

# 3-D Motion Estimation Using Linear and Bilinear Constraints

W. James MacLean      Allan D. Jepson

Oct 12, 2000

## 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. Randomly sampling from the flow field is not a good method when a large planar surface is present in the image, due to the nature of the linear constraints. A method for choosing flow samples in this case is presented. Results from three image sequences are presented.

## 1 Introduction

A key function of many biological vision systems is the ability to interpret time-varying images. Images are time varying for a variety of reasons, including the case where image motion is induced by an observer moving in a stationary environment, and also by the presence of other objects moving relative to the environment. For the purposes of navigation in a general environment it is of great importance to be able to interpret image motion arising from one of these causes.

Image motion induced by an observer's movement in a stationary environment is often referred to as *egomotion* [10]. For the purposes of this paper, other objects moving in the same environment are termed *independently moving objects* (IMOs).

Through the recovery of egomotion one can recover a *relative depth map*<sup>1</sup> of the environment. This assumes that either there are no IMO's present, or if there are, their image locations are known, and not included in the depth recovery process.

Ideally, one would like to be able to perform a segmentation of the image in which regions containing IMO's are identified, even if the motion parameters of the IMO cannot be recovered. A considerable amount of work has been done previously on this problem. Most of it has involved 2-D segmentation, which involves finding discontinuities in optic flow images. Darell & Pentland [3] propose a method for segmenting flow into layers based on the 3-D translation underlying the flow. Their work involves clustering 2-D constraints using an iterative and competitive algorithm. Jepson & Black [13] propose using a mixture model method for clustering component velocities in order to improve flow estimates in regions containing discontinuities in the flow field. Wang & Adelson [28] segmented image regions into patches, where each patch is modelled by an affine flow field. They used K-means clustering. All of these approaches are attempts to improve the integration of constraints during the estimation of optical flow.

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 [25] 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 [24] 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. The use of the EM algorithm for 3D motion segmentation was introduced by MacLean *et. al.* [22]. Feng and Perona [6] have suggested the use of the EM algorithm to segment 3D motion using the essential matrix approach, but they do not seem to indicate a method for determining the number of underlying motions.

In this paper the authors propose a methodology for 3-D image segmentation

---

<sup>1</sup>In a relative depth map, the ratio of the depths of two objects may be recovered, but not their absolute depths.

based on optical flow as an estimate of the motion field. This method attempts to recover egomotion parameters, as well as identify regions containing IMOs. The method is based on linear and bilinear constraints on 3-D motion which are simply derived from flow field values. These constraints are not sensitive to depth discontinuities in the image sequence, except to the extent that the linear constraints actually require a significant level of depth variation. Multiple motion processes are described by a finite mixture model, and the constraints are fit to the model using the EM algorithm.

An overview of the paper is as follows. Section 2 describes the theory which defines what is meant by the “motion field” and the methods for deriving linear and bilinear subspace constraints from it. Calculation of motion parameters from the constraints and known problems in using the constraints are examined. Finally, the form of the mixture model is described, and details of the application of the EM algorithm to the simultaneous problems of segmentation and parameter estimation are detailed. Section 3 presents the results of applying the method to three image sequences. The first sequence involves synthetic flow derived from a depth map (Z-buffer) of a computer animated scene. Noise is added to the flow to make it more realistic. This case allows examination of clustering linear constraints in the case where the motion field is known. The second sequence involves a robot observer moving in an industrial environment, in the presence of an IMO. Flow estimates are based on fitting affine and rational displacement models to the tracking of localised image texture. Clustering of both linear and bilinear constraints is used to demonstrate the potential of the method. The third sequence is the Otte sequence, for which ground-truth estimates of the motion are known. Again, clustering of both linear and bilinear constraints is used to estimate egomotion and attempt to identify the IMO. This sequence contains a dominant planar region which required special care in sampling the flow vectors for constraint generation. Section 4 discusses these results, and attempts to identify issues which need to be addressed in order to further improve the method. Section 5 summarizes the results in this paper, and suggests directions for future research.

## 2 Theory

Throughout this paper a right-hand coordinate system is used, and a pin-hole camera model with a planar receptor is adopted. The origin is coincident with the camera’s nodal point, meaning that all rays pass through the origin. The  $z$ -axis inter-

sects the image plane at right angles at a distance  $f$  from the origin, and defines the optic axis of the system. The receptor is placed in front of the origin in order to avoid the need to reflect coordinates. A point  $\vec{X}$  in the 3-D world is imaged to a point  $\vec{x}$  which lies in the image plane. Under perspective projection we write  $\vec{x} = \frac{f}{X_3} \vec{X}$  where  $\vec{X} = [ X_1 \ X_2 \ X_3 ]^T$ .

We are interested in the case where points in the 3-D world are in motion. A point  $\vec{X}$  has a motion described by

$$\vec{V}(t) = \frac{d\vec{X}}{dt} = \vec{T}(t) + \vec{\Omega}(t) \times \vec{X} .$$

Here  $\vec{T}$  represents the translational component of the point's motion with respect to the observer, and  $\vec{\Omega}$  is the rotational component. For the rest of the paper we will drop the reference to  $t$  for velocities.

The motion of  $x$  in the image plane is given by

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

Equation 1 is the ‘‘motion-field’’ equation. It represents an idealized form of optic flow. In reality we cannot measure the motion field directly, and use a variety of optic flow techniques to estimate it.

## 2.1 Constraints on 3-D Motion Using the Motion Field

Having estimated the motion field, attempts at recovery of 3-D motion are complicated by the appearance of  $X_3$  (the depth of point  $\vec{X}$  along the optical axis) in Eq. 1. In this paper we utilize a technique called ‘‘subspace methods’’ developed by Jepson and Heeger [14]. This technique allows constraints on 3-D motion to be calculated from the motion field estimates. These constraints come in two varieties: bilinear and linear.

### 2.1.1 Bilinear Constraints

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

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

This is an exact constraint on 3-D motion given the motion field. It has the special form that if  $\vec{T}$  is held constant then the equation is linear in  $\vec{\Omega}$  and *vice versa*. Hence the name ‘bilinear’. Only a single flow vector and its associated image location is required for each constraint. Since Eq. 2 is homogenous in  $\vec{T}$ , we can only recover  $\vec{T}$  up to a scale factor; *i.e.* we can only recover translation direction, and not magnitude. The bilinear constraint may be written more compactly as  $\vec{T}^T(\vec{a}(\vec{u}, \vec{x}) + B(\vec{x})\vec{\Omega}) = 0$  where  $\vec{a}$  is a  $3 \times 1$  vector valued function of image location and motion field, and  $B$  is a  $3 \times 3$  matrix valued function of image location alone. It should be noted that  $B$  is quadratic in  $\vec{x}$ . Soatto *et. al.* [26] appear<sup>2</sup> to derive and use the bilinear constraint for motion and structure estimation while using a spherical receptor surface. In a review of egomotion computation techniques, Tian *et. al.* [27] point out that this constraint was first derived by Bruss & Horn [2] in 1983.

### 2.1.2 Linear Constraints

While the bilinear constraints are easily derived, their solution requires non-linear techniques. It is possible to devise a new constraint which factors out the influence of  $\vec{\Omega}$  and is linear in  $\vec{T}$ .

Given motion field estimates at  $K \geq 7$  distinct points in the image,  $\{\vec{x}_k\}_{k=1}^K$ , construct a constraint vector  $\vec{\tau}$  according to

$$w\vec{\tau} = \sum_{k=1}^K c_k [\vec{u}(\vec{x}_k) \times \vec{x}_k] . \quad (3)$$

Here  $\vec{\tau}$  is a unit vector and  $w$  is the norm of the right-hand side of the equation. If the  $c_k$  are chosen to be orthogonal to all quadratic forms involving  $\vec{x}_k$  over the sample points, then  $\vec{\tau}$  is guaranteed to be orthogonal to  $\vec{T}$ . For  $K$  sampled points, it is possible to create  $K - 6$  linearly independent constraints.<sup>3</sup> These constraints form a basis for a subspace of  $\mathcal{R}^{\mathcal{X}}$ , hence the name subspace methods. Note that Eq. 3 represents a weighted sum of the bilinear constraints at each of the sampled image locations, where the contributions from  $B(\vec{x}_k)$  is 0.

---

<sup>2</sup>In their paper they refer to a “measurement vector”  $\vec{y}_i$  which needs to be equal to  $\vec{x} \times \vec{u}$  in order to match the bilinear constraint described in this paper.

<sup>3</sup>We require 7 sampled points to create one constraint, and 8 to create 2 constraints. Since we require at least 2 constraints to solve for  $\vec{T}$ , this is equivalent to the 8 point-correspondences required using the essential matrix method.

As a result of the  $c_k$ 's being orthogonal to all quadratic forms involving  $\vec{x}_k$ 's, it should be noted that  $\vec{\tau}$  will be identically zero if all the  $\vec{x}_k$ 's are sampled from a single planar surface imaged onto the receptor.

### 2.1.3 Solving for $\vec{T}$ and $\vec{\Omega}$

Given a set of linear constraints  $\vec{\tau}_i$ , how can we solve for  $\vec{T}$ ? Recall that we can only solve for the direction of  $\vec{T}$ , so we can arbitrarily set its magnitude to unity. This allows us to interpret the  $\vec{\tau}_i$  as lying on a great circle on the surface of the unit sphere, orthogonal to the direction of  $\vec{T}$ . Given a weighted set of linear constraints,  $\{w_i \vec{\tau}_i\}_{i=1}^N$ , the translational direction is given by the eigenvector corresponding to the smallest eigenvalue of

$$D = \sum_{i=1}^N w_i^2 \vec{\tau}_i \vec{\tau}_i^T . \quad (4)$$

This is the equivalent of minimizing the expression

$$E(\vec{T}) = \vec{T}^T D \vec{T} = \sum_{i=1}^N w_i^2 (\vec{T}^T \vec{\tau}_i)^2 .$$

This is a simple linear least-squares calculation.

Solution of the bilinear constraints is more difficult. Since these constraints are non-linear, closed-form analytic solutions are generally not available, and we must resort to iterative optimization algorithms. These typically require an initial guess for the parameters, and may converge slowly.

### 2.1.4 Known Problems

Given the difficulty of solving for 3-D motion using the bilinear constraints, why not just use the linear constraints exclusively? The answer lies in the fact that the linear constraints suffer from two main problems.

The linear constraints are exact in the absence of noise. When isotropic noise is added to the motion field, a non-isotropic noise distribution arises in the linear constraints. This results in a bias towards the optic axis when translational direction is estimated. Several methods for rescaling constraints have been proposed to solve this problem [20, 21, 18, 19].

A second problem arises from an assumption implicit in the formation of the linear constraints, namely that all of the flow samples used to create each constraint

come from one or more objects moving with a single rigid motion. This is satisfied when the image only involves the observer’s motion in an otherwise static environment. This is an unrealistic assumption in the general case, since we must allow for the possibility of object’s whose motion is different from the environment. We term these *independently moving objects* (IMOs). It can be shown that when a linear constraint is formed from flow samples taken from the background and an IMO, the resulting constraint lies somewhere in the plane defined by the constraints that would have resulted from either motion alone. While the bilinear constraints are more difficult to solve, they do not suffer from the two defects described above.

It should also be noted that in the event that the observer is undergoing pure rotation, then the linear constraints will be zero (*i.e.*  $w_i = 0$  for all  $i$ ). In this case it is easy to recover rotation.

## 2.2 Mixture Models

Attempting to recover 3-D motion in the (possible) presence of IMOs poses an interesting problem. Each estimated motion field vector may arise due to the relative 3-D motion between the observer and the stationary environment, or the relative 3-D motion between the observer and one or more IMOs. In the case where a data set contains observations, each of which arises from one of several underlying processes, the concept of a mixture model may be useful [23, 11].

In this case, each process has its own distribution and parameters. We wish to find i) which data points belong to which processes, and ii) the values of the parameters for each process. The first task is referred to as *clustering*, while the second is *parameter estimation*. It is usually assumed that the number of processes is known in advance, as the problem of testing for this number is in general difficult and unsolved. Mixture models may also be used for robust estimation. We have two distributions: the one we would like to fit to the data, and a second distribution to capture “outliers” in the data.

In the recovery of 3-D motion in image sequences, the linear and bilinear constraints will be our observed data. There will be one process to model egomotion, one process for each IMO, and one process to model outliers. Since we will not know in advance the number of IMOs, we will need to determine the number of IMO processes from the data (see Section 2.3.1). It remains to define distributions for the processes.

In the case of the linear constraints, the likelihood function (LF) involves the expected translational direction  $\vec{T}$  for each process, as well as a measure of cer-

tainty in the estimate. This certainty is modeled by  $\sigma$ , a measure of the spread of the likelihood function. The likelihood of a linear constraint  $\vec{\tau}_i$  with respect to a number of underlying translations  $\{\vec{T}_j\}_{j=1}^M$  is given by

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

$$\sum_{j=0}^M \pi_j = 1, \tag{5}$$

$$0 \leq \pi_j \leq 1$$

where  $M$  is the number of processes. The  $\{\sigma_j\}_{j=1}^M$  are dependent on the error in the motion field estimates. Note that estimating flow for an IMO, especially if it is small, may have different uncertainty than estimating flow for the stationary background, so we allow each motion process its own value. The  $\pi_j$  are called *mixture proportions* and are positive valued constants. The parameters  $\pi_j$ ,  $\sigma_j$  and  $\vec{T}_j$  are all unknowns, and will be estimated given the linear constraints.

The outlier distribution is modeled by the constant  $p_0$ . Since The domain of the  $\vec{\tau}_i$  is the unit sphere, we set  $p_0 = (4/3\pi)^{-1}$  (the reciprocal of the surface area of the unit sphere). (*cf.* [13])

The form of the LFs for the motion processes is taken to be a Gaussian modified for the unit sphere:

$$p(\vec{\tau}_i|\vec{T}_j, \sigma_j) = \frac{3}{4\pi(2 + \exp\{-1/2\sigma_j^2\})} \exp\left\{-\frac{(\vec{\tau}_i^T \vec{T}_j)^2}{2\sigma_j^2}\right\} \tag{6}$$

The form of this distribution requires some explanation. In a noiseless case, we expect all  $\vec{\tau}$ 's associated with a given  $\vec{T}$  to lie along a great circle on the unit sphere, where the plane defined by this great circle is orthogonal to  $\vec{T}$ . However, once noise is added, we expect that the linear constraints will not lie exactly in the plane, but will be close to it. This distribution describes a Gaussian ridge lying along a great circle (see Fig. 1).

Once the motion parameters are known or estimated, each linear constraint  $i$  can be assigned an ownership probability with respect to each motion process  $j$  as follows:

$$s_{ij} = \frac{\pi_j p(\vec{\tau}_i|\vec{T}_j, \sigma_j)}{p(\vec{\tau}_i|\vec{T}_1, \sigma_1 \dots \vec{T}_M, \sigma_M)}. \tag{7}$$

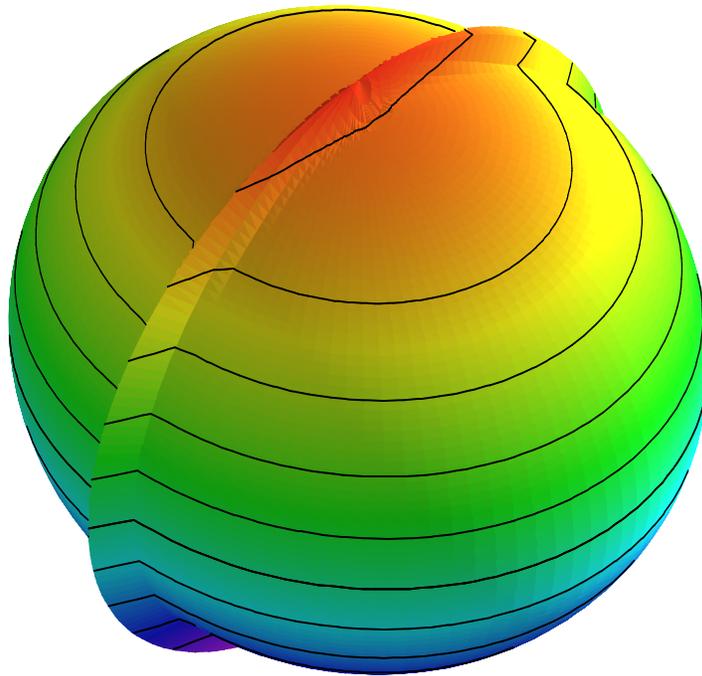


Figure 1: The likelihood function for constraint  $\vec{\tau}_i$  arising from translational motion  $\vec{T}_j$  is shown in this plot. The function forms a Gaussian ridge along a great circle on the unit sphere, where the spread of the ridge is controlled by  $\sigma_j$ .

The case for the bilinear constraints is very similar. All that changes is the form of the LF.

$$p(\vec{u}_i | \vec{T}_j, \vec{\Omega}_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp \left\{ -\frac{(\vec{T}_j^T (\vec{a}_i(\vec{u}_i) + B_i \vec{\Omega}_i))^2}{2\sigma_j^2} \right\} \quad (8)$$

Here, since each bilinear constraint is synonymous with a single flow vector, we have written the LF using  $\vec{u}$  as the observed data, and with image location and motion parameters as the independent parameters. This LF can be interpreted as a Gaussian whose parameter is the distance from a line defined in  $\vec{u}$  space. Also, since the domain of this LF is not compact we cannot use a constant value for the outlier process LF: a large-variance Gaussian is used instead.

The question which remains is “how can we simultaneously estimate motion parameters and ownership probabilities?”

### 2.3 The EM Algorithm

As stated in the previous section, mixture models are useful representations for the recovery of 3-D motion from image sequences. The linear and bilinear constraints, derived from motion field estimates, are the observed data. We wish to robustly estimate motion parameters for each motion process, and also determine which data (constraints) belong to each process. If we knew *a priori* which constraints belonged to which motion processes, the problem of parameter estimation is straightforward. This would be the case if we had identified and segmented IMO in advance. One the other hand, if we knew the parameters of the different motion processes, segmenting the constraints would be easy. This could correspond to the case where information from another motion-sensing modality is available. In general we cannot assume that we know either the segmentation or the motion parameters, so we must estimate both.

One method for doing this is to use the *EM algorithm* [4]. This is an iterative, two-step method use for simultaneous estimation of segmentation and process parameters in mixture models. We assume that we start with an initial guess for the motion parameters. The first step, named the “expectation step”, involves assuming that the motion parameters are indeed correct, and estimating ownership probabilities from them. The second step, called the “maximization step”, involves assuming that the calculated ownership probabilities are correct and re-estimating the motion parameters based on them. Each pair of “E” and “M” steps constitutes one iteration of the algorithm.

Equation 5 can be viewed as a likelihood function, *i.e.* it indicates the likelihood that the observed data came from the mixture model parametrized by the motion parameters and the mixing proportions. The EM algorithm is guaranteed to increase, or at worst leave unchanged, this likelihood function. Therefore the EM algorithm is a form of maximum likelihood estimation. In the case where the likelihood is unchanged, a maximum has typically been encountered.

The likelihood function is typically non-linear, and as such will have multiple maxima. There is no guarantee that the EM algorithm will ever find the global maximum. In fact, which maximum it finds is likely to be related to the initial guess provided, so it is of great importance that a good initial guess be used. There are also no general results for the rate of convergence of the algorithm.

### 2.3.1 Clustering Linear Constraints

The first step in recovering 3-D motion involves using the EM algorithm to cluster linear constraints recovered from the image sequence. Clustering linear constraints is simpler than clustering bilinear constraints since we can use them to provide an initial guess for  $\vec{T}$ . Once  $\vec{T}$  is recovered, it is simple to recover an estimate of  $\vec{\Omega}$ . Clustering linear constraints also provides a means for determining the number of motion processes involved in the mixture.

The method begins by considering the matrix  $D$  defined in Eq. 4. As stated before, the eigenvector corresponding to the minimum eigenvalue (hereafter referred to as the “minimum eigenvector”) provides an estimate for translational direction. This is premised on the idea that  $D$  will have one eigenvalue which is much smaller than the rest. It is possible that  $D$  will have two eigenvalues which are much smaller than the third. This can be expected in the case where the  $\vec{\tau}_i$  lie close to one another in a cluster on a great circle. In this case there may be two possible directions for  $\vec{T}$ . Let  $\lambda_1, \lambda_2$  and  $\lambda_3$  be the eigenvalues of  $D$  in decreasing order. In general there are three cases:

$\lambda_1 > \lambda_2 \gg \lambda_3$ : This indicates the possibility of one translational direction, *i.e.* the  $\vec{\tau}_i$  lie along a great circle. More specifically, the constraints lie in a cluster whose major axis is significantly larger than the minor axis. The major axis lies along the great circle (see Fig. 1).

$\lambda_1 \gg \lambda_2 \approx \lambda_3$ : This case occurs when all the constraints lie close together and are distributed in a roughly circular manner. This supports the possibility

of two translation directions. Instead of a ridge along the unit sphere, we observe a “bump”.

$\lambda_1 \approx \lambda_2 \approx \lambda_3$ : This is a degenerate case. It means that the constraints are distributed roughly equally in all directions. There may or may not be multiple unique underlying translations, but we have no indication of any preferred direction.

Assuming that  $D$  conforms to one of the first two cases, we can start the EM algorithm with one or two guesses for  $\vec{T}$ . In order to distinguish between the first two cases, we compare  $\lambda_2$  to the geometric mean of the other eigenvalues,  $\sqrt{\lambda_1 \lambda_3}$ . When  $\lambda_2$  is greater than this mean, we use only the minimum eigenvector of  $D$ . If  $\lambda_2$  is less than the mean, we also use the eigenvector associated with  $\lambda_2$ .

The EM algorithm is run with the initial guesses for the  $\vec{T}_j$ . During each E-step we calculated the ownership probabilities of each  $\vec{\tau}_i$  with respect to each motion process using Eq. 7. The mixing proportions for each process are also computed at this time. During each M-step a matrix  $\hat{D}_j$  is computed for each process as follows:

$$\hat{D}_j = \sum_{i=1}^N s_{ij} w_i^2 \vec{\tau}_i \vec{\tau}_i^T / \sum_{i=1}^N s_{ij} w_i^2 \quad (9)$$

The new value of  $\vec{T}_j$  is the minimum eigenvector of  $\hat{D}_j$ . This matrix is really just  $D$  where each  $\vec{\tau}_i$  is weighted by its ownership probability with respect to the  $j$ th process. The variance for the  $j$ th process is estimated by

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

The EM algorithm is continued until the parameters converge. Once the algorithm has stopped, the eigenvalues of  $\hat{D}_0$  (the outlier process) are examined. If there is evidence for new translational directions based on the criteria described above, then one or two new processes are added to the mixture, and the EM algorithm is repeated. This method is a form of “splitting” where new processes are split from the outlier population until either 1) the mixture proportion of the outlier population becomes too small (indicating that it “owns” very few constraints), or 2) the new processes cease to be unique as compared to previously existing processes. The latter case may be determined by comparing the translational directions  $\vec{T}_l$  of

the other processes to  $\hat{D}_j$  as follows:

$$p(\vec{T}_l|\hat{D}_j) = \frac{1}{k} \exp \left\{ -\vec{T}_l^T \hat{D}_j \vec{T}_l \right\}, k = \frac{4}{3} \pi (e^{-\lambda_1} + e^{-\lambda_2} + e^{-\lambda_3}) \quad (10)$$

Once the number of processes and their parameters have been determined, we can proceed to clustering of the bilinear constraints.

### 2.3.2 Clustering Bilinear Constraints

The previous section described the clustering of linear constraints. This clustering estimated the number of underlying motion processes, as well as parameters for each process. As discussed in Section 2.1.4 linear constraints have several problems, not the least of which is that they become outliers if the flow samples used to compute them are taken from more than one motion process.

To refine our estimates of 3-D translation, as well as estimating rotation, we continue by clustering the bilinear constraints. For each motion process we need to generate an initial guess for  $\vec{\Omega}_j$ . This can be done in a least-squares sense using the equation

$$\vec{\Omega}_j = \left( \sum_{i=1}^K B_i^T \vec{T}_j \vec{T}_j^T B_i \right)^{-1} \sum_{i=1}^K B_i^T \vec{T}_j \vec{T}_j^T \vec{a}_i. \quad (11)$$

Note the lack of ownership weights. They are not included since the ownership weight for any given linear constraint may not be a true reflection of the ownership of the bilinear constraints used to form it (for example, when the linear constraint is an outlier, the bilinear constraints associated with it need not also be outliers).

Starting with an initial guess for the motion parameters,  $\left\{ \vec{T}_j, \vec{\Omega}_j, \sigma_j \right\}_{j=1}^M$ , the EM algorithm is applied to the bilinear constraints. During each E-step we estimate ownership probabilities using

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

and from these estimate the mixing proportions. During each M-step a Newton-Rhaphson algorithm is used to minimize the cost function

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

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

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

When estimating variances, upper and lower bounds are set on the values of  $\sigma_j$ , based on expected error ranges in the flow data. If no bounds were placed, then one can conceive of extreme cases in which the  $\sigma$  values inflate until all constraints are “matched” by a single process, or they deflate until each each process matches only a single constraint. The latter case is less of a problem when the number of processes is fixed.

Running the EM algorithm on the bilinear constraints leads to improved estimates for the motion parameters. Since each constraint represents exactly one image location, they also provide a 3-D motion-based segmentation of the image.

## 2.4 Using Subspace Constraints to Detect IMOs and Recover Egomotion

In this section we propose a methodology for detecting IMOs and recovering the observer’s motion (egomotion) using linear and bilinear constraints together with the EM algorithm.

The EM algorithm is an iterative, 2-step algorithm for estimating model parameters in the case where each observed data may come from one of several underlying processes. In the case at hand, the observed data are the linear and bilinear constraints, and the processes are the observed motion due to the observer and each IMO. We would like to determine the parameters of motion and assign each constraint to a motion process (this is termed *ownership*). The proposed method is as follows:

First, assume all objects move with a single rigid motion, and estimate a translation direction using outlier rejection. This gives an estimate of the observer’s translational direction, as well as an indication of a population of constraints which do not fit this motion well. By examining the outlier population, it is possible to determine if evidence for another translational motion exists. If there is evidence, an estimate for this translation is introduced as a new process, and the EM algorithm is re-run on the data with the new processes in place. This process continues until either 1) there is no further evidence for new translational processes, or 2) the

new processes cease to have unique translational directions.

At this point we have an estimate for how many distinct motion processes exist as well as their translational directions. We now can apply the same methodology to clustering of the bilinear constraints.

## 2.5 When Subspace Methods Fail

There are cases where subspace methods will be useless for detection of IMOs. The most important of these occurs when an IMO’s flow field direction is locally very similar to that generated by egomotion. In this case the linear and bilinear constraints cannot distinguish between a depth discontinuity and an IMO. A simple example of this is seen when an observer riding on a train observes a car travelling on a road parallel to the train’s direction. In this case both the egomotion flow and the car’s flow are purely horizontal. Constraints derived from the car’s flow will be perfectly consistent with the egomotion process, and the IMO is undetectable using linear and bilinear constraint segmentation. A method has been devised to use evolution of depth structure to detect the IMO in this case [20, 21].

## 3 Results

In this section results from three image sequences are shown.

### 3.1 Cube Sequence

The first sequence presented involves synthetic flow generated from a pair of depth-map images,  $128 \times 128$  pixels in size. Each image location  $\vec{x}$  in the depth map contains a depth value  $X_3(\vec{x})$ , and can therefore be used to generate a flow motion field vector using Eq. 1, given a value for  $\vec{T}$  and  $\vec{\Omega}$ . In order to generate a motion field containing an IMO, two depth maps are used (see Figure 2).

The translation of the office relative to the observer is  $[1 \ 0 \ 1]^T$ . The translation of the cube relative to the observer is  $[0 \ 1 \ 0]^T$ . A rotation has been added to simulate the observer “fixating” a point near the centre of the image. This was done to improve the SNR of the linear constraints [20]. The angular extent is taken to be  $45^\circ$ . In order to make the flow more realistic, 10% random noise is added as  $\hat{u}(\vec{x}) = \vec{u}(\vec{x}) + \vec{n}$ . The noise component  $\vec{n}$  is chosen from a 2-D isotropic

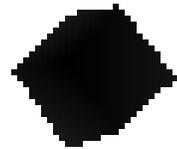


Figure 2: Two depth-map (Z-buffer) images are shown. On the left is the background image, representing an office scene. On the right is a small cube whose depth values are less (closer) than the elements of the office scene.

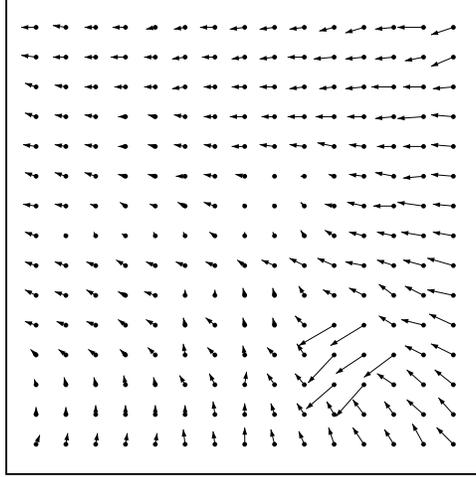


Figure 3: Resultant flow created from the cube and office depth map images (Figure 2). The cube, in the lower-right corner of the flow field, is falling.

normal distribution having a standard deviation equal to  $0.1\|\vec{u}(\vec{x})\|$ . The resulting flow field is shown in Figure 3.

For this flow field we only generate and cluster linear constraints. A dense set of  $\vec{\tau}$  are generated according to the convolution method of Jepson & Heeger [15]. A  $15 \times 15$  convolution mask is created by modifying a *difference-of-Gaussians* (DOG) function in order to satisfy  $F\vec{c} = 0$  where  $F$  is a matrix constructed from the image sampling locations of the convolution mask. This method has the advantage that the coefficients can be made independent of the absolute image position, since the null space of  $F$  is invariant under affine transformations of the image coordinates. The constraint vector  $\vec{c}$  can therefore be reused. The DOG has a centre standard deviation of 1.5 pixels, and a surround standard deviation of 3 pixels.

As each  $\vec{\tau}$  is generated its SNR is checked, and constraints with SNR less than 5 are discarded. The following definition is used for SNR:

$$\text{SNR} = \frac{\|\vec{\tau}\|}{\rho\sigma_u}$$

where

$$\sigma_u^2 = \sum_{k=1}^K c_k^2 \|\vec{u}(\vec{x}_k)\|^2$$

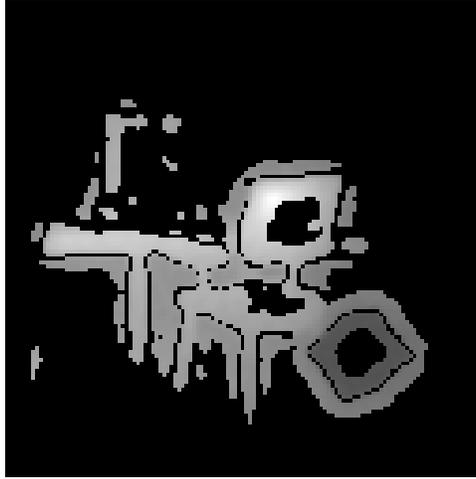


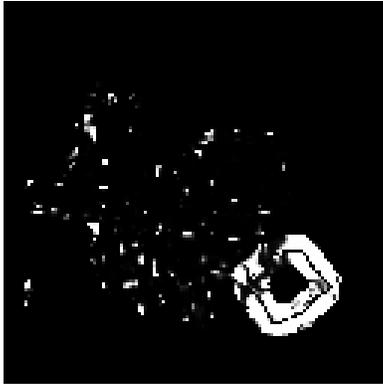
Figure 4: This figure shows the image location and relative magnitudes of the recovered linear constraints. Note that constraints are only recovered in locations of significant depth variation and constraint magnitude reaches a maximum at the point of fixation.

is a weighted average of the magnitudes of the flow vectors used to construct the constraint. The value of  $\rho$  is set to 0.1, the same as the noise magnitude coefficient. Figure 4 shows the relative magnitude and image location of the recovered constraints. Note that the constraint magnitude hits a maximum at the point of fixation. Approximately 3,000 constraints were recovered.

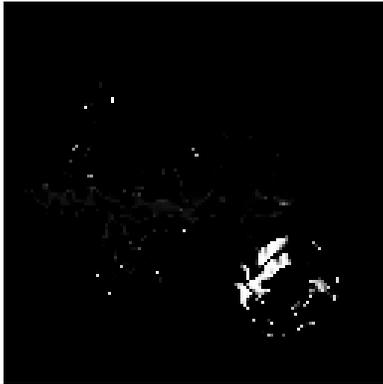
### 3.2 Results of Clustering Constraints

The results of clustering linear constraints according to the “great-circle” distribution of Eq. 6 are shown in Figure 5. There are 4 processes recovered including the outlier population. Two of these processes are merged at the end on account of being deemed too similar. The recovered processes form a segmentation of the flow into a background process and the IMO.

Table 1 shows numerical results for the clustering. The outlier process has a mixing proportion of 0.225157 corresponding to ownership of about 22.5% of the constraints. Process 2 is seen to represent the background, and claims about 71.5% of the constraints. Its recovered translational direction is  $[ 0.5654 \ 0.0013 \ 0.8248 ]^T$ , which differs from  $[ 1 \ 0 \ 1 ]^T$  by about  $10.5^\circ$  error. This deviation is due to the



Process 0 (outliers)



Process 1



Process 2

Figure 5: This figure shows the segmentation of the  $\vec{\tau}$ 's based on translational direction. Process 2 clearly belongs to the background motion, whereas Process 1 has garnered support from some of the constraints generated at the cube-background boundary. The outlier population also has considerable support from these constraints, as well as some from the background.

	Process 0 (outliers)	Process 1	Process 2	Process 3
Mixtures	0.225157	0.009908	0.715241	0.049695
$\vec{T}$		$\begin{bmatrix} -0.3510 \\ -0.0033 \\ -0.9364 \end{bmatrix}$	$\begin{bmatrix} 0.5654 \\ 0.0013 \\ 0.8248 \end{bmatrix}$	$\begin{bmatrix} -0.2146 \\ -0.0222 \\ -0.9764 \end{bmatrix}$
$\sigma$		0.0041	0.0632	0.0084

Table 1: This table shows the estimated parameters recovered by clustering the linear constraints. Process 2 represents the background constraints. Process 1 and 3 were merged, and represent the cube- background outliers.

bias discussed in Section 2.1.4, and may be reduced by rescaling the estimates according to the expected noise distribution of the constraints [20]<sup>4</sup>.

The translational estimate for the cube is based solely on outlier constraints formed across the cube-background boundary. For this reason we do not expect the estimate to be close to the correct value. In [20] a method of breaking great circles into smaller, localized clusters of constraints is presented. When this method is used, together with the bias correction algorithm discussed above, the estimate of the cube’s translational motion is found to be within  $1.9^\circ$  of the relative translation between the cube and background, as seen by the observer. Still, merely being able to identify an IMO either through its presence in the outlier population or as a separate motion process is the most important result to be found here.

### 3.3 Forklift Sequence

The forklift sequence consists of 10 frames captured in an industrial environment. Each image is  $640 \times 480$  pixels in size. The sequence was captured from a video camera mounted on a robot translating roughly along the optical axis of the camera.

<sup>4</sup>In [20, 21] it is shown that this error is reduced to about  $1.5^\circ$  or less.



Figure 6: 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 has been indicated by a 'x'.

The robot's speed was not measured, but was the equivalent of a fast walk. There is one IMO in the scene: a forklift moves from left-to-right across the robot's path, at speeds of up to 50 pixels/frame. The industrial environment is quite irregular but provides good texture for optic flow recovery, and good depth structure for the generation of  $\vec{\tau}$  constraints. Figure 6 shows a frame from the sequence.

Optic flow for this sequence was generated by clustering constraints that are consistent with either affine or rational models of flow [12]. The affine model for flow is

$$\vec{u}(\vec{x}) = \begin{bmatrix} \alpha_0 + \alpha_2 x_1 + \alpha_4 x_2 \\ \alpha_1 + \alpha_3 x_1 + \alpha_5 x_2 \end{bmatrix} \quad (12)$$

assuming that  $\vec{x} = [x_1 \ x_2 \ 1]^T$ . This model allows the flow to vary in a linear fashion over the patch. The rational model provides the position  $\vec{x}'$  of the displaced point as a function of the original position,  $\vec{x}$ .

$$\vec{x}'(\vec{x}) = \frac{1}{\alpha_6 x_1 + \alpha_7 x_2 + 1} \begin{bmatrix} \alpha_0 + \alpha_2 x_1 + \alpha_4 x_2 \\ \alpha_1 + \alpha_3 x_1 + \alpha_5 x_2 \end{bmatrix}. \quad (13)$$

The rational model adds two parameters to the affine model, and is an exact representation of the displacement of image points from a planar surface undergoing rigid motion [5]. Since the time period between frame samples is small in this sequence,<sup>5</sup> displacement can be used to provide a good model for estimating flow:

$$\vec{u}(\vec{x}) \approx \vec{x}'(\vec{x}) - \vec{x}. \quad (14)$$

The rational model is particularly useful for tracking surfaces that are known to be planar, such as the floor [12].

The constraints used in clustering were component velocities recovered by tracking of contours of constant phase [7, 8, 9]. The integration of these constraints was accomplished by the application of the EM-algorithm to solve for ownership and the parameters for each image region (patch) [12, 13].

Figure 6 shows a number of regions fit to one of the models. Each region is referred to as a ‘patch’. All of the patches in this image have been fit to the affine model, except for the floor which is fitted to the rational model. While it may seem that the patches in Figure 6 have already segmented the image, it is over-segmented with respect to independent object motion.

Each patch is sample at 6 points to provide flow estimates for generating constraints. Flow estimates are shown in Figure 7. Flow samples from two different patches are required to generate a linear constraint, since flow from a single affine patch would result in  $\vec{\tau} = 0$ . Each patch is paired with the floor patch for the generation of linear constraints. Therefore we obtain 30 linear constraints for clustering. The recovered constraints are shown in Figure 8.

The linear constraints are clustered according to the method in Section 2.3.1. Two translational motion are identified, one roughly along the optic axis, and the other in the horizontal direction. Table 2 shows the results of the linear clustering. The recovery of a good estimate for the motion of the forklift may at first

---

<sup>5</sup>The frame rate was about 15 frames per second, which is small relative to the velocity of the observer.

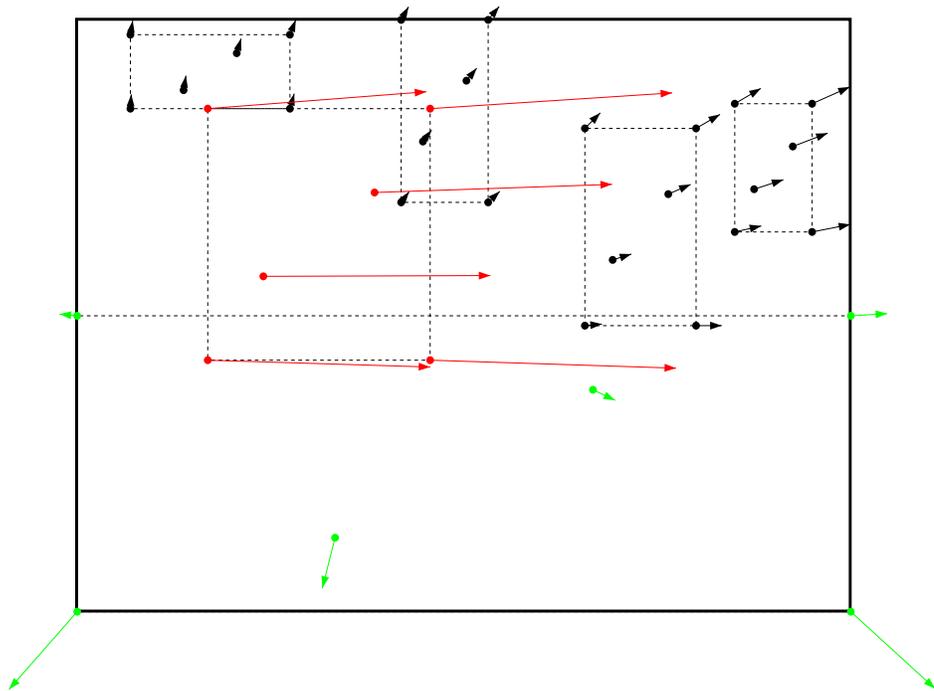


Figure 7: This figure shows the optic flow recovered from the frame shown in Figure 6. Each dashed box indicates a patch described by an affine or rational flow model. For each box six samples of the flow have been plotted, according to the model used to recover that patch. The green flow vectors are from the floor patch. The red flow vectors are from the moving forklift.

Process 1: $\vec{T} = [-0.0002 \ -0.0925 \ 0.9957]$ $f\vec{\Omega} = [0.33 \ 7.98 \ 4.69]$ $\vec{\sigma} = 3.48084$
Process 2: $\vec{T} = [-0.9948 \ 0.0216 \ 0.0996]$ $f\vec{\Omega} = [-2.94 \ -99.55 \ -6.55]$ $\vec{\sigma} = 0.64782$

Table 2: Clustering of the linear constraints from the forklift sequences results in the identification of two motion processes.

Mixtures: 0.1866 0.7075 0.1059
Process 1: $\vec{T} = [0.0102 \ -0.0925 \ 0.9957]$ $f\vec{\Omega} = [2.09 \ 2.27 \ -0.10]$ $\vec{\sigma} = 0.07033$
Process 2: $\vec{T} = [-0.9948 \ 0.0295 \ 0.0972]$ $f\vec{\Omega} = [-4.13 \ -99.26 \ -5.18]$ $\vec{\sigma} = 0.05602$
Process 1: FOE = ( 13.20, -119.24)
Process 2: FOE = (-13137.98, 390.18)

Table 3: Clustering of the bilinear constraints from the forklift sequences allows for refined estimates of the recovered motion parameters.

seem puzzling, since all the linear constraints associated with it are formed across IMO boundaries, and are therefore expected to be outliers. As discussed in Section 2.1.4, the resultant constraints are expected to be an average of constraints which might be expected from either the floor or forklift alone. The forklift has considerably larger motion field values than the floor, so we find that the forklift flow samples bias the constraint towards the actual motion of the forklift.

The results from the linear clustering stage are used as initial guesses for the bilinear clustering. There are 36 bilinear constraints in total, one for each recovered flow sample. The segmented bilinear constraints are shown in Figure 9, and the numerical values for the recovered parameters are listed in Table 3.

Process #1 has recovered the egomotion parameters. The intersection of the constraints corresponds to the focus of expansion (FOE) of the egomotion. It is seen to be slightly above the centre of the image, suggesting the camera was aimed slightly downward. The second motion process has recovered the forklift motion, resulting in parallel constraints as seen in the plane  $x_3 = f$ . If they were projected onto the plane  $x_1 = \text{constant}$  they would intersect at a FOE in a manner similar to the constraints for the first motion process.

The ownership values for each of the constraints are shown in Figure 10. Process #2 clearly owns the constraints generated from the forklift. The first bilinear constraint, from the floor patch, also shows strong ownership by the forklift motion, since its flow vector is compatible with the motion of the forklift. This serves to highlight the fact that constraints may be owned by more than one process.

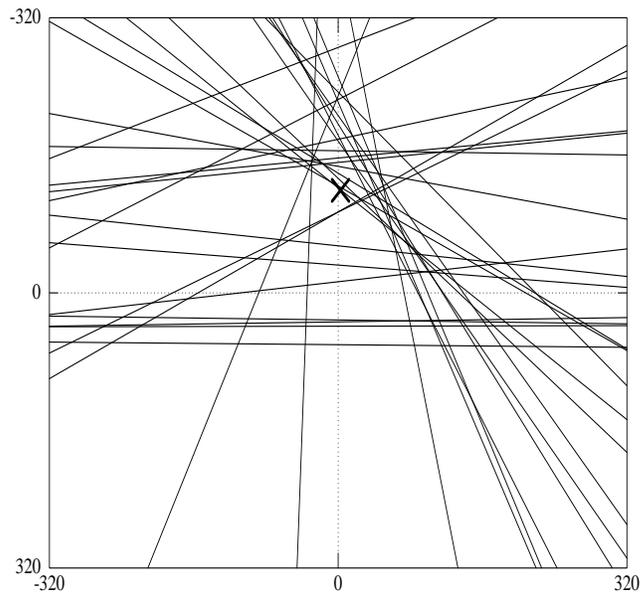


Figure 8: The linear translation constraints plotted in the image space for the frame shown in Figure 6. Each constraint defines a plane through the origin, which is shown intersecting the image plane  $X_3 = f$ . Units are shown in pixels. The 'x' shown is the FOE for the background motion after clustering the linear constraints. The FOE for the moving forklift is to the extreme left and does not appear in this figure.

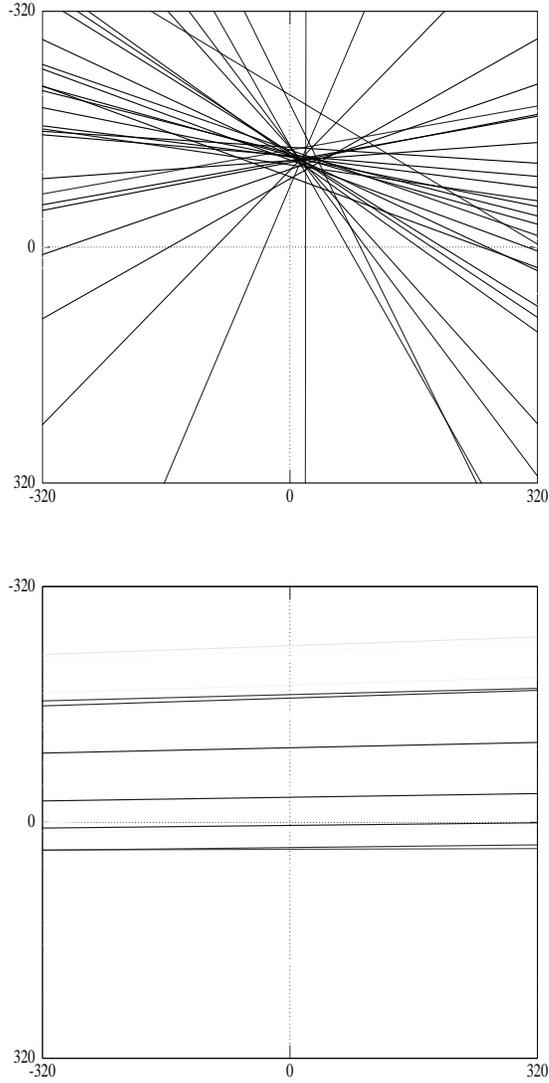


Figure 9: The top plot shows the translation constraints derived from the bilinear constraints as seen by the first motion. On the bottom is the plot of constraints as seen by the second motion. As seen in Figure 10, the second motion primarily owns constraints generated from the moving forklift. Grey-level indicates ownership probability for each motion.

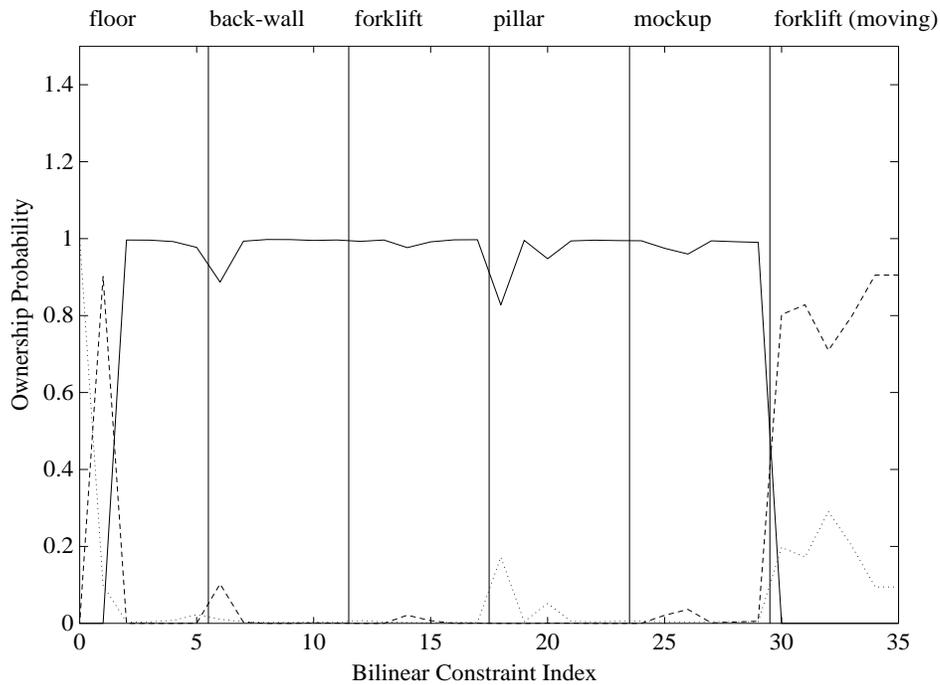


Figure 10: The ownership probabilities for the bilinear constraints as fitted by the EM algorithm. This plot has been segmented into six columns, each of which is labelled with the image patch from which its constraints are measured. The solid line indicates ownership by the first motion, the dashed line the second motion, and the dotted line the outlier process. Constraints from the floor, back-wall, forklift, pillar and mockup patches all show high ownership by the first motion (environment). The moving forklift patch shows high ownership by the second motion process, and is thus labelled as an IMO. Bilinear constraint #1 (from the floor patch) probably shows a strong ownership by the second motion process because its horizontal motion is consistent with that of the moving forklift.

### 3.4 Otte Sequence

This example shows the analysis of a sequence by Michael Otte in which the ground truth for the observer’s motion is known. The complete sequence has 30 frames in it: we analyze the 3-D motion for dense flow obtained from frames 25 and 26 (see Fig. 11). The flow was obtained using the methods outlined in [17, 16].

Because of the large planar surface present in the images, this sequence requires extra care when choosing flow samples for constructing linear constraints. Prior to sampling the flow, a quadratic model was robustly fit to the flow field in order to determine which flow samples are from the floor region and which are not. Figure 12 shows the probability that each image pixel belongs to the floor.

The following model was chosen for performing the fit:

$$\hat{v} = \begin{bmatrix} a_0 & \dots & a_5 \\ b_0 & \dots & b_5 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix} = M\vec{X}. \quad (15)$$

As an initial estimate for  $M$  the mean of the flow field was use (*i.e.*  $a_0 = \bar{v}_1$  and  $b_0 = \bar{v}_2$ , and all other  $a_i$  and  $b_i$  are zero). The model is then robustly fit using the EM algorithm. The probability that a given flow vector is a good fit to the model is given by

$$p(\vec{v}|\vec{x}, M) = \frac{1}{\det(C)^{1/2}(2\pi)^{m/2}} \exp\left\{-\frac{1}{2}\vec{e}^T C^{-1}\vec{e}\right\} \quad (16)$$

where  $\vec{e} = \vec{v} - M\vec{X}$ , and  $m = 2$ . The covariance matrix  $C$  is

$$C = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

Each iteration of the EM algorithm generates probability density estimates to be used as weights, and also generates new estimates of  $M$ .

The sampling of the flow field to create linear constraints is done according to the following scheme. First, we create 200 “patches” of 6 flow vectors each. Even numbered patches contain flow vectors randomly sampled from the quadratic surface ( $p > 0.6p_{max}$ ), and the odd numbered patches contain flow vectors randomly sampled from areas not belonging to the quadratic surface ( $p < 0.4p_{max}$ ). All flow

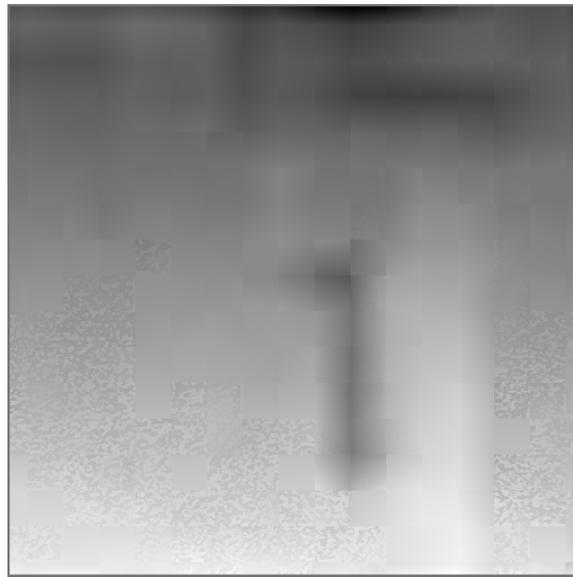


Figure 11: The horizontal (top) and vertical (bottom) components of the optic flow used as input to subsequent computations.

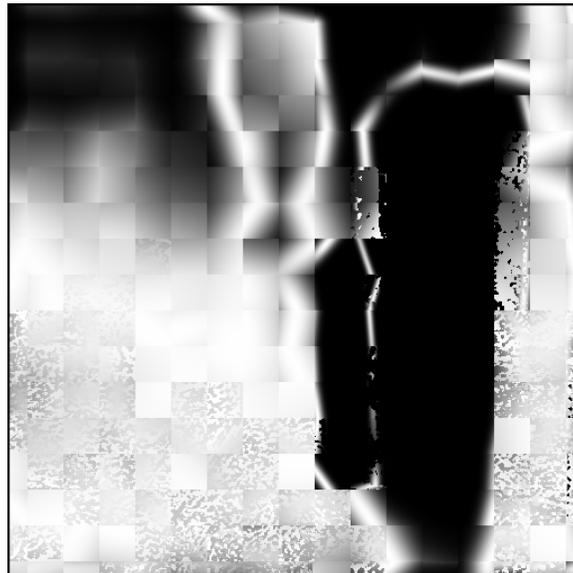
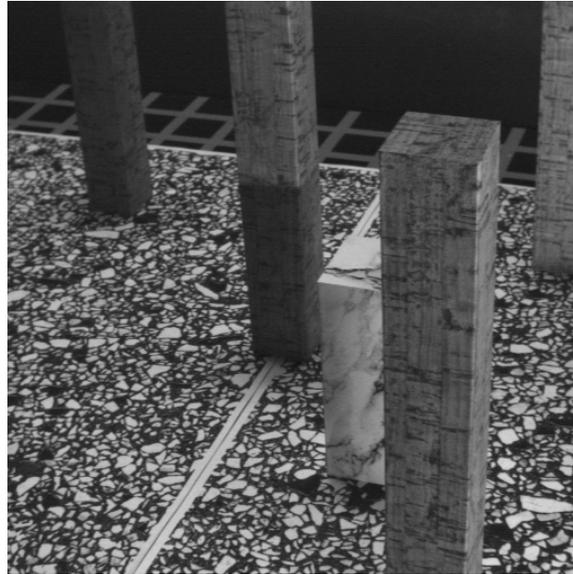


Figure 12: The bottom image shows the probability of each pixel in the dense flow image coming from a region of quadratic flow representing the floor region. The top image is the frame of the original sequence corresponding to the probability map.

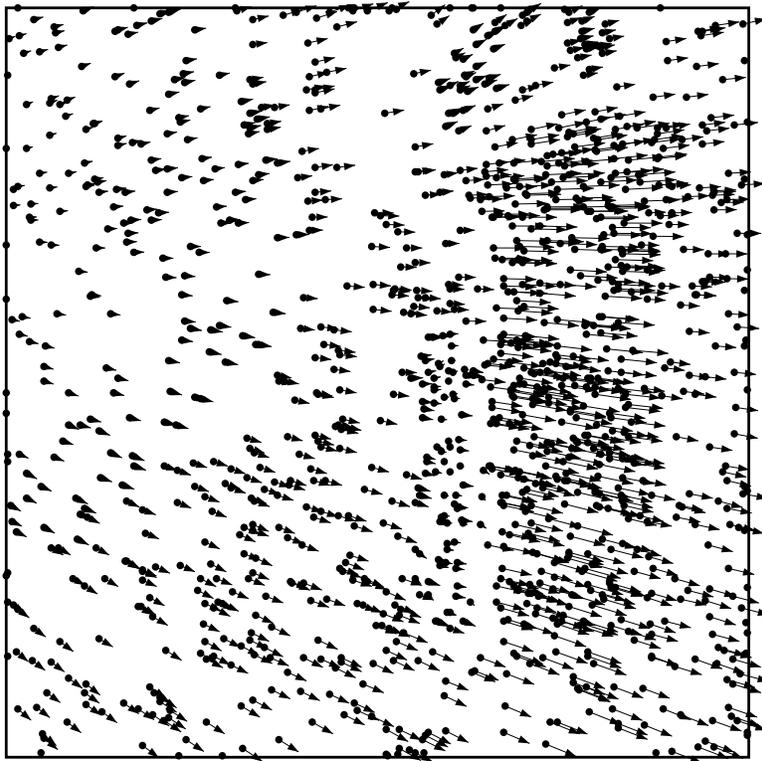


Figure 13: This figure shows the sampled flow vectors.

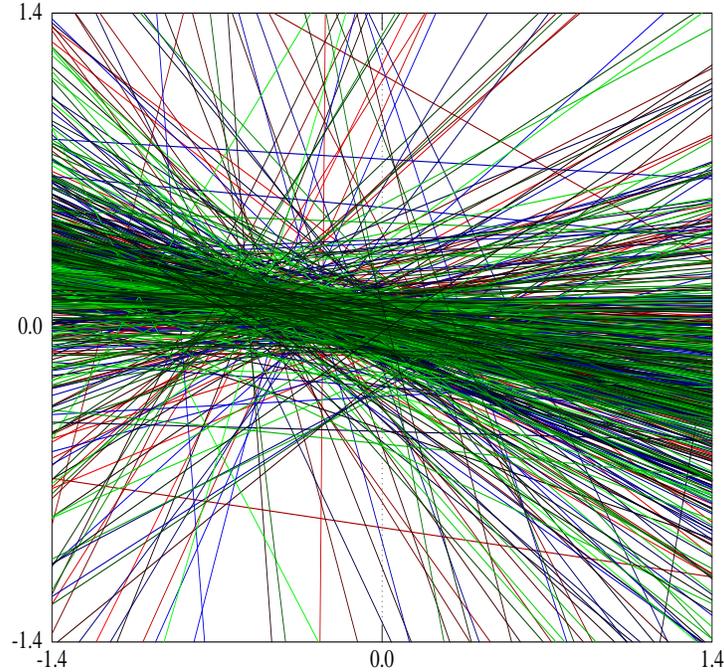


Figure 14: This figure plots the linear constraints obtained from the Otte sequence flow.

vectors are chosen to be unique. This means that no flow vector will be used in more than one patch. The sampled flow vectors are shown in Figure 13.

For generation of the linear constraints, each even numbered patch is paired with the subsequent odd numbered patch to create a sample of 12 flow vectors which are expected not to all come from a single planar surface. The resulting constraints are plotted in Figure 14.

The IMO in the image was not detected during clustering of the linear constraints, most likely due to the fact that it's motion lies strongly along the epipolar lines of the observer's motion.

For generation of the bilinear constraints, each flow vector from each patch was used as one constraint, giving a total of 1200 bilinear constraints for this sequence. Using the final value of  $\vec{T}$  as an initial guess, the bilinear constraints were robustly

fit. The final estimated translational direction is  $\vec{T} = [ -0.4879 \quad -0.0864 \quad 0.8686 ]^T$ . When compared to the ground-truth values of  $\vec{T} = [ -0.4799 \quad -0.1123 \quad 0.8701 ]^T$  we find the two values to differ by about  $1.68^\circ$ . The same analysis, except using uniform sampling of flow vectors, results in  $\vec{T} = [ -0.4394 \quad -0.0841 \quad 0.8944 ]^T$ . This represents an error of  $3.11^\circ$ . The result using non-uniform (intelligent) sampling is thus an improvement.

The clustering of the bilinear constraints, shown in Figure 15, is seen to be quite good. Examination of the outliers from the bilinear constraint fitting shows two regions of significant outliers. One region, towards the top-right of the image, is an area of little texture which gave rise to poor flow estimates. The second region involves flow estimates from the interior and boundary regions of the IMO. The outliers still give an indication of the presence of the IMO. The flow vectors are shown super-imposed over the image in Figure 16.

## 4 Discussion

In order to derive good estimates for 3-D motion, it is necessary to start with good flow estimates. In the last ten years a considerable improvement has been seen in the quality of flow estimates, making improved 3-D motion estimates correspondingly improved.

In the first two sequences presented the IMO was detected during the linear constraint clustering stage. While the motion parameters for the cube were not recovered correctly, the values recovered can be explained in terms of the expected behaviour of linear constraints which are formed across an IMO boundary. While the same is true for the forklift, the motion parameters recovered are quite reasonable, since the forklift's greater speed causes it's motion to dominate. In the Otte sequence, the IMO was not segmented during clustering of linear constraints. This again is expected, since the IMO's motion is strongly along the epi-polar lines for the egomotion, meaning that constraints generated from it will not appear as outliers, but rather as consistent with the egomotion. However, some outlier constraints are found along the IMO boundary, which allow us to at least be wary of the activity in this image region.

While it is not always possible to recover the IMO's motion, merely detecting it is of considerable value. It's detection could be used to trigger another method which would then recover the motion parameters.

Finally, also from the Otte sequence results, we see the necessity of sampling

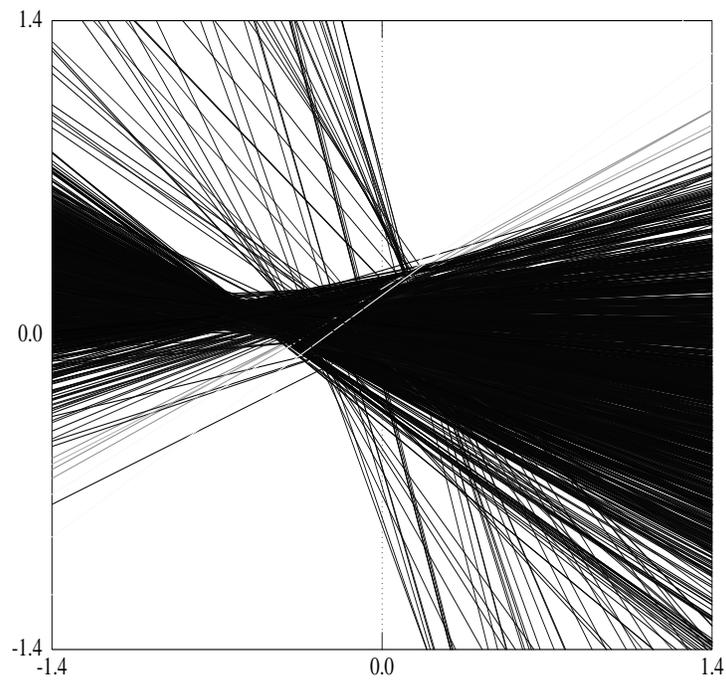


Figure 15: This figure shows the clustering of the bilinear constraints. A strong intersection is seen to the left, indicating the translational direction.

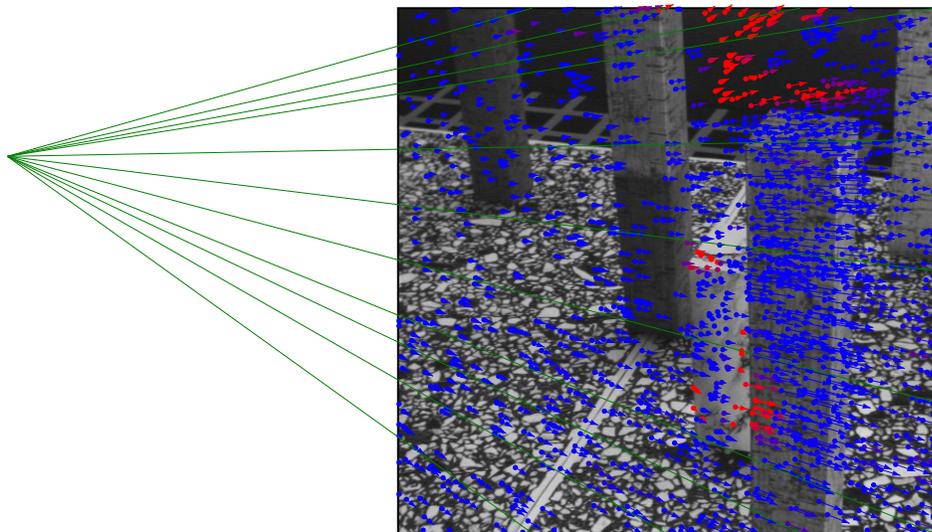


Figure 16: This figure shows frame 25 from the Otte sequence with flow vectors superimposed. The red flow vectors are outliers with respect to the fitting of the bi-linear constraints. The green lines are epi-poles found from the estimated direction of translation of the observer.

flow vectors in an intelligent manner. Had we randomly chosen flow vectors, most would have ended coming from the floor region. This, in turn, would have resulted in fewer linear constraints with a good signal to noise ratio. Since the main danger is one where a majority of the flow vectors are sampled from a single, dominating planar surface, it suffices to determine if such a surface exists. If it does, then we must limit the number of flow samples taken from it, and correspondingly increase the number of flow samples taken from other regions.

## 5 Conclusion

This paper has presented a method for performing 3-D motion segmentation on a sequence of images. The method uses optic flow estimates as an input, and constructs constraints on the 3-D motion parameters. These constraints are simultaneously segmented and motion parameters estimated by applying the EM algorithm to a finite mixture model in which one distribution describes each motion process, and an outlier distribution is used to capture constraints which are not well-fitted by any

of the motion processes. Clustering begins with the linear constraints, which are only dependent on the 3-D translation. At this stage, a method of splitting motion processes from the outlier population is used in an attempt to determine the number of motion processes present. This method is not guaranteed to find all motion processes, but it provides a tractable starting point for the problem. The results of the linear constraint clustering provides initial guesses for the number of motion processes and their parameters used to cluster the bilinear constraints. Due to the fact that bilinear constraints are less likely to be affected by IMO boundaries, improved motion parameter estimates are obtained. Finally, since each bilinear constraint corresponds to a single image location, the segmentation of these constraints will provide a 3-D motion-based segmentation of the image sequence.

The main place where this methodology can be improved is in the splitting of processes from the outlier population. If we consider spatially local clusters of linear constraints instead of just constraints which support a particular  $\vec{T}$ , we may be able to produce a better estimate of the number of motion processes. In a similar vein, one could imagine using a large number of motion processes, each with a unique and fixed preferred value for  $\vec{T}$ , being used to cluster the linear constraints. Each of these motion processes could be thought of as a sensor on the surface of the unit sphere which only becomes active if it senses enough compatible  $\vec{\tau}$  constraints. Underlying motion processes could be hypothesized based on the number and location of these active sensors.

## 6 Acknowledgements

The authors would like to thank Shannon Ju for providing the flow estimates for the Otte sequence, and David Wilkes for his help in capturing the forklift sequence.

## References

- [1] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans Pattern Analysis & Machine Intelligence*, 7(4):384–401, 1985.
- [2] Anna R. Bruss and Berthold K. P. Horn. Passive navigation. *Computer Vision, Graphics and Image Processing*, 21:3–20, 1983.

- [3] Trevor Darell and Alexander Pentland. Robust estimation of a multi-layered motion representation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 173–177, Princeton, New Jersey, October 1991.
- [4] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.
- [5] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. The MIT Press, Cambridge, Massachusetts, 1993.
- [6] Xiaolin Feng and Pietro Perona. Scene segmentation from 3d motion. In *Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbara*, pages 225–231, June 1998.
- [7] David J. Fleet. *Measurement of Image Velocity*. Kluwer Academic Publishers, Boston, Massachusetts, 1992.
- [8] David J. Fleet and Allan D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.
- [9] David J. Fleet and Allan D. Jepson. Stability of phase information. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 52–60, Princeton, New Jersey, October 1991.
- [10] J.J. Gibson. *The perception of the visual world*. Houghton Mifflin, Boston, Massachusetts, 1950.
- [11] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: the approach based on influence functions*. John Wiley & Sons, Inc., New York, 1986.
- [12] Michael Jenkin and Allan D. Jepson. Detecting floor anomalies. In *Proceedings of the British Machine Vision Conference*, September 1994.
- [13] Allan D. Jepson and Michael J. Black. Mixture models for optical flow computation. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 760–761, Los Alamitos, California, June 1993. IEEE Computer Society Press.

- [14] Allan D. Jepson and David J. Heeger. A fast subspace algorithm for recovering rigid motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 124–131, Princeton, New Jersey, October 1991.
- [15] Allan D. Jepson and David J. Heeger. Linear subspace methods for recovering translational direction. In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots*. Cambridge University Press, 1993. See also: Research in Biological and Computational Vision, Department of Computer Science, University of Toronto, RBCV-TR-90-40, Apr. 1992.
- [16] Shanon X. Ju, Michael J. Black, and Allan D. Jepson. Estimating image motion in layers: the “skin and bones” model. *International Journal of Computer Vision*, 1998.
- [17] Shanon Xuan Ju, Michael J. Black, and Allan D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *Proceedings Computer Vision and Pattern Recognition Conference, San Francisco*, pages X–Y, June 1996.
- [18] Kenichi Kanatani. *Geometric computation for machine vision*. Clarendon Press, Oxford, England, 1993.
- [19] Kenichi Kanatani. Renormalization for unbiased estimation. In *Proceedings of the 4th International Conference on Computer Vision*, pages 599–606, Berlin, Germany, May 11–14 1993.
- [20] W. James MacLean. *Recovery of Egomotion and Segmentation of Independent Object Motion Using the EM Algorithm*. PhD thesis, University of Toronto, 1996.
- [21] W. James MacLean. Removal of translation bias when using subspace methods. In *Proceedings of the 7th International Conference on Computer Vision*, pages 753–758, Los Alamitos, CA, September 1999. IEEE Computer Society Press.
- [22] W. James MacLean, Allan D. Jepson, and Richard C. Frecker. Recovery of egomotion and segmentation of independent object motion using the EM algorithm. In *Proceedings of the 1994 British Machine Vision Conference*, pages 175–274, York, England, September 1994. British Machine Vision Association.

- [23] Geoffrey J. McLachlan and K. E. Basford. *Mixture models: Inference and applications to clustering*. Marcel Dekker Inc., New York, 1988.
- [24] Randal C. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, 7(1):33–46, 1991.
- [25] D. Sinclair. Motion segmentation and local structure. In *Proceedings of the 4th International Conference on Computer Vision*, pages 366–373, Berlin, Germany, May 11–14 1993.
- [26] Stefano Soatto and Roger Brockett. Optimal structure from motion: Local ambiguities and global estimates. In *Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbara*, pages 282–288, June 1998.
- [27] Tina Y. Tian, Carlo Tomasi, and David J. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 315–320. IEEE Computer Society, IEEE Computer Society Press, June 1996.
- [28] John Y. A. Wang and Edward H. Adelson. Layered representation for motion analysis. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 361–366, Los Alamitos, California, June 1993. IEEE Computer Society Press.