Stochastic Motion Clustering

P H S Torr and D W Murray

Department of Engineering Science, University of Oxford
Parks Road, Oxford, OX1 3PJ, UK
Email: phst|dwm@robots.oxford.ac.uk

Abstract. This paper presents a new method for motion segmentation, the clustering together of features that belong to independently moving objects. The method exploits the fact that two views of a rigidly connected $3D$ point set are linked by the 3×3 Fundamental Matrix which contains all the information on the motion of a given set of point correspondences. The segmentation problem is transformed into one of finding a set of Fundamental Matrices which optimally describe the observed temporal correspondences, where the optimization is couched as a maximization of the *a posteriori* probability of an interpretation given the data. To reduce the search space, feasible clusters are hypothesized using robust statistical techniques, and a multiple hypothesis test performed to determine which particular combination of the many feasible clusters is most likely to represent the actual feature motions observed. This test is shown to be to computable in terms of a 0-1 integer programming method, alleviating the combinatorial computing difficulties inherent in such problems.

1 Introduction

Motion is a powerful cue for image and scene segmentation in the human visual system as evidenced by the ease with which we see otherwise perfectly camouflaged creatures as soon as they move, and by the strong cohesion perceived when even disparate parts of the image move in a way that could be interpreted in terms of a rigid motion in the scene.

Motion segmentation is often an essential part of the application of visual sensing to robotics. It turns out to be a most demanding problem, receiving considerable attention over the years [19, 18, 1, 14, 13, 10, 8, 5, 16]. As in [13, 8], we here pursue an optimal approach to the detection of independent motion, but one which works with unknown camera calibration and unknown camera motion. The case for algorithms that do not require calibration has been strongly made in [4]. Camera calibration is often impractical, especially when parameters are changed during the course of processing, and can introduce correlated errors into the system. As motion segmentation is a precursor to structure and motion recovery it appears unwise to predicate segmentation on knowledge of the camera motion. Instead, our approach can incorporate partial or full calibration and motion information if available, but it does not require them.

We adopt a scene-based rather than imaged-based segmentation, as suggested in [13], but, instead of adopting a specific scene model, utilise the fact that two views of a static $3D$ point set are linked by the 3×3 *Fundamental Matrix* [**F**] relating the uncalibrated image coordinates of the points seen from the two

Lecture Notes in Computer Science, Vol. 801
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94
© Springer-Verlag Berlin Heidelberg 1994

camera positions [7, 4, 3]. The Fundamental Matrix encapsulates the epipolar geometry and contains all the information on the camera's intrinsic and extrinsic parameters from a given set of point correspondences.

The method in overview is:

1. Generate hypothetical clusters by randomly sampling a subset of matches and using them to calculate a solution. Include in the cluster all matches consistent with that solution.
2. Prune solutions that have too few elements or are the same as existing ones.
3. Perform a multiple hypothesis test to determine which particular combination of the many feasible clusters is most likely to represent the actual motions.

2 Optimal Clustering

Consider a set of observed motion data, consisting of temporal matches of image features between two frames. The segmentation problem is to group the data into several clusters arising from independently moving objects, together with a cluster of unexplained data, in a way that maximizes the probability of the underlying interpretation Θ given the motion data D.

The most likely interpretation is obtained by maximizing the joint likelihood function of the measurements over all possible interpretations:

$$\max_{\Theta} \Pr[\Theta|D] \quad . \tag{1}$$

Using Bayes' Theorem this may be rewritten as finding

$$\max_{\Theta} \left(\frac{\Pr[D|\Theta] \Pr[\Theta]}{\Pr[D]} \right) \tag{2}$$

and, as the prior $\Pr[D]$ does not depend on Θ, this is further simplified to finding

$$\max_{\Theta} \left(\Pr[D|\Theta] \Pr[\Theta] \right) \quad . \tag{3}$$

In the motion segmentation problem this optimization is far from straightforward. However we shall demonstrate a method, initially proposed for multi-target tracking problems [12, 2], which converts the optimization into a 0-1 integer programming problem. First though, we will develop a probabilistic model for $\Pr[D|\Theta]$ and $\Pr[\Theta]$.

3 Probabilistic Models

We define a segmentation or interpretation Θ as a set of s clusters such that (i) each measurement belongs to a cluster, $D(\kappa_1) \cup D(\kappa_2) \cdots \cup D(\kappa_s) = D$; and (ii) each measurement belongs to only one cluster, $D(\kappa_i) \cap D(\kappa_j) = 0$; where $D(\kappa_j)$ is the set of matches in cluster κ_j. The last cluster, κ_s, is a cluster containing all unexplained (possibly mis-matched) data. It should be noted that the number of clusters may vary from hypothesized segmentation to segmentation. It is also assumed that the data in each cluster is independent from other clusters, so that $\Pr[D|\Theta] = \prod_{j=1}^{s} \Pr[D(\kappa_j)|\Theta]$.

3.1 Cluster model

One way in which motion clustering algorithms can fail is when the motion models they employ are too restrictive. For example, clustering based on similarity of image velocities alone [17] will fail when a static scene is viewed by a camera undergoing cyclotorsion. Again, schemes based on linear variation of the motion flow field [1, 20] will produce spurious segmentations at depth discontinuities when the camera is translating. The need for a more general model is apparent.

A completely general model of rigid motion, even given an uncalibrated camera, is provided through the Fundamental Matrix. Suppose that a set of points is correctly clustered, and arise from an object which has undergone a rotation and non-zero translation. After the motion, the set of homogeneous image points $\{\mathbf{x}_i\}, i = 1, \ldots N$, is transformed to the set $\{\mathbf{x}'_i\}$ related by

$$\mathbf{x}_i'^T [\mathbf{F}] \mathbf{x}_i = 0 \tag{4}$$

where $[\mathbf{F}]$ is the 3×3 Fundamental Matrix [7, 4]. Thus there is a constraint linking rigid motion in the world to inhomogeneous image coordinates in image one (x, y) and image two (x', y'), viz:

$$f_1 x'_i x_i + f_2 x'_i y_i + f_3 x'_i + f_4 y'_i x_i + f_5 y'_i y_i + f_6 y'_i + f_7 x_i + f_8 y_i + f_9 = 0 \ . \tag{5}$$

When there is degeneracy in the data such a unique solution for $[\mathbf{F}]$ cannot be obtained, it is desirable to adopt a simpler motion model. For small independently moving objects, there may be an insufficient range of x and y to determine the quadratic constraint, and we use a linear constraint. If all the points in three space lie within a small field of view and have a small depth variation the quadratic terms in Equation (5) become insignificant and we are left with the Fundamental Matrix reduced to its affine form [21, 16]

$$[\mathbf{F}]_A = \begin{bmatrix} 0 & 0 & f_3 \\ 0 & 0 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \tag{6}$$

and all the points obey the constraint

$$f_3 x'_i + f_6 y'_i + f_7 x_i + f_8 y_i + f_9 = 0 \ . \tag{7}$$

Whichever constraint is used (see later), the computed Fundamental Matrix for a whole cluster provides an epipolar line for each feature in that cluster, and the quality of the proximity of the match to the epipolar yields a measure of whether the cluster is a good one.

Given a cluster, we wish to evaluate the likelihood of its existence, $\Pr[D(\kappa_j)|\kappa_j]$, where $D(\kappa_j)$ are the n_j data items comprising cluster κ_j. Let us assume that the distance from a match to its epipolar line is normally distributed with zero mean and variance σ^2, a value which we can estimate from an analysis of the characteristics of the feature matcher. The sum of squares of these distances is a χ^2 variable with degrees of freedom $N_j = n_j - 7$ (or $n_j - 4$ in the case of the

affine camera). If we let V_j be the sum of squares divided by the variance then the probability density function of the cluster is

$$Pr[D(\kappa_j)|\kappa_j] = \frac{1}{2^{d_j}\Gamma(d_j)} V_j^{d_j-1} e^{-\frac{V_j}{2}} \quad , \tag{8}$$

where $d_j = \frac{N_j}{2}$ and $\Gamma()$ is the Gamma function [11]. The matches not originating from any cluster (mis-matches or false matches) are modelled as independent identically distributed events with a uniform probability density function $\frac{1}{v}$ and so

$$\Pr[D(\kappa_s)|\kappa_s] = \left(\frac{1}{v}\right)^{n_s} \tag{9}$$

where n_s is the number of mis-matches.

In [13] a prior $\Pr[\Theta]$ was used which favoured spatial clustering. In this work however potential clusters are hypothesized by a cluster generation stage, and so we choose $\Pr[\Theta]$ as a uniform distribution. Thus the total likelihood function for a given partition is:

$$\Pr[D|\Theta] = \Pi_{j=1}^{s-1} \left[\frac{1}{2^{d_j}\Gamma(d_j)} V_j^{d_j-1} e^{-\frac{V_j}{2}}\right] \left(\frac{1}{v}\right)^{n_s} \quad . \tag{10}$$

The optimum segmentation is that which maximizes the likelihood function given in Equation (10). One approach might be to form every segmentation Θ of data into clusters and mis-matches and so arrive at the maximum likelihood data association, but of course this brute force approach is computationally feasible only in trivial cases. The key to reducing the computational complexity is inferring the existence of clusters. Thus, using theory developed for data association in tracking [12, 2], we have devised a clustering algorithm consisting of two parts: feasible cluster construction, followed by a Bayesian decision process that selects the optimal combination of clusters. We describe these in the next sections.

4 Cluster Hypothesis Generation

Given that a large proportion of our data may belong to different populations the approach is the opposite to conventional smoothing techniques. Rather than using as much data as is possible to obtain an initial solution and then attempting to identify the cluster, we randomly sample as small a subset of the data as is feasible to estimate the parameters to generate hypotheses.

To estimate the Fundamental Matrix we select seven points and form the data matrix:

$$[\mathbf{D}] = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_7x_7 & x'_7y_7 & x'_7 & y'_7x_7 & y'_7y_7 & y'_7 & x_7 & y_7 & 1 \end{bmatrix} \quad . \tag{11}$$

The selection process exploits spatial cohesion by randomly selecting one feature and its match, then selecting the six nearest feature matches within the image [1].

Examining the null space of this matrix, we find a one parameter family of solutions for $[\mathbf{F}]$: $\alpha[\mathbf{F}]_1 + (1-\alpha)[\mathbf{F}]_2$. Introducing the constraint $|[\mathbf{F}]| = 0$ [4] allows us to obtain a cubic in α from which we obtain 1 or 3 solutions—all of which are examined to see which gives the best result.

The cluster of seven points is now grown by testing whether or not other feature pairs are consistent with the derived Fundamental Matrix. A t-test is made on the distance of the feature in each image to the epipolar line for that feature defined by the Fundamental Matrix. Effectively we are using a non-linear estimator to determine the Fundamental Matrix by minimizing the Euclidean distances of features to epipolar lines. Minimizing this distance has been shown to be superior to minimizing the residuals[9]. By introducing the t-test our method gains robustness to outliers.

Degeneracy in the data can be detected by testing if the null space of the matrix in Equation (11) is greater than 2. If this is the case, 4 of the 7 points are randomly selected and an affine matrix and epipolar lines computed instead [16].

We now calculate how many samples are required. Ideally, we would consider every possible sample, but this is computationally infeasible. Instead we choose m, the number of samples which gives a probability in excess of 95% that we have included a good sample within our selection. (Here a good sample means a set of low noise, consistent data.) The expression for this probability, Υ, is [15]:

$$\Upsilon = 1 - (1 - (1-\epsilon)^p)^m \quad , \tag{12}$$

where ϵ is the probability that each match is inconsistent with the rest, p the dimension of the parameter, m the number of samples needed. Figure 1 gives some sample values of the number m of samples required to ensure $\Upsilon \geq 0.95$ for given p and ϵ. The utilization of spatial cohesion for cluster generation significantly reduces ϵ, so we select $m = 500$.

Dimension	ϵ						
p	0.1	0.2	0.3	0.4	0.5	0.6	0.7
4	3	6	11	22	47	116	369
7	5	13	35	106	382	1827	13696

Fig. 1. *The number m of samples required to ensure $\Upsilon \geq 0.95$ for given p and ϵ, where Υ is the probability that all the data points we have selected in one of our samples are consistent.*

4.1 Pruning

After the initial cluster generation, clusters deemed too small are pruned. To determine what is a significant size for the cluster we calculate the probability

[1] An extension to the method would be to sample from the set of all candidate matches proposed by the feature matcher, only accepting matches as verified if they are included in a valid cluster. This would allow the segmentation to guide the matching process.

that cluster might have occurred randomly. Given a trial cluster associated with Fundamental Matrix [**F**], we accept a point as belonging to that cluster if it lies within a certain distance b of its epipolar line determined by the t-test threshold t_α. If the image size is $l \times l$ pixels then the maximum area that is swept out in the image within b of an epipolar line is below $2\sqrt{2}bl$. This gives the probability of given point lying on an epipolar line by chance as $\xi = 2\sqrt{2}b/l$.

Now the probability of obtaining a cluster of size k follows a Binomial distribution:

$$\Pr(k) = \binom{n}{k} \xi^k (1-\xi)^{n-k} \quad , \tag{13}$$

which, if $\mu = n\xi < 5$ and $n > 20$, can be well-approximated by a Poisson distribution:

$$\Pr(k) = \frac{e^{-\mu}\mu^k}{k!} \quad . \tag{14}$$

Summing these probabilities for $k = 0, 1, \ldots, h$ gives us the probability $\Pr(0 \leq k \leq h)$ of a cluster existing smaller or equal in size to h. In our work we prune all clusters κ_j with size $n_j \leq h$ such that $\Pr(0 \leq k \leq h) \leq 0.05$.

4.2 Merging clusters

Because of the way the hypotheses are generated, there may be some clusters that possess a major overlap. If the distances to epipolar lines are viewed as identically distributed Gaussian variables with zero mean, then the sum of the squares of these distances divided by their variance is a χ^2 variable and one can apply a χ^2 test to see whether or not two clusters should be merged.

However, if this was done for every pair of solutions there would be a significant rise in computation. To avoid this, a threshold β is set on a measure of the distance between any two solutions $\mathbf{f}_a$ and $\mathbf{f}_b$, and the merge χ^2 test only performed if

$$\mathbf{f}_a^T \mathbf{f}_b < \beta \quad . \tag{15}$$

The vector $\mathbf{f} = (f_1, \ldots, f_9)^T$ is made from the elements of the Fundamental Matrix [**F**].

5 Selecting a segmentation

In this section we shall show, given a feasible cluster set, that the optimization process to achieve segmentation can be cast as a 0-1 integer programming problem, one for which efficient computer algorithms exist.

Suppose that we have l postulated clusters. We represent a segmentation Θ as a binary vector $\boldsymbol{\Theta}$ with a number of elements equal to the number of hypothesized clusters, i.e.

$$\boldsymbol{\Theta} = (\theta_1, \theta_2 \cdots \theta_l)^T \quad , \tag{16}$$

where the jth cluster is included in the partition if $\theta_j = 1$ and not otherwise. For example, suppose we generated 5 potential clusters, then a segmentation corresponding to $\{\kappa_1, \kappa_3, \kappa_5\}$ would be represented by $\boldsymbol{\Theta} = (1, 0, 1, 0, 1)^T$.

Let the negative log-likelihood function of cluster κ_j be:

$$c_j = -\ln(\Pr[D(\kappa_j)|\kappa_j]) \tag{17}$$

where $D(\kappa_j)$ are the n_j data items in cluster κ_j. If we define $\mathbf{c}$ with jth element c_j, the negative log-likelihood function of all the measurements for the segmentation Θ under consideration is

$$\mathbf{c}^T\boldsymbol{\Theta} + c_s \quad , \tag{18}$$

where c_s is the negative log-likelihood of the cluster of mis-matches

$$c_s = -n_s \ln v^{-1} \quad . \tag{19}$$

It would be convenient for the purposes of the integer programming algorithm to make Equation (18) purely the inner product of $\boldsymbol{\Theta}$ and a vector. Now c_s may be written as

$$c_s = -n_s \ln v^{-1} = -(n - \sum_j \theta_j n_j) \ln v^{-1} \tag{20}$$

where $n = \sum_j n_j$ is the total number of matches. Let us also define

$$\tilde{c_j} = c_j + n_j \ln v^{-1} \quad . \tag{21}$$

Then, if we create vector $\tilde{\mathbf{c}}$ with jth element $\tilde{c_j}$, we have:

$$\mathbf{c}^T\boldsymbol{\Theta} + c_s = \tilde{\mathbf{c}}^T\boldsymbol{\Theta} \quad . \tag{22}$$

In order to enforce the constraint that each match may only be in one cluster let us define the matrix $[\mathbf{A}]$ such that $A_{ij} = 1$ if the jth postulated cluster contains the ith datum and 0 otherwise. Then the optimization problem can be expressed as finding:

$$\begin{array}{l} \text{minimize} \quad [\tilde{\mathbf{c}}^T\boldsymbol{\Theta}] \\ \text{subject to } [\mathbf{A}]\Theta \leq 1 \end{array} \tag{23}$$

To solve this we have used the branch and bound method from NAG Library [6]. Note any cluster with $\tilde{c_j} > 0$ can be pruned as it will never form part of the feasible set.

6 Results

The first set of experiments has been performed on computer generated data. Points were generated at random 3D positions and imaged by a moving synthetic camera with a field of view of 75° (corresponding to an image size of 500 × 500 units and focal length 325 units). Between views the camera was translated a random amount between 0 and 100 units in a random direction and rotated about a randomly chosen axis between 0 and 2 degrees. After imaging, points were perturbed by Gaussian noise with zero mean and standard deviation of unity. Features belonging to another moving object were then introduced. The object had various sizes, giving rise to 10 to 50% of the total number of features, the latter being typically 250.

For each size of object, the random experiments were repeated 100 times and graphs showing the percentage of correctly classified motion matches on both the large and small object, as a function of the small objects relative size. The results are shown in Figure 2, which is a graph of the percentage of each object correctly identified against the size of the smaller object as a percentage of the total number of features. It can be seen that the algorithm performs best if there is a larger object and a smaller object, but still gives good results when the two objects are of equal size.

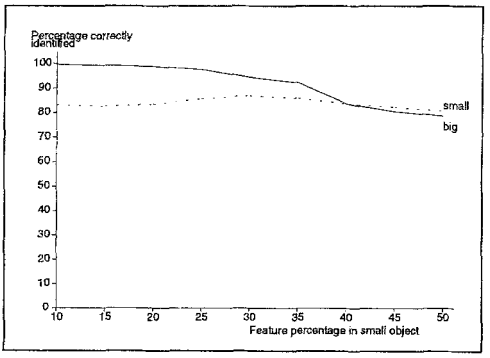

Fig. 2. *Showing the percentage of the objects correctly identified.*

Figure 3 shows typical results from experiments with real imagery. The top-left image shows motions matches obtained from two heads nodding and wagging in front of a stationary background. Two of the segmented clusters are shown at the top right and bottom-left, and the unsegmented matches shown at the bottom-right. A third cluster containing just stationary background points was also detected but is not shown here. It can seen that the largest cluster of features is on the right hand face and this also grabs some features on the left which happen to be consistent with the motion of both faces. We suggest that this sort of ambiguity can only be resolved over time, by using a sequence of images.

A considerable problem with stochastic segmentation algorithms has been their lack of speed. By introducing the hypothesis and test paradigm, the computation time of the present algorithm has been reduced substantially. In the example on real data, the segmentation algorithm took only 10% of time taken to compute and match corner features.

7 Conclusion

In this paper we have shown how robust methods may be used to solve the problem of motion segmentation. Our methods are based on scene constraints rather than image based approximations. We have set out a rigorous maximum likelihood analysis of the problem and eschewed heuristics. Combinatorial difficulties that might otherwise prevent the use of a Bayesian analysis have been alleviated

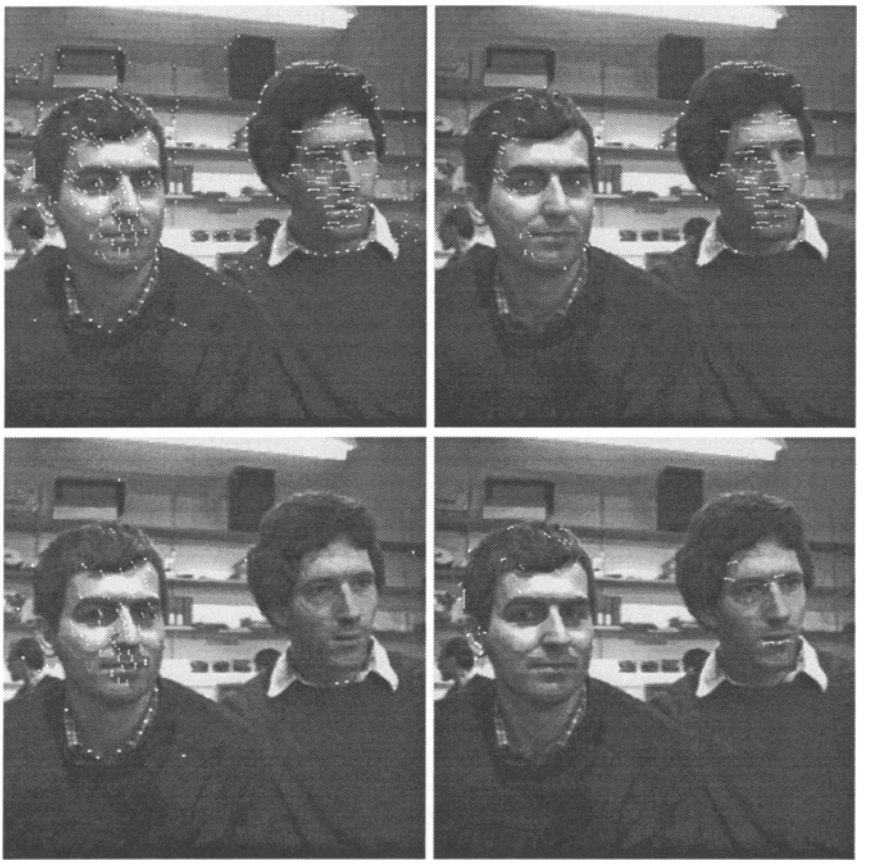

Fig. 3. *Showing an image taken from a smoothed sequence of a two heads rotating. The top left image shows the detected feature matches for two images in the sequence. All the points that are tested to see whether or not they are stationary. The top right shows the largest moving cluster detected, the bottom left the next largest cluster and the bottom right the mis-matches.*

through the use of 0-1 integer programming techniques, thus ensuring that the segmentation algorithm does not provide a significant computational overhead to the feature detection and matching algorithm. Indeed it has been shown that the segmentation algorithm may be used to aid the feature matching process.

References

1. G. Adiv. Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field. In *Proceedings, CVPR '85 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, June 10–13, 1985)*, IEEE Publ. 85CH2145-1., pages 70–77. IEEE, IEEE, 1985.

2. Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.
3. S. Demey, A. Zisserman, and P. A. Beardsley. Affine and projective structure from motion. In D. Hogg, editor, *Proc. British Machine Vision Conference*, pages 49–58. Springer-Verlag, Sept 1992. Leeds.
4. O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of 3rd European Conference on Computer Vision*, pages 563–578, 1992.
5. E. François and P. Bouthemy. Multiframe based identification of mobile components of a scene with a moving camera. In *Proc. CVPR.*, 1991.
6. Numerical Algorithms Group. *NAG Fortran Library vol 7.* NAG, 1988.
7. R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proc. ECCV-92*, pages 579–87, 1992.
8. F. Heitz and P. Bouthemy. Multimodal motion estimation and segmentation using markov random fields. In *Proc. 10th Int. Conf. Pattern Recognition*, pages 378–383, 1991.
9. Q. T. Luong, R. Deriche, O. D. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: analysis of different methods and experimental results. Technical Report 1894, INRIA (Sophia Antipolis), 1993.
10. P. F. McLauchlan, I. Reid, and D. W. Murray. Coarse image motion for saccade control. In D. Hogg, editor, *Proc. British Machine Vision Conference.* Springer-Verlag, Sept 1992. Leeds.
11. P. L. Meyer. *Introductory Probability and Statistical Applications.* Addison-Wesley, 1970.
12. C. L. Morefield. Applications of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, 22:302–312, 1977.
13. D.W. Murray and B.F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:220–228, 1987.
14. D.W. Murray and N.S. Williams. Detecting the image boundaries between optical flow fields from several moving planar facets. *Pattern Recognition Letters*, 4:87–92, 1986.
15. P. J. Rousseeuw. *Robust Regression and Outlier Detection.* Wiley, New York, 1987.
16. L.S. Shapiro. *Affine Analysis of Image Sequences.* PhD thesis, Oxford University, 1993.
17. S. M. Smith and J. M. Brady. A scene segmenter; visual tracking of moving vehicles. In *Intelligent Autonomous Vehicles*, pages 119–126, 1993.
18. W.B. Thompson and T.C. Pong. Detecting moving objects. *International Journal of Computer Vision*, 4(1):39–58, 1990.
19. S. Ullman. Relaxed and constrained optimisation by local processes. *Computer Graphics and Image Processing*, 10:115–125, 1979.
20. A.M. Waxman and J.H. Duncan. Binocular image flows: Steps toward stereo-motion fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:715–729, 1986.
21. A. Zisserman. Notes on geometric invariance in vision. BMVC Tutorial, 1992.