# Recovery of Egomotion and Segmentation of Independent Object Motion Using the EM Algorithm

W. James MacLean,* Allan D. Jepson† & Richard C. Frecker*
University of Toronto, Toronto, Canada M5S 1A1

**Abstract**

This paper examines the use of the EM algorithm to perform motion segmentation on image sequences that contain independent object motion. The input data are linear constraints on 3-D translational motion and bilinear constraints on 3-D translation and rotation, derived from computed optical flow using subspace methods. The problems of outlier detection, deciding how many processes, and the initial guesses for the EM algorithm are considered. Results obtained from an image sequence are presented.

## 1 Introduction

In order for an observer to navigate in its environment, it is important that the observer can detect other independently moving objects and avoid collisions. The motion of the observer complicates this task. For the purpose of this paper we divide image motion into two categories: *egomotion* and *motion due to independent moving objects*. *Egomotion* is defined as the image motion induced by an observer moving through a static environment. *Motion due to independently moving objects* is defined as the image motion induced by the movement of an object relative to the observer when that object is not stationary with respect to the environment at-large. It is possible to recover both the observer's motion relative to its environment and a relative depth map for the environment from the captured images [5, 4]. The recovery of correct 3-D motion parameters relies on segmenting the *optic flow* into distinct regions that correspond to unique 3-D motions. An image sequence containing independent object motion allows proper recovery of relative motion parameters only if the image can be segmented into regions, each of which corresponds to a distinct relative motion.

Some work has already been done on the problem of motion segmentation. We first consider work done on the problem of 2-D segmentation. Darell & Pentland [2] used a method that assigned 2-D constraints to different regions using a competitive and iterative algorithm, but only for the case of translational motion. Jepson & Black [10] used a mixture-model approach to cluster component

*Institute of Biomedical Engineering, Department of Electrical & Computer Engineering
†Department of Computer Science & Canadian Institute for Advanced Research

Figure 1: This is a frame from a sequence (of 10 frames) collected by a robot-observer translating roughly along the optical axis in an industrial environment. The forklift and its driver are translating to the right. The boxes indicate image regions for which affine or rational models for optic flow have been fitted. The focus-of-expansion (FOE) of the background motion for each frame in the sequence have been indicated by a '$\times$' (see Section 5)

velocities, and hence achieve improved optic flow estimates. Their method allows shared ownership of constraints amongst regions. Wang & Adelson [17] segmented image regions into patches whose optic flow at any point could be modelled as an affine transformation of the image coordinates of that point. The segmentation was achieved using a K-means approach. These methods do segmentation in 2-D, and attempt to solve the problem of proper integration of constraints.

There have also been attempts at segmentation based on 3-D motion. Adiv [1] identified regions in the image whose motion was consistent with the movement of a planar surface, and grouped these according to their mutual consistency for various 3-D motions. Sinclair [16] segments images by recovering the 3-D angular velocity field for the image, and using a simple clustering algorithm for identifying planes in angular velocity space. This method also requires identifying planar surfaces in the image. Both of these methods require the existence (and identification) of planar surfaces in the image. Nelson [14] describes a method which could properly be thought of as a 3-D method. Given the observer motion, he compares the expected motion field against measured component velocities, and where significant deviation is found assumes independent object motion. This method has the drawback of requiring *a priori* knowledge of the observer motion, and does not attempt to distinguish between different independent moving objects.

In this paper we present a method for motion segmentation based on clustering constraints on 3-D translational velocity. These constraints are derived using subspace methods, which have the advantage of not being sensitive to depth discontinuities in the static environment (in fact, it benefits from them). The clustering is achieved through the application of the EM algorithm to the constraints, using a finite-mixture model. The results of this clustering are then used to provide an initial guess for parameter fitting using bilinear constraints on translation and rotation. We first give a brief over-view of the subspace methods, then define mixture models and the EM algorithm. Results are given for an image sequence from an industrial environment

## 2 Constraints on 3-D Relative Motion

A relative motion can be described by its translation, $\vec{T}$, and rotation, $\vec{\Omega}$. The rotation is about an axis which passes through the nodal point of the imaging system, which is defined as the origin in our coordinate system. We consider a point in 3-D space, $\vec{X} = (X_1, X_2, X_3)^T$, where $X_3$ lies along the optical axis of the camera. The motion field at the image of this point, namely $\vec{x} = (x_1, x_2, f) = \frac{f}{X_3}\vec{X}$, can be defined in terms of the motion parameters [9]:

$$\vec{u}(\vec{x}) = \begin{bmatrix} 1 & 0 & -x_1/f \\ 0 & 1 & -x_2/f \end{bmatrix} \left( \frac{f}{X_3(\vec{x})}\vec{T} + \vec{\Omega} \times \vec{x} \right) \tag{1}$$

where $\vec{T}$ and $\vec{\Omega}$ are motion of the background with respect to the observer, $f$ is the focal length of the system, and $X_3(\vec{x}) = X_3$ is the projection of $\vec{X}$ onto the optical axis. The flow field can be thought of as having two components—a translational component and a rotational component. Note that only the translational component is affected by the distance to points in the image. Therefore, any discontinuities in the optic flow field must be due to variations in depth,[1] a fact exploited by Rieger & Lawton [15] in their method for recovering translational motion. It is also exploited by the subspace methods.

A simple algebraic manipulation of Eqn. 1 [9] allows us to derive the following *bilinear* constraint on $\vec{T}$ and $\vec{\Omega}$:

$$\vec{T}^T(\vec{x} \times \vec{u}(\vec{x})) + (\vec{T} \times \vec{x})(\vec{x} \times \vec{\Omega}) = 0 \tag{2}$$

This is an exact constraint on the motion field, although it is non-linear in the motion parameters. Only a single flow vector (and its image location) are required to define each constraint. This constraint is also independent of the depth of the point imaged at $\vec{x}$. Eqn. 2 can be rewritten as $\vec{T}^T(\vec{a}(\vec{u}) + B\vec{\Omega}) = 0$ where $\vec{a}$ is $3 \times 1$ and $B$ is $3 \times 3$. Both are functions of $\vec{x}$, and $\vec{a}$ is also a function of $\vec{u}$.

It is possible to derive a *linear* constraint on $\vec{T}$ from 7 or more bilinear constraints [7]. Given optic flow sampled at K discrete points in the image, $\{\vec{x}_k\}_{k=1}^K$, we construct a constraint vector $w_i\vec{\tau}_i = \sum_{k=1}^K c_{ik}\,[\vec{u}(\vec{x}_k) \times \vec{x}_k]$. Here $\|\vec{\tau}_i\| = 1$, and $w_i$ is the norm of the right-hand side of the expression. Through suitable choice of the $\vec{c}_i = [c_{i1} \ldots c_{iK}]^T$ we can guarantee that the constraints $\{\vec{\tau}_i\}_{i=1}^N$ will be orthogonal to $\vec{T}$, *i.e.* $\vec{\tau}_i^T\vec{T} = 0, i = 1 \ldots N$. From Eqn. 2 we see that a sufficient condition on the $\vec{c}_i$ is that they are orthogonal to all quadratic forms involving

[1] These variations in depth can be classified into two types, depending on whether the depth variation is due to a boundary formed by an independently moving object or not. The former is of importance to motion segmentation using the subspace methods.

$[\vec{x}_k]_1$ and $[\vec{x}_k]_2$. This effectively annihilates the contribution due to $\vec{\Omega}$. In the absence of independently moving objects we would expect all the constraints shown to intersect at a common point, the *focus-of-expansion* (FOE). The presence of the forklift which is moving to the right causes additional constraints which are inconsistent with the background motion, hence it becomes necessary to segment the constraints based on the underlying 3-D motions.

Since the $\vec{c}_i$ are orthogonal to all quadratics in image location, the technique requires a variation in depth that is not planar over the image region from which the optic flow is sampled to create a non-zero constraint. The practical importance of this is that no constraint can be generated if all the flow samples come from a single planar surface. The constraints are generated by pair-wise combining flow samples from the boxes in Figure 1 (this is done since flow samples from a single box are consistent with a planar surface—this is a consequence of the affine model used to estimate the flow). In the event that the box representing the forklift is paired with a box from the background, the fundamental rigidity assumption of the subspace methods is not met. Note that if *a priori* segmentation information is available, then the constraints can be generated using custom masks that never cross independent object motion boundaries. Generation of suitable $\vec{c}_i$ coefficients is straightforward once the sampling geometry is known

# 3 Mixture Models

When a set of data has more than one underlying process, *i.e.*, any given data point in the set will have been generated by one of several processes, the concept of a mixture of distributions is useful [12]. Each process[2] will have its own distribution and parameters. Our task is to i) estimate the parameters for each process, and ii) determine the probability that a given data point is the result of a given process. We assume in advance that we know the number of underlying processes and the form of each corresponding distribution. Testing for the number of processes in a mixture is a difficult and, in general, unsolved problem [12]. Part ii) of this objective is commonly referred to as *clustering*. We can consider our linear and bilinear constraints on relative motion as observations arising from one of several underlying motion processes. We first consider mixtures involving linear constraints and translational motions. The probability density function (PDF) of an observed constraint $\vec{\tau}_i$ with respect to a number of underlying translations $\left\{\vec{T}_j\right\}_{j=1}^{M}$ can be written as

$$p(\vec{\tau}_i|\vec{T}_1, \sigma_1 \ldots \vec{T}_M, \sigma_M) = \pi_0 p_0 + \sum_{j=1}^{M} \pi_j p(\vec{\tau}_i|\vec{T}_j, \sigma_j), \sum_{j=0}^{M} \pi_j = 1, 0 \leq \pi_j \leq 1$$

where $M$ is the number of processes. The variances $\{\sigma_j\}_{j=1}^{M}$ depend on the noise in the optic flow. The $\pi_j$ are positive valued constants representing the mixing proportions of the distributions. Both the $\pi_j$ and $(\vec{T}_j, \sigma_j)$ parameters may be

[2] The processes are also referred to as *populations* or *modes*.

unknown, and can be estimated given the data. We take the form of the PDF's to be a Gaussian modified for the unit sphere,

$$p(\vec{\tau}_i|\vec{T}_j, \sigma_j) = \frac{3}{4\pi(2+\exp\{-1/2\sigma_j^2\})} \exp\left\{-\frac{(\vec{\tau}_i^T\vec{T}_j)^2}{2\sigma_j^2}\right\} \quad p_0 = \text{constant}$$

where $p_0$ is a uniform distribution meant to model outliers in the data [*cf.* [10]].

Once parameters are known or estimated, each observed constraint can be assigned an ownership probability to each process, $j = 0 \ldots M$:

$$s_{ij} = \pi_j p(\vec{\tau}_i|\vec{T}_j, \sigma_j)/p(\vec{\tau}_i|\vec{T}_1, \sigma_1 \ldots \vec{T}_M, \sigma_M)$$

Similarly for bilinear constraints we can write

$$p(\vec{u}_i|\vec{T}_j, \vec{\Omega}_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left\{-\frac{(\vec{T}_j^T(\vec{a}_i(\vec{u}_i) + B_i\vec{\Omega}_i))^2}{2\sigma_j^2}\right\}$$

where $(\vec{T}_j, \vec{\Omega}_j)$ represent the underlying motion process. It should be noted that the $\sigma_j$'s will not be the same between the linear and bilinear models.

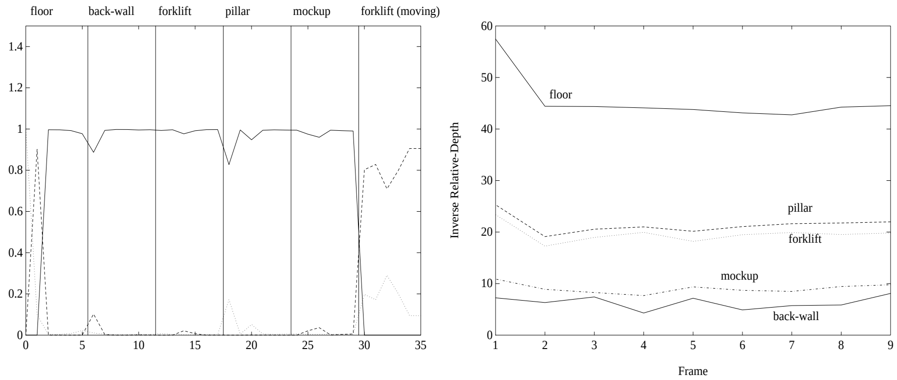
# 4 Application of the EM Algorithm

While the mixture model provides a powerful way of modeling the constraint data, it also requires careful thought in order to proceed to a solution. If the ownership probability of each constraint were known, then it is straightforward to calculate the motion parameters. Conversely, if the motion parameters were known then assigning ownership is again relatively straightforward. We expect outliers to arise due to constraints being generated across independently moving object boundaries, as well as from errors in recovered optic flow.

A number of researchers have used the EM algorithm to estimate the parameters of mixture models [12, 3]. The EM-algorithm is an iterative, 2-step method where "EM" stands for *expectation* & *maximization*, the two basic steps involved.

The algorithm starts with an initial guess for the motion parameters. The expectation step assigns an ownership probability for each constraint to each motion process on the assumption that the current motion parameters are the correct ones. The mixture proportions are also estimated. The maximization step solves for the motion parameters on the assumption that the assigned ownership values are correct. Each expectation-maximization pair constitutes one iteration of the algorithm. Dempster *et. al.*[3] have shown that each iteration of the EM algorithm is guaranteed to improve (or, at worst leave unchanged) the likelihood function of the model. The case in which the likelihood function is left unchanged corresponds to having found a local maximum in the likelihood function.

No results are given for the rate of convergence to a maximum likelihood point, although methods for improving the convergence rate have been proposed [13]. The existence, in general, of local maxima in the likelihood function leads to the importance of a good initial guess for the algorithm. A poor choice of initial guess may lead to slow convergence, or convergence to a local, not global, maximum.




a) b)

Figure 2: a) The ownership probabilities for the bilinear constraints as fitted by the EM algorithm. The solid line indicates ownership by the first motion, the dashed line the second motion, and the dotted line the outlier process. b) The recovered inverse-depth values for each patch over the sequence are plotted. The moving forklift is excluded, as depth values recovered for it are not valid. Note that the relative distances to various objects are correct.

## 4.1 Clustering Linear Constraints

The linear constraints $\{\vec{\tau}_i\}_{i=1}^N$ are exact in the absence of noise. They can be used to recover the direction of $\vec{T}$, but not its magnitude.[3] Once translational direction has been recovered, it is straightforward to recover the rotation and relative depth information. For the remainder of this paper we will only be interested in the direction of $\vec{T}$, and will assume $||\vec{T}|| = 1$. Recovery of translational direction can be accomplished by a linear least-squares technique. The translational direction will be given by the eigenvector corresponding to the smallest eigenvalue of the following matrix:

$$D = \sum_{i=1}^{N} w_i^2 \vec{\tau}_i \vec{\tau}_i^T \tag{3}$$

This is equivalent to minimizing $E(\vec{T}) = \vec{T}^T D \vec{T}$.

We wish to generate estimates for the underlying translational directions based on the linear constraints, $\{w_i \vec{\tau}_i\}_{i=1}^N$, $||\vec{\tau}_i|| = 1$ as well as determine the number of translational directions represented by these constraints. We start with an initial estimate for either one or two translations based on the $D$ matrix of Eqn. 3. Our estimate for $\vec{T}_1$ will be the eigenvector corresponding to the smallest eigenvalue of $D$. If the second smallest eigenvalue is also small compared to the largest, then a second translation $\vec{T}_2$ is hypothesized in the eigendirection corresponding to the second smallest eigenvalue of $D$. A further discussion of this point is given below.

At each "E"-step in the EM algorithm the ownership probability $s_{ij}$ of constraint i by process j is calculated as well as the mixture proportions $\pi_j = \alpha \sum_{i=1}^N s_{ij}$

[3] This is a feature inherent in the problem itself [6].

where $\alpha$ is determined by the fact that the $\pi_j$ sum to 1. Then, during the "M"-step the translation parameters $\vec{T}_j$ are re-evaluated by using the eigenvector corresponding to the smallest eigenvalue of

$$\hat{D}_j = \sum_{i=1}^{N} s_{ij} w_i^2 \vec{\tau}_i \vec{\tau}_i^T / \sum_{i=1}^{N} s_{ij} w_i^2 \tag{4}$$

as the new estimate. The variances are estimated by

$$\sigma_j^2 = \sum_{i=1}^{N} s_{ij} (\vec{\tau}_i \vec{T}_j^T)^2 / \sum_{i=1}^{N} s_{ij}$$

This continues until the parameters converge.

It is worthwhile to consider a geometric interpretation of the constraints. In the event that one eigenvalue of $D$ is significantly smaller than the other two, then the constraint vectors lie close to a great circle on the unit sphere, and the correct translational direction is the vector normal to the plane defined by this great circle. Once the EM algorithm has terminated, we re-examine our current estimate for the number of processes. This is done by testing for structure in process 0 (the outlier process). If we generate $\hat{D}_0$ and examine its eigenvalues ($\lambda_1 > \lambda_2 > \lambda_3$), we expect to find one of three cases:

$\lambda_1 > \lambda_2 \gg \lambda_3$. This indicates the possibility of one new translational direction, i.e. great circle. This case occurs when the constraints are clustered in an elliptical shape with the major axis significantly larger than the minor. This gives support for a single translational direction.

$\lambda_1 \gg \lambda_2 \approx \lambda_3$. This suggests that there may be two possible translational directions. This case occurs when all the constraints are close together and distributed roughly in a circular fashion. In this case there are two possible eigendirections for the translation to lie in.

$\lambda_1 \approx \lambda_2 \approx \lambda_3$. This indicates that the constraints in process 0 are distributed roughly equally in all 3 directions. There may or may not be unique underlying translations, but we have no indication of a preferred direction.

In order to distinguish between the first two possibilities we compare $\lambda_2$ to the geometric mean of the largest and smallest eigenvalues, namely $\sqrt{\lambda_1 \lambda_3}$. In either of the first two cases we add new translational directions, as defined by the eigenvectors of $\hat{D}_0$, to our mixture model and repeat the EM algorithm. This is repeated until either the mixing proportion of process 0 becomes too small, indicating it has ownership of few constraints, or until the new translational directions cease to be unique as compared to the processes already existing. This can be done by comparing

$$p(\vec{T}|D) = \frac{1}{k} \exp\left\{-\vec{T}^T D \vec{T}\right\}, k = \frac{4}{3}\pi(e^{-\lambda_1} + e^{-\lambda_2} + e^{-\lambda_3}) \tag{5}$$

## 4.2 Clustering Bilinear Constraints

The preceding section outlined a method for recovering the number of translational processes as well as estimating their directions. We now describe a method for clustering bilinear constraints and estimating rotational motion.

| |
|---|
| **Initial guesses:** |
| Process 1: $\vec{T}$ = [-0.0002 -0.0925 0.9957] $\vec{\Omega}$ = [ 0.33 7.98 4.69] $\vec{\sigma}$ = 3.48084 |
| Process 2: $\vec{T}$ = [-0.9948 0.0216 0.0996] $\vec{\Omega}$ = [ -2.94 -99.55 -6.55] $\vec{\sigma}$ = 0.64782 |
| **Final Results:** |
| Mixtures: 0.1866 0.7075 0.1059 |
| Process 1: $\vec{T}$ = [ 0.0102 -0.0925 0.9957] $\vec{\Omega}$ = [ 2.09 2.27 -0.10] $\vec{\sigma}$ = 0.07033 |
| Process 2: $\vec{T}$ = [-0.9948 0.0295 0.0972] $\vec{\Omega}$ = [ -4.13 -99.26 -5.18] $\vec{\sigma}$ = 0.05602 |
| Process 1: FOE = ( 13.20, -119.24) |
| Process 2: FOE = (-13137.98, 390.18) |

Table 1: Results from the fitting the bilinear constraints for the frame shown in Figure 1.

The number of processes is now fixed. For each estimated $\vec{T}_j$ we calculate a least-squares estimate for $\vec{\Omega}_j$:

$$\vec{\Omega}_j = \left( \sum_{i=1}^{K} B_i^T \vec{T}_j \vec{T}_j^T B_i \right)^{-1} \sum_{i=1}^{K} B_i^T \vec{T}_j \vec{T}_j^T \vec{a}_i$$

In each step of the EM algorithm ownership probabilities are calculated as

$$s_{ij} = \pi_j p(\vec{u}_i | \vec{T}_j, \vec{\Omega}_j, \sigma_j) / p(\vec{u}_i | \vec{T}_1, \vec{\Omega}_1, \sigma_1 \ldots \vec{T}_M, \vec{\Omega}_M, \sigma_M)$$

and updated parameters for $(\vec{T}_j, \vec{\Omega}_j)$ are generated by using a Newton-Rhapson algorithm to minimize

$$f(\vec{T}_j, \vec{\Omega}_j) = \sum_{i=1}^{K} s_{ij} \left[ \vec{T}_j^T \left( \vec{a}_i + B_i \vec{\Omega}_j \right) \right]^2$$

subject to the constraint $||\vec{T}_j|| = 1$ and holding the $s_{ij}$'s fixed. Variances are estimated as

$$\sigma_j^2 = \sum_{i=1}^{K} s_{ij} \left[ \vec{T}_j^T \left( \vec{a}_i + B_i \vec{\Omega}_j \right) \right]^2 / \sum_{i=1}^{K} s_{ij}$$

The EM-algorithm is allowed to run until the parameters converge. This provides us with improved estimates for $(\vec{T}_j, \vec{\Omega}_j)$ as well as clustering the constraints to processes. Since each constraint is tied to an image location, this gives us a segmentation of the image based on underlying 3-D motion. As with the linear constraints, we check the uniqueness by comparing $\vec{T}$ parameters using Eqn. 5.

# 5 Results

Figure 1 shows a frame from a sequence taken by a robot navigating in an industrial environment. The forklift and driver are translating to the right at roughly 50 pixels/frame, while the robot is moving forward.[4] Optic flow was recovered from

[4] The robot's speed was not measured, but was the equivalent of a fast walk.

the sequence using a method that fits flow in image regions (patches) to functions that are either affine or rational in image coordinates $\vec{x}$ [11]. The $\vec{\tau}$ were recovered by considering patches in a pair-wise manner[5]: 6 flow samples were generated for each patch, using the 4 corners of each patch plus two interior points. The constraints were clustered according to the method in Section 4.1 and gave estimates for two translational directions (see Table 1). Bilinear constraints were then generated for each sample point and clustered according to Section 4.2. Figure 1 plots the FOE values recovered for the first motion for each frame in the sequence. The results are summarized in Table 1. Figure 2a suggests that motion process 2 belongs to the patch fitting flow for the moving forklift, and that motion process 1 owns the remainder of the constraints.

It is necessary to check that the recovered translational directions are unique: for each process we generate a set of translation constraints from the bilinear constraints (given $\vec{\Omega}$) and use this to generate a $D$ matrix as described in Eqn. 4. We then test each $\vec{T}$ against each $D$ by Eqn. 5. In Table 1 we see that $p(\vec{T}_2|D_1)$ and $p(\vec{T}_1|D_2)$ are zero, indicating that $\vec{T}_1$ and $\vec{T}_2$ are indeed distinct. Therefore we have segmented the moving forklift in the image.

Once $\vec{T}$ and $\vec{\Omega}$ are known for the egomotion (the first motion), it is possible to estimate relative-depth values for each sampled point from Eqn. 1. The estimates for the centre of each patch are shown in Figure 2b. We see that the relative depths make sense, in that closer objects (the floor, pillar, and stationary forklift) have larger inverse-depths than do the farther objects (the back wall, and the mockup windows). The moving forklift (not shown) gives negative depth values when considering the egomotion parameters: this provides another method of detecting that it is moving independently.

The bilinear constraint clustering ran in about 7 seconds/frame on a Silicon Graphics 4D/340VGX, with comparable times for the linear constraint clustering.

## 6 Conclusions

We have described a method of segmenting images containing both egomotion and independent object motion based on 3-D motion constraints. Results are given from an image sequence taken in an industrial environment.

The authors would like to acknowledge the assistance of David Wilkes in acquiring the image sequence analyzed in this paper. This work was supported by ITRC, NSERC, and OGS.

## References

[1] Adiv G. **Determining three-dimensional motion and structure from optical flow generated by several moving objects**. *IEEE Trans Pattern Analysis & Machine Intelligence*, PAMI-7(4):384–401 1985

[5] This is necessary since the flow from a single patch is described by a first-order polynomial in $\vec{x}$ and as such will not generate a constraint. The floor patch was paired once with each other patch.

[2] Darell T, Pentland A. **Robust estimation of a multi-layered motion representation**. *Proc of the IEEE Workshop on Visual Motion*, October 7–9 1991. Princeton, New Jersey 173-177

[3] Dempster AP, Laird NM, Rubin DB. **Maximum likelihood from incomplete data via the EM algorithm**. *Journal of the Royal Statistical Society B*, 39:1–38

[4] Gibson JJ. **The perception of the visual world**. Houghton Mifflin, Boston, Ma. 1950

[5] Helmholtz H. **Treatise on physiological optics**. Dover, New York 1910

[6] Horn BKP. **Robot vision**. The MIT Press, Cambridge, Massachusetts 1986

[7] Jepson AD, Heeger, DJ. **A fast subspace algorithm for recovering rigid motion**. *Proc. of IEEE Workshop on Visual Motion*, Princeton, NJ (Oct. 1991), 124–131 1991

[8] Jepson AD, Heeger, DJ. **Linear subspace methods for recovering translational direction**, in **Spatial Vision in Humans and Robots**, Eds. L. Harris and M. Jenkin, Cambridge Univ. Press 1993

[9] Jepson AD, Heeger, DJ. **Subspace methods for recovering rigid motion, II: Theory.** In preparation.

[10] Jepson A, Black MJ. **Mixture models for optical flow computation**. *Proceedings 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* June 15–18 1993, New York City, New York 760–761

[11] Jenkin M, Jepson A. **Detecting Floor Anomalies**. *Proceedings of the British Machine Vision Conference*, 1994

[12] McLachlan GJ, Basford KE. **Mixture models: Inference and applications to clustering**. Marcel Dekker Inc, New York 1988

[13] Neal RM, Hinton GE. **A new view of the EM algorithm that justifies incremental and other variants**. Submitted to Biometrika, 1993

[14] Nelson RC. **Qualitative detection of motion by a moving observer**. *International Journal of Computer Vision* 7(1):33–46 1991

[15] Rieger JH, Lawton DT. **Processing differential image motion**. *J Opt Soc Am A* 2(2):354–359 1985

[16] Sinclair D. **Motion segmentation and local structure**. *Proceedings of the 4th International Conference on Computer Vision*, Berlin, May 1993 366–373

[17] Wang JYA, Adelson EH. **Layered representation for motion analysis**. *Proceedings 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* June 15–18 1993, New York City, New York 361–366