1.250 pixels σ=0.065	0.500 pixels σ=0.026	0.250 pixels σ=0.013
100 points	error=8.383 σ=15.26	error=3.377 σ=8.388	error=2.067 σ=7.767	error=0.5363 σ=1.017	error=0.2766 σ=0.3996
60 points	10.29 σ=16.67	4.696 σ=10.88	2.002 σ=4.965	0.9931 σ=5.161	0.2765 σ=0.4131
30 points	14.16 σ=18.18	6.317 σ=11.50	2.950 σ=7.263	1.251 σ=3.838	0.7640 σ=2.254
10 points	38.08 σ=27.55	22.99 σ=23.65	11.54 σ=17.09	4.938 σ=10.06	3.127 σ=9.110

As would be expected, error decreases as the number of conjugate points increases and as the amount of noise decreases. The large standard deviations stem from occasional outliers; the scatter graphs in Figs. 1–4 give a visual indication of how the error values are distributed.

Recall that the algorithm becomes unstable as the objects move more parallel to each other in 3D when considered under the fixed-camera formulation. This instability is demonstrated in Fig. 1. Notice that there are few outliers for angles above approximately 20°. Thus for the remaining scatter graphs as well as for the table just presented, trials in which the angle between the object motion vectors was less than 20° were eliminated. For every trial in all the scatter graphs, 100 conjugate points were used per object and an average of 1.25 pixels of noise was added per point.

The scatter graph in Fig. 2 shows how error is reduced as noise is reduced. Notice that there are some outliers even at small noise levels, but the general trend is clear.

Fig. 3 demonstrates how error is reduced as the objects appear larger on the image plane. Notice that when the average object size covers less than about 40 pixels on the retina, error increases rapidly.

It might be hypothesized that the algorithm would be stabilized by greater retinal object motion. However,

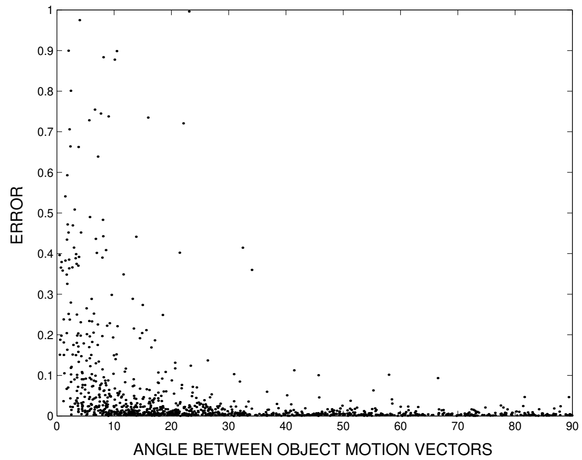

Figure 1: Calibration error vs. angle (in degrees) between object motion vectors considered under the fixed-camera formulation.

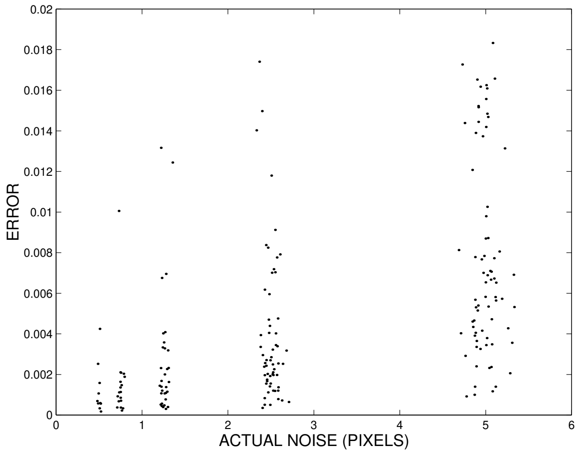

Figure 2: Calibration error vs. added noise.

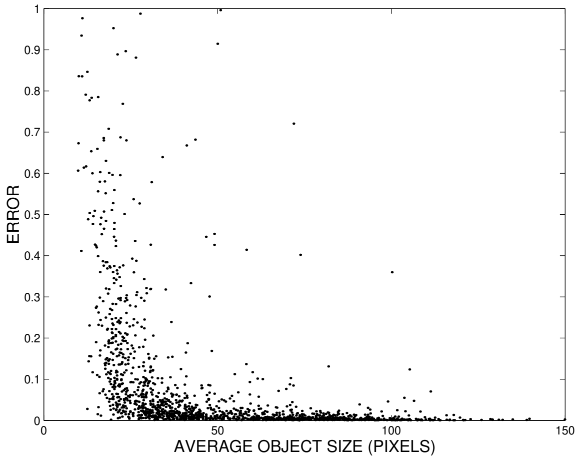

Figure 3: Calibration error vs. retinal object size.

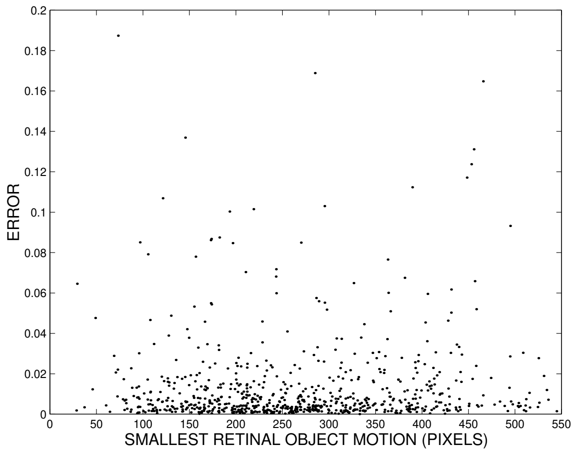

Figure 4: Calibration error vs. retinal object motion.

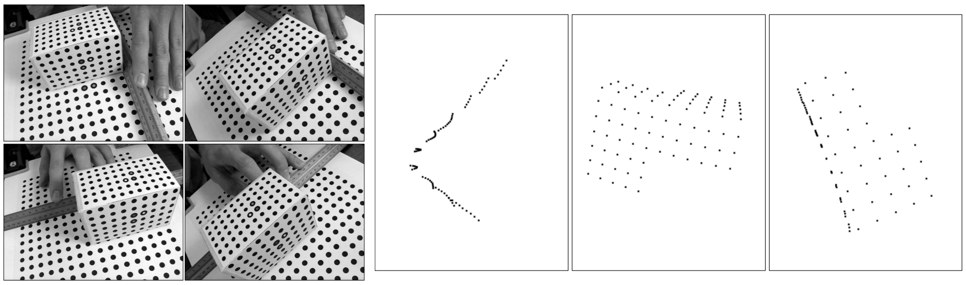

Figure 5: The four views on the left are the source views of the box that were used to find the two fundamental matrices for calibration. Views from camera A are on the left and views from camera B are on the right; the top pair shows object 0 moving towards the camera while the bottom pair shows object 1 moving laterally. The three rightmost views show the affine reconstruction of the box as seen from different angles, including an end-on view that shows a clear angle between the two surfaces of the box.

Fig. 4 shows that error was not affected by the amount of apparent motion of the objects across the image plane, at least for the ranges tested. It would be expected that, as the amount of retinal motion approached the noise level, the error would increase; this was not tested, however.

Finally, the table below shows how the algorithm was significantly stabilized by the use of more moving objects. Error values have again been scaled by 100. Also note the improvement gained by using 30 conjugate points rather than 10; this could be due to increased stability brought on by increased conjugate points in the algorithm we used for finding the fundamental matrices.

CALIBRATION ERROR (10^{-2})			
	2 objects	3 objects	4 objects
100 points	2.067	1.443	1.023
60 points	2.002	1.691	1.245
30 points	2.950	2.351	2.178
10 points	11.540	6.957	6.508

5 Experiments with real data

In this section, we present the results from two experiments performed with real scenes.

5.1 Experiment I

The first experiment was designed to produce very reliable data. The object that was used in the experiment was covered with a regular dot pattern (see Fig. 5), and the center of each dot was determined to subpixel accuracy by an automatic algorithm that found the center of mass of each dot. The cameras were fixed in position throughout the experiment.

Only one actual object was used, but it was moved in two different directions and thus served as two different objects. This means the two objects were not visible at the same time, but that fact is irrelevant to the algorithm when the cameras are in fixed positions relative to each other (e.g., as on a stereo rig). A situation like this might happen commonly in practice. For instance, consider a pair of fixed cameras monitoring the intersection of two roads. Occasionally, lone vehicles will cross the intersection, going in either direction. Each vehicle would give rise to a fundamental matrix, and over time the affine calibration could be accurately computed.

Figure 6: Views used in the second experiment. From left to right: the view from camera A at time 0, the view from camera B at time 0, and the view from camera A at time 1.

The ground truth affine calibration between the two views was acquired by using a three-dimensional calibration grid containing several hundred points at known positions. Each camera matrix was computed directly from the known 3D to 2D correspondences stemming from the calibration grid. Prior to this, radial distortion was corrected for as a separate step by minimizing the curved appearance of straight lines on the calibration grid.

The ground truth affine calibration, as determined directly from the full camera matrices, was

$$\mathbf{H}_{AB}^{\infty} = \begin{bmatrix} 0.005270 & -0.002681 & 0.3752 \\ 0.002858 & 0.004966 & -0.9269 \\ 0.0000009253 & -0.0000000624 & 0.005347 \end{bmatrix}$$

while the affine calibration determined using the motion of the box was

$$\mathbf{H}_{AB}^{\infty} = \begin{bmatrix} 0.005127 & -0.002625 & 0.3773 \\ 0.002789 & 0.004809 & -0.9260 \\ 0.0000009684 & -0.0000001226 & 0.005186 \end{bmatrix}$$

The distance between the matrices, using the same error metric used for the synthetic experiments, was 2.71×10^{-6}, or about $0.13°$. Note that only the two fundamental matrices arising from the motion of the box were utilized; a third fundamental matrix corresponding to the stationary background could have also been used.

As can be seen in the raw source images, the cameras had significant radial distortion. This was never completely corrected for, as is evident in the slight curvature of the lines in the reconstructed box object (Fig. 5). Nonetheless, even with some remaining distortion error, our technique produced an affine calibration very close to the "ground truth" calibration (which may have had some errors itself).

5.2 Experiment II

The second experiment utilized objects that had more natural texture so that fewer and less reliable point correspondences were obtained. In this experiment, several objects were placed on a piece of paper such that the paper could be slid across a table to simulate motion of the objects or the cameras. As before, the object was viewed by two cameras that were fixed in position throughout the experiment. The input images that were used for this experiment are shown in Fig. 6. Notice that the center view is zoomed in and has much less radial distortion than the left view. The left and center views, corresponding to camera A and camera B respectively, form one pair representing the object at time $t = 0$. From this pair, a fundamental matrix was recovered via standard techniques using about 30 point correspondences that were selected by hand. Next, the object was slid across the table in a manner approximating a pure translation. One final view was then captured from camera A only; this is shown as the rightmost view in Fig. 6. A second fundamental matrix was computed using the right and center views, again using about 30 point correspondences selected by hand.

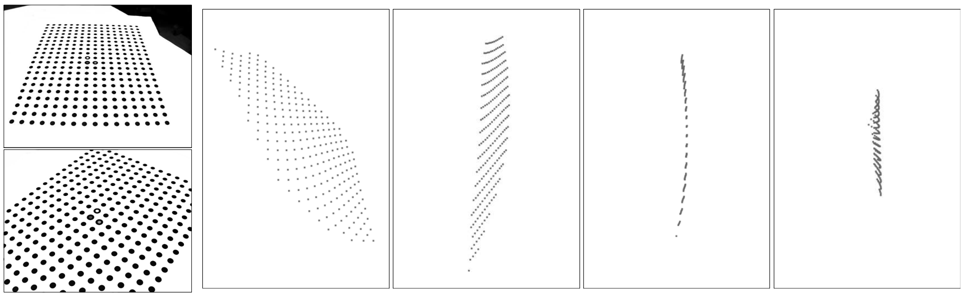

Figure 7: As an additional test of the affine calibration determined in the second experiment, affine reconstruction of a planar calibration grid was performed using two views of the grid *(left)*. Four views of the reconstructed surface are shown on the right.

Our algorithm was then applied to the two fundamental matrices, yielding an affine calibration of

$$\mathbf{H}_{AB}^{\infty} = \begin{bmatrix} 0.351 & 0.153 & 0.196 \\ -0.433 & 0.505 & 0.151 \\ -0.222 & -0.053 & 0.546 \end{bmatrix}$$

The ground truth affine calibration was determined from vanishing points. In particular, a regular grid was viewed by both cameras as it was placed in various orientations in space. The vanishing points of this grid, found automatically by a separate program, represent conjugate directions in the two views; four such points at infinity are sufficient for finding the affine calibration and many more than four were actually used. The affine calibration thus determined was

$$\mathbf{H}_{AB}^{\infty} = \begin{bmatrix} 0.359 & 0.139 & 0.208 \\ -0.431 & 0.497 & 0.128 \\ -0.211 & -0.069 & 0.557 \end{bmatrix}$$

Again, agreement is very good despite the many potential sources of error in this experiment. The distance between the matrices, using the same error metric, was 0.000756, or about $2.2°$.

As an additional test of accuracy, the affine calibration determined by our algorithm was used to reconstruct a regular, planar grid of points that was viewed by both cameras (see Fig. 7). The reconstruction shows some curvature in the grid lines, probably resulting in part from residual lens distortion errors since reconstruction using the "ground truth" affine calibration yielded similar curvature artifacts. Radial distortion was prominent in camera A and less so in camera B; this distortion was corrected for as a separate preprocessing step using the same method as in the first experiment. However, some distortion seems to have remained. Despite this, we still see agreement in the two affine calibrations even though they were determined by distinct methods.

6 Conclusion

Dynamic scenes contain sources of information that are not present in static scenes, but not many methods exist to utilize this extra information. This paper presented a linear algorithm for determining the affine calibration between two camera views of a dynamic scene. The algorithm has been shown to work on both synthetic and real data. Through experiments with synthetic data, it has been shown that the algorithm degrades gracefully with noise and the results improve as more moving objects are incorporated.

The equality in Eq. 9 represents a new constraint for the calculation of fundamental matrices for moving objects; this constraint could be combined with other constraints like the epipolar or trilinear constraint to improve fundamental matrix accuracy, or used in conjunction with the modulus constraint to determine better affine calibration.

It remains to be investigated how the ideas of this paper could be extended to utilize more than two views. The trilinear tensor thus available should stabilize the fundamental matrix calculation and improve results. Moreover, it may be possible to compute the affine calibration directly from pairs of trilinear tensors.

References

[1] M. Armstrong, Andrew Zisserman, and P.A. Beardsley. Euclidean structure from uncalibrated images. In *Proc. British Machine Vision Conference*, pages 509–518, 1994.

[2] M. Armstrong, Andrew Zisserman, and Richard Hartley. Self-calibration from image triplets. In *Proc. European Conference on Computer Vision*, LNCS 1064/5, pages 3–16. Springer-Verlag, 1996.

[3] P. Beardsley and Andrew Zisserman. Affine calibration of mobile vehicles. In R. Mohr and W. Chengke, editors, *Europe-China workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 214–221. Xidan University Press, Xi'an, China, 1995.

[4] F. Devernay and Olivier D. Faugeras. From projective to euclidean reconstruction. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 264–269, 1996.

[5] Olivier D. Faugeras. Stratification of 3-dimensional vision: Projective, affine, and metric representations. *Journal of the Optical Society of America*, 12(3):465–484, March 1995.

[6] Richard I. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Trans. Pattern Analysis and Machine Intell.*, 16(10):1036–1041, 1994.

[7] Richard I. Hartley. In defence of the 8-point algorithm. In *Proc. Fifth Int. Conf. on Computer Vision*, pages 1064–1070, 1995.

[8] R. Horaud and G. Csurka. Self-calibration and Euclidean reconstruction using motions of a stereo rig. In *Proc. Sixth Int. Conf. Computer Vision*, pages 96–103, 1998.

[9] Russell A. Manning and Charles R. Dyer. Dynamic view morphing. Technical Report 1387, Computer Sciences Department, University of Wisconsin-Madison, 1998.

[10] Russell A. Manning and Charles R. Dyer. Interpolating view and scene motion by dynamic view morphing. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 1, pages 388–394, 1999.

[11] M. Pollefeys. *Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 1999.

[12] M. Pollefeys and L. Van Gool. A stratified approach to metric self-calibration. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 407–412, 1997.

[13] M. Pollefeys and L. Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):707–724, August 1999.

[14] Gideon Stein. *Geometric and Photometric Constraints and Structure from Three Views*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1998.

[15] Gideon Stein. Tracking from multiple view points: Self-calibration of space and time. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages I:521–527, 1999.

[16] L. Van Gool, T. Moons, M. Proesmans, and M. Van Diest. Affine reconstruction from perspective image pairs obtained by a translating camera. In *Proc. Int. Conf. on Pattern Recognition*, pages A:290–294, 1994.

[17] Andrew Zisserman, P. Beardsley, and I. Reid. Metric calibration of a stereo rig. In *IEEE Workshop on Representation of Visual Scenes, Boston*, pages 93–100, 1995.