Object Tracking using Affine Structure for Point Correspondences

Gurmeet Singh Manku* Pankaj Jain† Amit Aggarwal Lalit Kumar
Subhashis Banerjee

Department of Computer Science and Engineering
Indian Institute of Technology
New Delhi 110016
email: *suban@cse.iitd.ernet.in*

Abstract

A new object tracking algorithm based on affine structure has been developed and it is shown that the performance is better than that of a Kalman filter based correlation tracker. The algorithm is fast, reliable, viewpoint invariant, and insensitive to occlusion and/or individual corner disappearance or reappearance. Detailed experimental analysis on a long real image sequence is also presented.

1 Introduction

Tracking and computation of structure and motion have been treated in the past as two separate problems. On one hand, traditional approaches to tracking have been based primarily on intensity correlation matching, with the more sophisticated approaches employing recursive Kalman filtering [6, 7] for temporal consistency. Little attempt has been made to impose structural constraints on the features being tracked [2, 8]. On the other hand most approaches to structure and motion parameter estimation have assumed that reliable correspondences are already available [4, 9]. The successes of these approaches are crucially dependent on the correctness of the assumed correspondences.

Reid and Murray [6, 7] have developed a real-time object tracker which uses a constant image velocity Kalman filter to establish the point correspondences across frames. They use the point correspondences obtained using their Kalman filter based correlation matcher to compute the 3D affine structure of the fixation point and the affine camera projection equations in each frame, and use these to localize the fixation point in each frame. The method is fast, requiring $O(n)$ operations to track n corners in each frame, and the fixation point can also be located with reasonable accuracy.

In this paper we further improve the Kalman filter based tracking algorithm. Our work attempts to unify feature tracking and structure computation and provide simultaneous solutions to both, each of which aids the other. We use the Kalman filter based method proposed by Reid and Murray to obtain the initial correspondences and subsequently impose the structure based constraints to obtain better correspondence results. Such an approach improves upon the performance of the Kalman filter based correlation tracker with minimal overheads and enhances the quality of correspondences making the structure and motion computations more accurate. Using the structure and motion we localize the fixation point in each frame more accurately, and the gaze demand generated is smooth and immune to occlusion or disappearance of the corners. We show that using our approach it is possible to obtain significantly smaller average per-pixel error in the localization of points on the image than what is obtained using the Reid-Murray algorithm. Such improved correspondence results may be useful not only for locating fixation points for tracking but also for any other application that uses structure, like object recognition, video indexing etc. Further, we compute the basis (affine camera projection parameters) in each frame from a large number of corners and the method does not depend on the accuracy of the detection of few particular corners. In each frame we compute the corresponding basis and consequently no change of basis is required. The computation required in each frame is directly proportional to the number of corners tracked ($O(n)$), and consequently the algorithm can be implemented in real-time.

We assume that the object in view is undergoing

*Present address: Dept. Computer Science, University of California, Berkeley, USA

†Present address: Dept. Computer Science, Stanford University, Stanford, USA

affine motion, which is more general than the customary assumption of rigid transformations [4]. We use the affine camera projective model [5, 4] and recover structure up to an arbitrary affine transformation. The affine camera model is valid when the field of view is small and the depth of the object is small compared to the viewing distance, which is expected in any foveal tracking application [5, 6]. When the perspective effects are small, it is convenient to assume the parallel projection model of the affine camera which explicitly models the ambiguities. Further, it is possible to detect when the affine camera approximation fails to hold.

In what follows we describe our new approach to feature tracking. In Sec. 2, we describe the Kalman filter based corner tracking algorithm [6] which we use to obtain the initial matches for our algorithm and discuss its limitations. In In Sec. 3, we describe our new algorithm for tracking using affine structure. In Sec. 4 we present tracking results on a long real image sequence and compare the performance with the Kalman filter based tracking algorithm of [6].

2 Kalman Filter based Tracking (Reid and Murray)

Kalman filters comprise a class of linear unbiased minimum-error covariance sequential state estimation algorithms. A *constant image-velocity* Kalman filter is used for each corner [6] to be tracked. The state vector for any corner, $\mathbf{x}(k)$ is given by a 4 x 1 matrix $[x(k), y(k), \dot{x}(k), \dot{y}(k)]^T$. State transition is modeled by $\mathbf{x}(k+1) = F\mathbf{x}(k) + \mathbf{v}(k) - \mathbf{u}(k)$, where $\mathbf{v}(k)$ are *iid* Gaussian zero-mean, temporally uncorrelated noise with covariance Q, $\mathbf{u}(k)$ is the camera's displacement (ego-motion) from time step $k-1$ to k, and

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Measurement is noisy and given by a 2 x 1 vector $\mathbf{z}(k) = H\mathbf{x}(k) + \mathbf{w}(k)$, where $\mathbf{w}(k)$ are iid Gaussian, zero-mean, temporally uncorrelated noises with covariance R, and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The covariance matrix R can be constructed by noting that the localization error in corner detection has the same variance as that of the Gaussian mask used by the corner detector.

The position of each corner is predicted in the new frame using the Kalman filter prediction equation [1] and all corners found within a search area whose size depends on the predicted covariance matrix P are cross-correlated with the corner in the previous frame. The best match, above a certain minimum threshold (typically 0.7) is selected and the Kalman filter's state vector and state covariance is updated using the standard equations [1]. For each corner unmatched in the previous frame (Kalman filter not initialized) the position in the next frame is predicted using the ego-motion of the camera. If a match is found successfully by cross-correlation then the Kalman filter for the corner is initialized.

The method yields several false matches across frames. This is because matching is based on the correlation of intensities in a window around the corner which is inherently unreliable. Moreover, the method tracks each corner individually, without regard to the interrelation of the features among themselves, *i.e.*, the structure. Invoking the assumptions of an affine camera model and affine motion of the object, it is possible to impose some constraints on the corners collectively. The motivation for imposing such constraints is to be able to reject false matches reported by the Kalman filter and also force some more matches, thereby improving the overall performance and reliability.

3 Matching using Affine Structure

We assume that the object in view is undergoing affine motion and that the images in the successive views are formed by affine projection. In the affine projection model [5] all projection rays are parallel and the epipoles are at infinity resulting in parallel epipolar lines. Given $n \geq 4$ point correspondences in two or more views, it is possible to determine the affine structure of the n point configuration [4] and the affine structure is invariant to affine motion. See [5] for an exposition of the affine multiple views geometry.

We derive the following constraint for matching based on affine multiple views geometry: *The point matches between the k^{th} and the $(k+1)^{th}$ views must be consistent with the affine structure computed up to the k^{th} view.*

In what follows we briefly describe the overall structure of our new algorithm and then describe some of the crucial procedures used in more detail.

3.1 Overall algorithm

Track_mode = **KALMAN**
for each successive image in the sequence do
{
Match using Kalman filters; (Sec. 2)

If **Track_mode** = **KALMAN**
(*Comment: Kalman filter based tracking*)
{

Gaze Point = center of mass of matched corners;
If **Initialization of basis** (Sec. 3.2) is successful then set **Track_mode** to **AFFINE**;
}
else (*Comment: Affine structure based tracking*)
{
If **basis computation** (Sec. 3.3) is unsuccessful then
{
Set **Track_mode** to **KALMAN**;
Gaze Point = center of mass of matched corners;
continue with the next frame;
}
Compute affine structure;(Sec. 3.6)
Reject outliers using affine structure; (Sec. 3.4)
If **Re-computation of basis** is unsuccessful then
{
Set **Track_mode** to **KALMAN**;
Gaze Point = center of mass of matched corners;
continue with the next frame;
}
Compute affine structure;
Force matches using affine structure; (Sec. 3.5)
Locate the **Gaze point** using its affine structure; (Sec. 3.7)
}
Update Kalman filters for all matched corners;
}

3.2 Initialization of basis and structure

We use the Tomasi and Kanade [9] procedure to initialize the basis and the affine structure of the corners which have a match history of at least $F \geq 2$ frames by constructing a $2F \times P$ measurement matrix $\mathbf{W}$ given below. We use $F = 6$ in our experiments.

$$\mathbf{W} = \begin{bmatrix} \Delta x_1^1 & \Delta x_2^1 & \dots & \Delta x_P^1 \\ \Delta y_1^1 & \Delta y_2^1 & \dots & \Delta y_P^1 \\ \Delta x_1^2 & \Delta x_2^2 & \dots & \Delta x_P^2 \\ \Delta y_1^2 & \Delta y_2^2 & \dots & \Delta y_P^2 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where $(\Delta x_i^j, \Delta y_i^j)$ are the centered coordinates of the i^{th} corner in the j^{th} frame. Carrying out a singular value decomposition of $\mathbf{W}$ yields its factorization into

$$\mathbf{W} = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & \mathbf{e_3} \\ \mathbf{e_1'} & \mathbf{e_2'} & \mathbf{e_3'} \\ \vdots & & \end{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_P \\ \beta_1 & \beta_2 & \dots & \beta_P \\ \gamma_1 & \gamma_2 & \dots & \gamma_P \end{bmatrix} \qquad (1)$$

The rows of the first matrix give the affine bases in the F views and the second matrix gives the invariant affine structure of the P points. The factorization of the observation matrix is the best rank 3 approximation possible in a least square sense. If the residual error is large suggesting a rank 4 situation it indicates that the affine projective model is invalid. A rank 2 situation suggests degeneracy - planar object or planar motion.

3.3 Basis Computation

Consider all corners in the previous frame which have been matched and for which affine structure is known. Compute the 4×2 basis matrix $\mathbf{h}$ using least squares methods such that

$$\begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & 1 \\ \alpha_2 & \beta_2 & \gamma_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{31} & h_{32} \\ h_{41} & h_{42} \end{bmatrix} = \begin{bmatrix} x_1^N & y_1^N \\ x_2^N & y_2^N \\ \vdots & \vdots \end{bmatrix}$$

The goodness of fit provides us with an estimate of how closely the observed matches follow the theoretical model. If time is at a premium then the re-computation of basis may be carried out only if the goodness of fit turns out to be really bad. The least squares procedure also returns the matrix $\sigma_{\mathbf{h}}$, which contains the variance of the corresponding elements of $\mathbf{h}$.

Note that with the above procedure we always obtain a corresponding basis in each frame with respect to an invariant set of 3D affine coordinates. Consequently no change of basis is ever required.

3.4 Rejection of Outliers using Affine structure

Out-lier rejection is done by considering all corners in the previous frame that have been matched and whose affine structure is known and predicting the position of corresponding corners in the current frame by projecting the affine structure. If the discrepancy between the predicted and the matched position is more than what is predicted by $\sigma_{\mathbf{h}}$ then we label the corner as unmatched. For a corner with affine structure (α, β, γ) the predicted position is:

$$x = \alpha h_{11} + \beta h_{21} + \gamma h_{31} + h_{41}$$
$$y = \alpha h_{12} + \beta h_{22} + \gamma h_{32} + h_{42}$$

3.5 Forcing of Matches using Affine structure

Consider all corners in the previous frame that have not been matched in the current frame but whose affine structure is known. We predict the position of those corners in the current frame by a projection mechanism similar to that used in rejection of outliers and search for the match in the window defined about the predicted positions using correlation matching. In addition, for all those corners whose affine structure is known from the past frames and which have not been matched in the previous frame, we force matches by projecting the affine coordinates in the current frame and searching for corners in the present frame in the neighborhood of the predicted position determined by $\sigma_{\mathbf{h}}$ and the variance of the affine structure. This procedure helps in recovering corners which fail to appear in one or two frames and retain the structure information.

3.6 Affine Structure Computation

Affine structure computation is similar to computation of basis except we have the **h** matrix and compute the affine structure. For all the matched corners the affine structure (α, β, γ) can be computed by least squares such that

$$\begin{bmatrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ h'_{11} & h'_{21} & h'_{31} \\ h'_{12} & h'_{22} & h'_{32} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x - h_{41} \\ y - h_{42} \\ x' - h'_{41} \\ y' - h'_{42} \end{bmatrix}$$

where the primed variables refer to the previous frame.

3.7 Gaze Point Computation

Determination of the gaze point by fixating on the centroid of all the matched points results in jitter in the gaze demand. This is because the centroid tends to shift due to disappearance and re-appearance of feature points. Instead, as suggested in [7], we determine the gaze point by determining the position of a fixed point in the new frame using the computed affine basis. Let the invariant affine coordinates of the gaze point be (α, β, γ) and let **h** be the basis matrix. The new gaze point is then given by

$$\begin{aligned} x_{gaze} &= \alpha h_{11} + \beta h_{21} + \gamma h_{31} + h_{41} \\ y_{gaze} &= \alpha h_{12} + \beta h_{22} + \gamma h_{32} + h_{42} \end{aligned}$$

In our experiments, we observed that the errors involved in the computation of basis are minimum in the fourth component of the basis *i.e.* its origin. We have therefore set all the components of the affine coordinates of the gaze point to zero which corresponds to the center of mass chosen in Tomasi Kanade factorization (Sec. 3.2).

3.8 Computational cost

Consider the problem of tracking n corners using the above algorithm. Kalman filtering (Sec. 2), computing affine structure (Sec. 3.6), rejecting outliers using affine structure (Sec. 3.4) and forcing matches using affine structure (Sec. 3.5) are all $O(n)$ operations and the least square computation of affine structure involves a fixed size basis matrix.

The only non-linear steps in the above algorithm are the initialization of structure and basis using the Tomasi-Kanade factorization (Sec. 3.2) and the basis computation (Sec. 3.3). These require least square solutions with matrices whose sizes are proportional to n. However, these processes can also be made of constant time by using only a fixed number of corners. We have found in our experiments that choosing a fixed number (30) of corners with maximum match age for these computations does not affect the performance of the overall algorithm. Consequently, the overall tracking algorithm can be made to work in linear time.

Further, note that in our algorithm it is not necessary to handle the degenerate (2D) and the non-degenerate (3D) cases separately in all the procedures mentioned above.

4 Results

We have implemented the corner tracking algorithm in real-time (at 25 *Hz*) using a SUN machine for frame grabbing and corner detection [3] and a Pentium (120 *MHz*) for tracking and robot control. We have carried out extensive experimentation on some image sequences. In what follows we present the tracking results in a 150 frames off-line sequence obtained at 7.5 *Hz*. The sequence consists of facial motion including nod, shake and curl. The sequence contains instances of both degenerate and non-degenerate motion. In Fig. 1 we present the tracking results in six frames. Some statistics contrasting the performance of the Kalman filter based algorithm against our method have been plotted in Fig. 1 to 3.

The affine structure based algorithm finally matches roughly the same number of corners in each frame as the Kalman filter algorithm (Fig. 2). But in each frame about 8-10% of the corners matched by the Kalman filter are rejected by the various stages of out-lier rejection and a similar number of new matches are found by forcing matches using affine structure. This increases the quality of matching as is evident from the comparison of the average per-pixel error in structure based prediction using both methods (Fig. 3). Consequently, the gaze point can be located more accurately. In particular, we have observed that the quality of gaze fixation is significantly better than the Kalman filter based algorithm when there is an abrupt change in the motion and the Kalman filters fail to predict accurately.

Also, the structure based method of gaze fixation, as suggested in [7], generates a smoother gaze demand as compared to center of mass tracking (Fig. 4) and results in a smaller camera speed (Fig. 5).

In Fig. 6 we plot the 3D bases across the entire sequence to demonstrate that our method computes corresponding bases across the frames and no change of basis is ever required. The corresponding average variance of the invariant affine coordinates of the point set is given in Fig. 7.

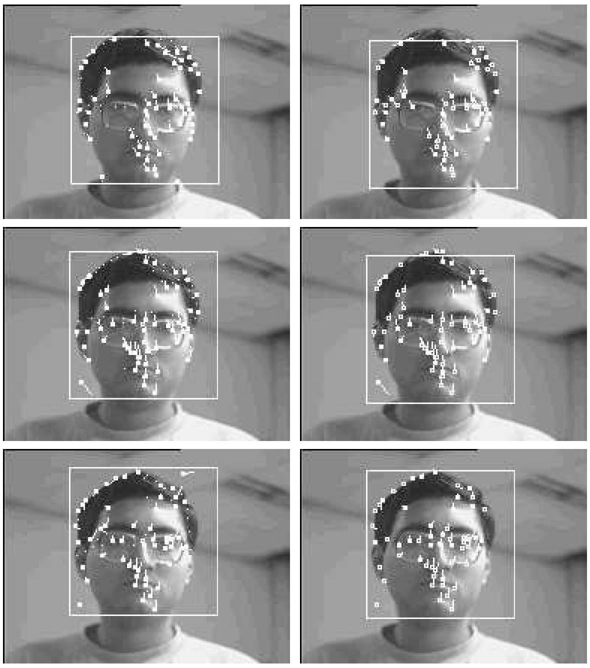

Figure 1: Frame numbers 10, 12 and 14 of the sequence. The figures on the left show the matched corners after Kalman filtering. The figures on the right show the matches after the entire algorithm. The tail with each corner shows its displacement (scaled by a factor of 2). The big rectangle is the fovea under view.

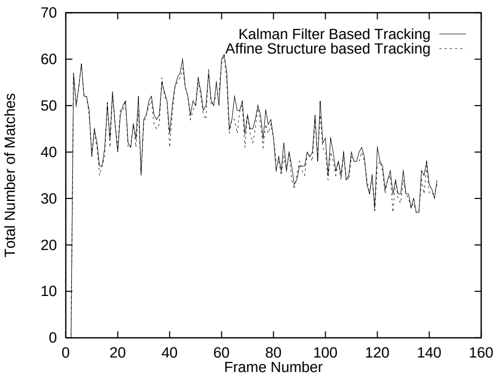

Figure 2: Total number of matches.

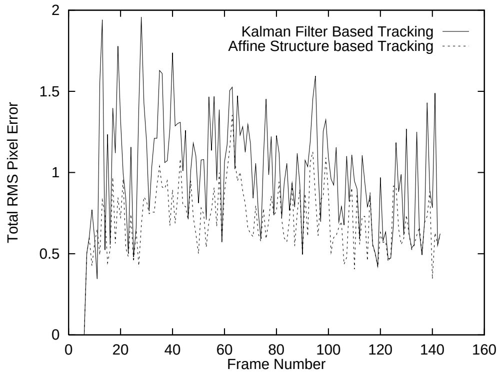

Figure 3: Average per-pixel error in prediction using structure and basis.

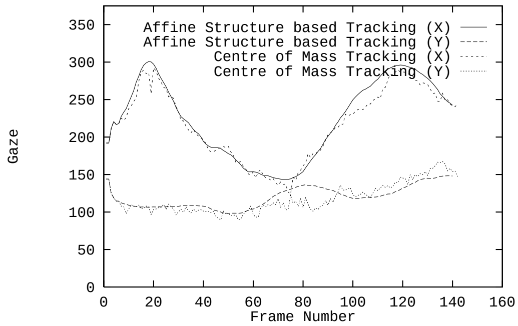

Figure 4: X and Y coordinates of the Gaze point.

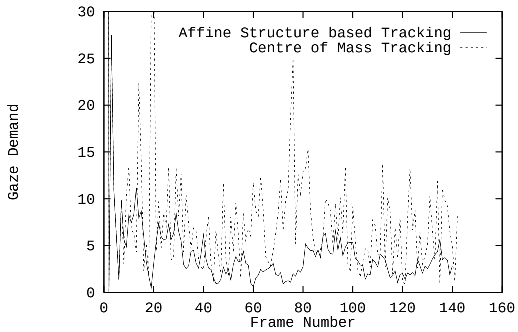

Figure 5: Camera speed.

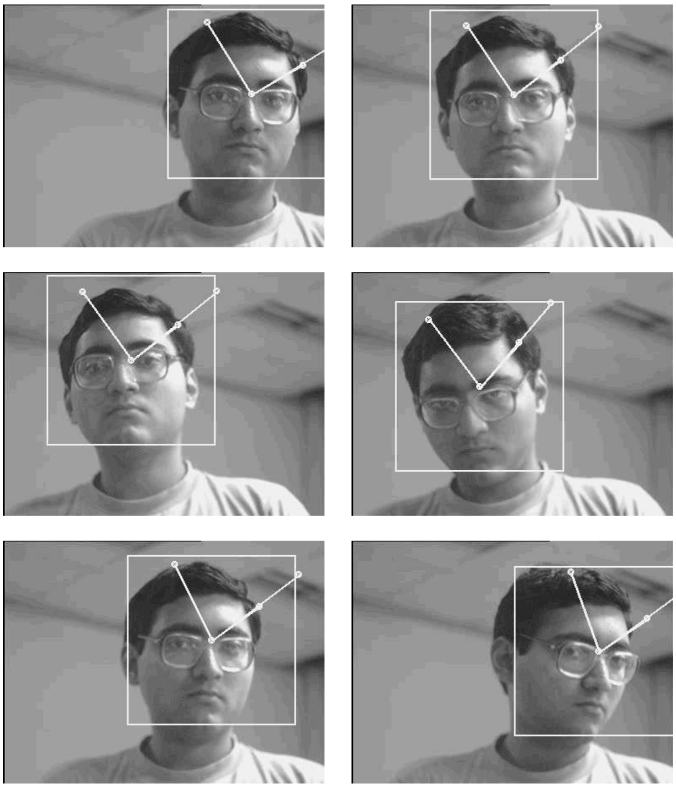

Figure 6: The corresponding basis computed in Frame numbers 20, 40, 60, 80, 100 and 120 (left to right and top to bottom). In this particular case the second and the third basis vectors nearly coincide in the image projections. For visual clarity a circle has been put at the end of the third basis vector

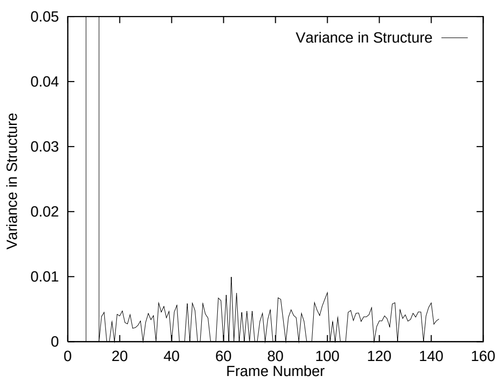

Figure 7: Average variance of affine structure coordinates.

5 Conclusions

We have presented a tracking algorithm based on affine structure which significantly improves the performance over just Kalman filter based tracking with very little extra computational overheads. The algorithm does not suffer from problems like the occlusion and disappearance of corners and never requires a change of basis. The basis computation is done from a large set of corners which leads to higher accuracy. The average per-pixel error in prediction of image positions of corners from the computed structure and basis is also smaller. Consequently, the gaze demand is smoother and more accurate. Moreover, such improved correspondences and structure and basis computation may be useful for other applications like object recognition. The algorithm is fast and amenable to real-time implementation.

Acknowledgments

We wish to thank Vardhan Verma, Himanshu Nautiyal and Ashish Thusoo for their help in the course of this work.

References

[1] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic Press., 1988.

[2] A. Blake, R. Curwen, and A. Zisserman. A Framework for Spatio-temporal Control in the Tracking of Visual Contour. *Intl. J. Computer Vision*, 11(2):127–145, 1993.

[3] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. 4th Alvey Vision Conf.*, pages 153–158, 1988.

[4] J. J. Koenderink and A. J. van Doorn. Affine Structure from Motion. *Journal of the Optical Society of America, Series A*, 8:377–385, 1991.

[5] J. L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision.* MIT Press, 1992.

[6] I. D. Reid and D. W. Murray. Tracking Foveated Corner Clusters using Affine Structure. In *Proc. Intl. Conf. Computer Vision*, pages 76–83, 1993.

[7] I. D. Reid and D. W. Murray. Active Tracking of Foveated Feature Clusters using Affine Structure. *Intl. J. Computer Vision*, 18(1):41–60, 1996.

[8] G. Sudhir, S. Banerjee, and A. Zisserman. Finding Point Correspondences in Motion Sequences Preserving Affine Structure. In *Proc. British Machine Vision Conf.*, 1993. Also accepted for CVGIP: Image Understanding.

[9] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Method. *Intl. J. Computer Vision*, 9:137–154, 1992.