ELSEVIER

Artificial Intelligence 78 (1995) 45-86

Artificial Intelligence

3-D motion estimation from motion field*

Naresh C. Gupta [a,b], Laveen N. Kanal [a,b]

[a] *Department of Computer Science, University of Maryland, College Park, MD, USA*

[b] *L.N.K. Corporation, Inc., Riverdale, MD, USA*

Received October 1993; revised January 1995

Abstract

Several experiments suggest that the first stage of motion perception is the measurement of visual motion. The result of this stage is called the *motion field*, which assigns a velocity vector to each point in the image plane. The second stage involves interpreting the motion field in terms of objects and motion in the three-dimensional world. Recovering 3-D motion of the object from the motion field has been difficult owing to the nonlinear system of equations involved, and the sensitivity of the system to noise. The need for the stability of the system is essential as only the optical flow field can be recovered from a sequence of images, which is at best a crude approximation to the motion field.

We define two sets of "basic" parameters, which can be recovered from the motion field by solving a linear system of equations. The relationship between the basic parameters and the motion parameter being one-to-one and linear, we obtain a closed form solution for the 3-D motion parameter by solving a system of linear equations only. We prove the correctness, completeness and robustness of the approach and in that sense the problem of recovering the motion parameter from the motion field may be said to be "solved". We present the results of extensive experimentation with real and simulated image sequences.

1. Introduction

A single image provides only two-dimensional (2-D) information about the three-dimensional (3-D) world, as the 3-D information (depth and motion) is lost during the image formation. However, there are several cues in the images which enable us to recover the lost 3-D information to a certain extent. A valuable source of 3-D information is provided by the relative motion of elements in the changing 2-D image. Several

* Supported by NSF Research Grant # IRI 8802419 and in part by LNK Corporation.

SSDI 0004-3702(95)00031-3

experiments have demonstrated the remarkable ability of the human visual system to reliably extract 3-D structure and motion from an image sequence in the absence of any other cue. A particularly compelling method of demonstrating this phenomenon is to present observers with apparent motion sequences of a random dot surface undergoing motion in space.

There are several experiments which indicate that the first stage of motion perception is the measurement of visual motion. The result of this stage is called the *motion field*, which assigns a velocity vector to each point in the image plane. The second stage involves interpreting the motion field in terms of objects and surfaces in the three-dimensional world. In this paper we present a solution for the second stage of the image sequence interpretation problem. We present an algorithm to recover the 3-D motion parameter of a rigid object undergoing a general motion.

Image sequence interpretation has been an active area of research for the last two decades. Several algorithms have been proposed to recover 3-D motion from a sequence of images. We present a brief overview of the previous efforts in Section 2. Though many algorithms have been proposed, none works well for the noisy, real-world image sequences. In Section 3, we discuss the reasons and underline the associated difficulties with the problem. In Section 4 we define two sets of parameters which can be recovered from the motion field via solving a linear system of equations; we also show that the 3-D motion parameter is related to these basic parameters via a linear and one-to-one relationship. In the subsequent sections we look at the correctness, completeness and robustness of the approach and in that sense these sections represent the most significant difference between this work and previous efforts. In the next section we stabilize the solution using nonlocal constraints and Green's divergence theorem. We then present details of our extensive experimentation. Some general observations and the relevance of this work are presented in the concluding section.

2. Literature review

Gibson hypothesized [4–6] that there is sufficient information in the flow field to recover a unique, physically correct interpretation of 3-D motion and structure. Since then numerous algorithms have been proposed to interpret the motion field for the 3-D information. These algorithms can be grouped into six main categories.

2.1. *Closed form solution from flow fields*

Koenderink and van Doorn [19] proposed an algorithm to measure the invariant properties of motion and surface shape by using the local deformations in the flow field (e.g., divergence, curl, shear). Longuet-Higgins and Prazdny [24] and Waxman [38] built on their work and obtained a closed form solution for the flow field in terms of flow derivatives. A serious problem with these approaches is their sensitivity to noise in the flow field.

2.2. Least squares methods

Bruss and Horn [2] proposed a global approach that combines information in the entire visual field to choose the 3-D motion and structure that fits the flow field best in the least squares sense. The cases when there is only translation or only rotation are simple and stable closed form solutions are derived. In the case of a general motion, iterative techniques were used to solve a nonlinear system of equations for the translational parameter.

Adiv [1] minimizes the same residual function as Horn and Bruss, but in a different way. The flow field is divided into patches and motion parameters are then estimated for each patch under the assumption of planarity. For each patch and each translation direction, an error measure is computed, from which a best translation is selected for each patch. Patches having similar motion are merged.

Other researchers have also proposed techniques that pick solutions by minimizing a residual function over a discretely sampled solution space. Lawton [21] used it to obtain the focus of expansion of a translating camera. Prazdny [31] proposed an approach in which the velocity field is decomposed into rotational and translational components. The rotational motion is hypothesized and the best translation is estimated for the remaining flow field.

Heegar and Jepson [12] minimize the same residual function as Horn and Bruss. The depth and the rotation parameters are eliminated to arrive at a measure of error as a function of the translation. The error function is evaluated for each candidate translation for a large number of patches from which a translation is selected for each patch. The translation direction which is selected for most patches is chosen to be the translation.

2.3. Subdividing the problem

Several researchers have proposed splitting the problem of structure and motion estimation into several subproblems. The idea is to eliminate one set of motion parameters and get a constraint only in rotation or translation. Solving that constraint gives the value of that set of parameters, which is then used to estimate the other. Prazdny [31] proposed solving first for the rotational component of motion. Under the assumption of smooth surfaces, he derived a system of nonlinear equations in terms of rotation only.

Helmholz pointed out that if two points at different depths are projected onto the same image point, the difference in the image velocities points towards the focus of expansion. Reiger and Lawton [32] proposed an approximate algorithm to estimate the focus of expansion from local flow-vector differences. The problem with this approach is the noise sensitivity in computing the flow difference.

2.4. Direct methods

Negahdaripour and Horn [29] proposed several direct methods for recovering the 3-D motion parameters. They substituted the motion field equations in the optical flow constraint, to get a constraint in terms of 3-D parameters and image intensity gradients.

Several algorithms were proposed for special cases of object motion; the motivation being to avoid the computation of the optical flow—which has proved to be difficult. Hanna [10] proposed an iterative algorithm that estimates the motion of the camera directly from the image intensity.

2.5. Long sequence

Several algorithms have been proposed which use a sequence of flow fields to estimate the motion parameters. Matthies and Kanade [27] used a precise dynamical model for temporal changes in depth to recover structure from a sequence of images. The limitation of their approach is the need to know the camera motion as well as the restriction that motion may only be translation and parallel to the image plane. Heel [13] used one Kalman filter per pixel to recover a dense depth map.

2.6. Linear methods

Tsai and Huang [35] proposed a linear algorithm for a planar patch. First, a set of eight "pure parameters" are determined by solving a set of linear equations. The motion and structure parameters are recovered from these "pure parameters" by solving a sixth order polynomial. Several linear algorithms were proposed later [22,36] for a general curved object in motion. The "essential" parameters were first recovered from eight feature point correspondence. Motion parameters are then recovered by taking the SVD of the essential parameter matrix E. The E-matrix algorithms are quite simple and efficient. But they are highly sensitive to noise, especially to perturbations of image coordinates. One of the reasons for this is that the linear constraint they utilize is weaker than the rigid body motion constraint.

Jepson and Heegar [17] proposed an elegant solution to estimate the motion parameter by decomposing the problem into three steps. The linear constraint for the translation parameter is obtained by eliminating the rotational component locally and closely resembles the current work. Once the translation is known, rotation and depth map can be estimated easily. We discuss their approach in detail later and show how our work improves on their effort. Sundareswaran [34] proposed an approximate method to compute the motion from the motion field. A constraint is obtained which is approximately linear.

3. Interpreting the motion field

In this section we introduce the problem of motion field interpretation. We present the set of equations which relate the motion field to the 3-D motion and shape information. We discuss the difficulties associated with interpreting the motion field and present a brief overview of our solution methodology.

For an observer moving with the translational velocity $\boldsymbol{t} = (t_1, t_2, t_3)$ and rotational velocity $\boldsymbol{\omega} = (A, B, C)$ with respect to the surface, the image motion field is given by [15]:

$$u(x,y) = (-t_1 + xt_3)h(x,y) + Axy - B(x^2+1) + Cy, \tag{1}$$

$$v(x,y) = (-t_2 + yt_3)h(x,y) + A(y^2+1) - Bxy - Cx, \tag{2}$$

where h is the inverse depth map $1/Z$.

The problem of understanding the motion field can be stated as follows:

Given the motion field $(u(x,y), v(x,y))$ *over the image plane* R, *find the structure of the scene* $h(x,y)$, *and the scaled* 3-*D motion* (t_1, t_2, t_3, A, B, C).

For an image represented discretely at n points in space, there are $n+5$ unknowns: depth at n points and 5 motion parameters (it is easy to see why we can recover the translation up to a scalar constant only) and $2n$ motion field constraints. The problem is thus sufficiently constrained and can be solved easily, in theory. However, in practice, it turns out to be an extremely difficult problem due to the following two reasons:

- nonlinearity,
- noise sensitivity.

3.1. Nonlinearity

The task of understanding the motion field for 3-D motion is difficult because the relationship between the motion field, 3-D motion and depth turns out to be nonlinear. As we mentioned earlier, the problem is sufficiently constrained, and can be solved easily given a technique to solve a system of nonlinear equations. Unfortunately, nonlinear equations of the form (1) and (2) are rather difficult to solve and none of the approaches guarantees a solution. A solution may be obtained for a very constrained situation of scene geometry and motion, but we have to resort to iterative techniques for the general case. These iterative techniques require enormous computations, do not converge, and are extremely difficult to analyze for stability and performance.

Nonlinearity is difficult to handle; for any hope of solving the problem in a realistic sense, nonlinearity must be avoided, i.e., the nonlinear constraints must be reduced in some way to yield a linear relationship among various variables.

3.2. Noise sensitivity

The second major problem with interpreting the motion field is the sensitivity of the motion parameter to the noise in the flow field. The problem is compounded by the fact that the motion field is only an abstract concept and cannot be recovered from a sequence of images. What is recoverable from a sequence of images is the normal flow; and even that is corrupted by noise due to discretization and numerical derivative computation. Optical flow, an approximation of the motion field, has to be recovered from the sequence of images by making additional assumptions about the local structure of the underlying object. What we have at best, is a very crude approximation of the motion field.

It is evident from this that if we hope to obtain a system to interpret an image sequence for shape and motion, the algorithm for recovering shape and motion from the motion field should be extremely robust and must tolerate large amounts of noise.

We have argued elsewhere [7], that the stability of vision algorithms can be achieved by using the abundance of information available in space and time in a useful way. We believe that the reason noise has come to be such a nemesis in computer vision is due to the inability of the algorithms to invoke significant spatio-temporal support to stabilize the constraints.

Several algorithms try to recover the shape and motion by looking at the local structure of the flow field. We reiterate that in this noise corrupted world, there is not enough reliable information to solve such a difficult problem locally. The local constraints are "overworked", i.e., they try to do too much with the unreliable local information that is available. In the case of motion field interpretation, the local constraints try to estimate the local variations in the flow field which are almost impossible to recover in a reliable way. An important question arises:

> *Why do we have to solve the problem locally and let so much information go to waste?*

In the next several sections we present an algorithm to overcome the above-mentioned problems. The proposed approach solves a system of linear equations only and uses all the available information in space so as to yield a stable system. We emphasize the key issues again:

- We cannot handle more than linear constraints reliably. Linear constraints have to be obtained for any chance of success.
- Stability of the system has to be a prime criterion and the information available in space suggests a way to achieve that.

4. Recovering motion parameters

In this section we present an algorithm to recover the motion parameter from the motion field. We define two sets of *basic parameters* which can be recovered from the motion field by solving a linear system of equations. We show that a unique translation and the rotation vector can be recovered from either set of basic parameters.

4.1. Basic parameters

Eliminating depth from the motion field equations (1) and (2) we get:

$$\frac{u(x,y) - Axy + Bx^2 + B - Cy}{v(x,y) - Ay^2 - A + Bxy + Cx} = \frac{xt_3 - t_1}{yt_3 - t_2},$$

which after some algebra yields

$$\begin{aligned}&(Bt_2 + Ct_3)x^2 + (At_1 + Ct_3)y^2 - (At_2 + Bt_1)xy\\&\quad - (At_3 + Ct_1)x - (Bt_3 + Ct_2)y + (At_1 + Bt_2)\\&\quad - t_1v(x,y) + t_2u(x,y) + t_3(xv(x,y) - yu(x,y)) \equiv 0 \qquad (3)\end{aligned}$$

for all (x,y) in the image plane. Note that if (t_1,t_2,t_3) is a solution for the above equation, then (kt_1,kt_2,kt_3) is also a solution for any nonzero k and hence the scale

ambiguity. That is, we can recover translation (t_1, t_2, t_3), such that $|t|$ is unity. Also, note that we cannot estimate if the motion is towards or away from the image plane. However, the direction ambiguity can be resolved as only one of the directions will yield positive depth from the motion field equations (1) and (2).

We can rewrite the above equation (3) as:

$$ax^2 + by^2 - cxy - dx - ey + f - t_1 v(x,y) + t_2 u(x,y) + t_3 w(x,y) \equiv 0 \qquad (4)$$

where

$$a = Bt_2 + Ct_3, \quad b = At_1 + Ct_3, \quad c = At_2 + Bt_1,$$
$$d = At_3 + Ct_1, \quad e = Bt_3 + Ct_2, \quad f = At_1 + Bt_2,$$
$$w(x,y) = xv(x,y) - yu(x,y).$$

The *key observation* is that given t and any three of (a,b,c,d,e,f), the motion parameter can be recovered by solving a system of linear equations. It transpires that if a linear constraint can be obtained for recovering t and any three of (a,b,c,d,e,f), the motion parameter can be recovered by solving a linear system of equations only. A family of algorithms is thus possible, depending on which three of the coefficients are chosen. We describe two of them in detail. Several other combinations were also tried but failed to yield a stable system.

Note that if t is given along with four or more of the coefficients, no motion parameter may be obtainable unless the given coefficients were obtained with additional constraints. That leads us to the reason why the linear equation (4) cannot be solved by itself for the coefficients and the translation. The resulting solution may not have any motion parameter as a solution unless we impose three additional constraints. These three additional constraints turn out to be nonlinear in form and hence make the system difficult to solve.

We define

$$\mathcal{B}_1 = (a, c, d, t_1, t_2, t_3), \qquad (5)$$

$$\mathcal{B}_2 = (b, c, e, t_1, t_2, t_3) \qquad (6)$$

to be the two sets of *basic parameters*. We show below how the motion parameter can be uniquely determined from either basic parameter set.

4.2. Motion from basic parameters

In this section we show that the translation and the rotation vector can be uniquely recovered from either $\mathcal{B}_1$ or $\mathcal{B}_2$. Translation can be determined directly from $\mathcal{B}_1$. Once the translation is known, rotation can be expressed in terms of $\mathcal{B}_1 = (a, c, d, t_1, t_2, t_3)$ as

$$\begin{bmatrix} 0 & t_2 & t_3 \\ t_2 & t_1 & 0 \\ t_3 & 0 & t_1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} a \\ c \\ d \end{bmatrix}. \qquad (7)$$

Eq. (7) has a unique solution unless $t_1(t_3^2 + t_2^2)$ is zero, which is zero if and only if t_1 is zero or both t_2 and t_3 are zero. In case t_1 is zero, Eq. (7) reduces to

$$\begin{bmatrix} 0 & t_2 & t_3 \\ t_2 & 0 & 0 \\ t_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} a \\ c \\ d \end{bmatrix} \tag{8}$$

which yields two estimates for A and an infinite number of solutions for B and C which lie on a line. If t_2 and t_3 are zero then unique solutions for B and C are obtained whereas A is undetermined. Hence, the basic parameter $\mathcal{B}_1$ yields a unique 3-D motion unless t_1 is zero or both t_2 and t_3 are zero.

Similarly, translation can be recovered directly from $\mathcal{B}_2 = (b, c, e, t_1, t_2, t_3)$ and the rotation is given by

$$\begin{bmatrix} t_1 & 0 & t_3 \\ t_2 & t_1 & 0 \\ 0 & t_3 & t_2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} b \\ c \\ e \end{bmatrix}. \tag{9}$$

It can be easily verified that there is a unique solution for the above equation unless t_2 is zero or both t_1 and t_3 are zero. In case of t_2 being zero, we get two estimates for B and an infinite number of solutions for A and C.

Remark. We revisit the case when four or more of the coefficients are given along with t. Suppose we are given (a, b, c, d), we have

$$\begin{bmatrix} 0 & t_2 & t_3 \\ t_1 & 0 & t_3 \\ t_2 & t_1 & 0 \\ t_3 & 0 & t_1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix},$$

a system of four linear equations in three variables. The system is over-constrained and has a unique solution iff

$$(t_3^2 + t_2^2)(b - dt_1)t_1 = (1 - t_1^2)(at_1 - t_2c + dt_2^2).$$

The above constraint has to be satisfied by the coefficients if we wish to obtain a solution for the motion parameter. It can be guaranteed only if we impose the nonlinear constraint when recovering the basic parameters. It is easy to see that it will make the problem extremely difficult as the linear technique described below for recovering basic parameters cannot be used.

4.3. Recovering basic parameters

In this section we recover basic parameters from the flow field. We present a linear constraint between basic parameters, the motion field and its spatial derivatives from which a unique basic parameter and hence a unique 3-D motion can be obtained.

Differentiating Eq. (4) with respect to x and y respectively, we get

$$2ax - cy - d - t_1v_x + t_2u_x + t_3w_x \equiv 0, \tag{10}$$

$$-cx + 2by - e - t_1 v_y + t_2 u_y + t_3 w_y \equiv 0, \tag{11}$$

where

$$u_x = \frac{\partial u}{\partial x}; \quad v_x = \frac{\partial v}{\partial x}; \quad u_y = \frac{\partial u}{\partial y}; \quad v_y = \frac{\partial v}{\partial y}; \quad w_y = \frac{\partial w}{\partial y}; \quad w_y = \frac{\partial w}{\partial y}.$$

We call the above two constraints *BaP constraints*. Eq. (10) imposes a linear constraint on $\mathcal{B}_1$ and Eq. (11) on $\mathcal{B}_2$. The basic parameters can be easily estimated, given the motion field and its first derivatives for sufficient points. In particular, given u, v, u_x and v_x at five points in the image plane, $\mathcal{B}_1$ is given by the null space of the matrix

$$\begin{bmatrix} 2x_1 & -y_1 & -1 & -v_x(x_1,y_1) & u_x(x_1,y_1) & w_x(x_1,y_1) \\ 2x_2 & -y_2 & -1 & -v_x(x_2,y_2) & u_x(x_2,y_2) & w_x(x_2,y_2) \\ 2x_3 & -y_3 & -1 & -v_x(x_3,y_3) & u_x(x_3,y_3) & w_x(x_3,y_3) \\ 2x_4 & -y_4 & -1 & -v_x(x_4,y_4) & u_x(x_4,y_4) & w_x(x_4,y_4) \\ 2x_5 & -y_5 & -1 & -v_x(x_5,y_5) & u_x(x_5,y_5) & w_x(x_5,y_5) \end{bmatrix}. \tag{12}$$

It is easy to show that the above system has nonunique null space if the surface is quadratic in x or the five points are collinear. Similarly, if u, v, u_y and v_y are known at five points on the image plane, $\mathcal{B}_2$ can be recovered by obtaining the null space of a matrix.

Generally, we can extract a dense flow field and thus will have available to us the flow field and its derivatives at significantly more points. We wish to use all the available information to improve the stability of the system. The problem now reduces to:

$$\boldsymbol{m}(x,y) \cdot \mathcal{B}_1 = 0, \quad \forall (x,y) \in R,$$

or

$$M\mathcal{B}_1 = \mathbf{0}, \tag{13}$$

where

$$\boldsymbol{m} = (2x, -y, -1, -v_x, u_x, w_x),$$

and M is a $XY \times 6$ matrix and X, Y is the image size.

The above is a homogeneous system of equations in six variables and has a nontrivial solution iff the system has rank 5 or less. The solution space is given by the null space of matrix M which can be easily obtained by performing the singular value decomposition of M. The null space is spanned by singular vectors with zero singular values. [1]

5. Motion parameter: analysis

In the previous section we presented an algorithm to compute the motion parameter from the motion field. The proposed approach recovers a unique motion parameter by

[1] Matrix M has at least one zero singular value as it has rank 5 or less.

solving two systems of linear equations. In this section we prove the correctness and the completeness of the approach. A summary of the results is:

- The motion parameter obtained is the correct motion parameter.
- A unique motion parameter is obtained whenever the motion field yields a unique 3-D interpretation.
- Infinite motion parameters are obtained in case of a planar or a quadratic surface in motion. Moreover, for quadratic surfaces the observer should be on the surface itself. These are the cases when the motion field yields multiple 3-D interpretations.

5.1. *Correctness of the motion parameter*

In this section we prove the correctness of the approach. In particular, we prove that the motion parameter obtained by solving Eqs. (13) and (7) is indeed the correct motion parameter.

Let $\boldsymbol{\alpha} = (A, B, C, t_1, t_2, t_3)$, and the function $\mathcal{P}(\boldsymbol{\alpha}, x, y)$ be defined as:

$$\begin{aligned}\mathcal{P}(\boldsymbol{\alpha}, x, y) = {} & ax^2 + by^2 - cxy - dx - ey + f \\ & - t_1 v(x, y) + t_2 u(x, y) + t_3 (xv(x, y) - yu(x, y)),\end{aligned} \tag{14}$$

which is the left-hand side of Eq. (3). The problem of recovering 3-D motion from the motion field can now be stated as:

> *given a motion field* $(u(x, y), v(x, y))$ *over* R, *find* $\boldsymbol{\alpha}$ *such that* $\mathcal{P}(\boldsymbol{\alpha}, x, y)$ *is zero overall.*

We assume that there is at least one solution to the motion recovery problem, i.e. the motion field is generated due to the motion between the sensor and a rigid object. Let $\boldsymbol{\alpha}_o$ be the correct motion parameter and $\Omega_o = \{\boldsymbol{\alpha}_o\} \cup \Omega$ be the set of solutions for Eq. (14).

Let Ω_1 be the set of solutions for Eq. (15) which is obtained by differentiating $\mathcal{P}$ with respect to x:

$$\begin{aligned}\mathcal{P}_x(\boldsymbol{\alpha} x, y) = {} & 2ax - cy - d \\ & - t_1 v(x, y) + t_2 u(x, y) + t_3 (xv(x, y) - yu(x, y)).\end{aligned} \tag{15}$$

Since

$$\mathcal{P}(\boldsymbol{\alpha}_o, x, y) \equiv 0 \quad \Rightarrow \quad \frac{\partial \mathcal{P}(\boldsymbol{\alpha}_o, x, y)}{\partial x} \equiv 0,$$

the set of solutions for Eq. (15) is a superset of the set of solutions for Eq. (14). We have,

$$\{\boldsymbol{\alpha}_o\} \cup \Omega \subseteq \Omega_1.$$

The solution set for Eq. (15) is obtained by solving two linear systems of equations; first obtain the basic parameters and then the motion parameter. A linear system of equations is either consistent, inconsistent or singular resulting in a unique, none or

infinite solutions, respectively. But, since Eq. (14) has at least one solution, Eq. (15) must have at least one solution. We are left with two cases:

Case 1: *Eq.* (15) *has a unique solution.* This happens when matrix M in Eq. (13) is of rank 5 and t_1 and t_2 or t_3 are nonzero. Let the unique solution be $\boldsymbol{\alpha}_1$. From the above we have,

$$\{\boldsymbol{\alpha}_o\} \cup \Omega \subseteq \{\boldsymbol{\alpha}_1\} \quad \Rightarrow \quad \boldsymbol{\alpha}_o = \boldsymbol{\alpha}_1 \text{ and } \Omega = \emptyset.$$

To rephrase, if Eq. (15) has a unique solution

- the correct 3-D motion parameter is obtained,
- this is the only solution for the motion field problem.

Case 2: *Eq.* (15) *has infinite number of solutions.* This happens when matrix M in Eq. (13) has rank 4. We cannot say which one of these infinite solutions is the solution for Eq. (14), and these are the cases when a unique 3-D motion is not obtained. In the next section we show that these are exactly the cases when the motion field has multiple shape and motion interpretation.

5.2. Completeness of motion parameters

In this section we consider the completeness of the approach. In particular, we prove that if the given motion field has a unique motion and shape interpretation, the correct motion parameter is obtained. Moreover, a unique solution is not obtained only if the motion field renders multiple motion and shape interpretations.

In the previous subsection we proved that either a unique solution or an infinite number of solutions is obtained for the motion parameter. The cases when infinite motion parameters are obtained are:

(1) No unique basic parameter is obtained.
(2) Only a unique $\mathcal{B}_1$ is obtained and t_1 is zero or t_2 and t_3 are zero.
(3) Only a unique $\mathcal{B}_2$ is obtained and t_2 is zero and t_1 and t_3 are zero.
(4) Unique $\mathcal{B}_1$ and $\mathcal{B}_2$ are obtained with both t_1 and t_2 being zero.

Except for case (1), a unique translation is obtained anyway and the ambiguity is only in rotation. Later, we outline a way to resolve the ambiguity in rotation for one of the three cases—others can be handled similarly. This leaves us with a nonunique motion parameter only if no unique basic parameter is obtained.

A nonunique estimate for the basic parameter $\mathcal{B}_1$ (and $\mathcal{B}_2$) is obtained only if matrix M in Eq. (13) (and similar for $\mathcal{B}_2$) is of rank 4 or less. We show in Appendix A that this happens only if the object is planar or is a certain type of quadratic surface. Moreover, the viewer must be on the surface itself to yield infinite solutions for the basic parameters. Specifically, the quadratic surfaces that give rise to infinite 3-D motion estimation belong to the class of seven parameter surfaces described by

$$s_1X^2 + s_2Y^2 + s_3Z^2 + s_4XY + s_5XZ + s_6YZ - X - Y + s_7Z = 0,$$

which is a subset of all the possible quadratic surfaces. It has been proved [15,25] that except for some quadratic surfaces, the motion field leads to an unambiguous motion and shape. We also prove that, but in a different way; ours is a constructive

proof whereas Horn's proof is existential. It has been proved [11, 15, 23, 26, 28] that in case of planar or quadratic surfaces, the motion field has more than one correct interpretation. Therefore, we do not obtain a unique motion parameter only when the motion field has multiple correct interpretations. Furthermore, we obtain the motion parameters whenever the motion field yields a unique interpretation and in that sense our approach for recovering the motion parameter is complete.

Remark. Above we discussed the case when no basic parameters are obtained. Infinite motion parameters are obtained in three other situations (2), (3), and (4) discussed above. We outline a solution for one of them here; for others it is similar. In case of only $\mathcal{B}_1$ available with t_1 equal to zero, two estimates for A and infinite solutions are obtained for B and C (which lie on a line, see Section 4.1). The two estimates for A must be equal as there is at least one correct motion interpretation. The ambiguity for the B and C can be broken by going back to Eq. (4) and using the correct estimates of t_1, t_2 and A to obtain a linear constraint for B and C.

6. Solving an overdetermined system

In the last two sections we presented an algorithm to recover the motion parameter from the motion field. A solution was developed in terms of the singular vectors of the matrix M for the noise free motion field. We also proved the correctness and the completeness of the approach. In this section we develop the solution for the noisy motion field.

Let ΔM be the error introduced in matrix M due to the noise in the motion field. The problem now is reduced to estimating the basic parameters such that

$$(M + \Delta M)\mathcal{B}_1 \approx 0.$$

We are interested in techniques which are robust, yield an unbiased estimate, and degrade gradually.

There are several techniques to solve an overdetermined linear system. These can be grouped into three main categories:

- error minimizing techniques,
- maximum likelihood techniques,
- robust nonlinear techniques.

The error minimizing (ErrMin) techniques entail formulating a quadratic error function to be minimized. Least squares (LS) is the most popular and often used technique. However, total least squares (TLS) and constrained total least squares (CTLS) methods are becoming increasingly popular and show remarkable improvement in several situations. The maximum likelihood (ML) estimator tries to obtain parameters with the maximum probability given the data set and the a priori distribution. It can be shown that for several noise distributions, LS and ML techniques yield similar solutions.

One of the significant problems with the ErrMin and ML techniques is their inability to handle outliers. Even a single data element with large error can perturb the estimates significantly. Several robust techniques to handle outliers have been proposed recently,

which can handle even up to 50% outliers. However, the computational requirements become severe.

6.1. Least squares estimator

Least squares provides an easy way to solve an overdetermined system of equations. We wish to find $\mathcal{B}_1$ such that the norm

$$\| M\mathcal{B}_1 \|_2$$

is minimized. A trivial solution to the above problem is given by setting $\mathcal{B}_1$ to zero, resulting in zero minimum error. Therefore, we need to impose a constraint such that a non trivial solution is obtained. This constraint can be imposed in two ways; (1) by setting t_3 (or any other parameter) to unity, or (2) requiring that the length of $\mathcal{B}_1$ is unity. The two constraints are equivalent unless t_3 is zero. The second constraint however is quadratic in nature and for the sake of simplicity we choose the first constraint. However, a closed form solution for the basic parameter with the second constraint can be obtained as well and is presented later. The problem is now reduced to recovering $\mathcal{B}_1$ such that

$$\| M\mathcal{B}_1 - \boldsymbol{w} \|_2$$

is minimized. Here, with a slight abuse of notation we now define M as a $XY \times 5$ matrix consisting of only the first five columns of M in (13), $\mathcal{B}_1$ is (a, c, d, t_1, t_2), and $\boldsymbol{w}$ is a vector of size XY consisting of w_x. The LS solution is given by

$$M^{\mathrm{T}} M\mathcal{B}_1 = M^{\mathrm{T}}\boldsymbol{w},$$

which in the continuous case reduces to:

$$\begin{bmatrix} 4\int x^2 & -2\int xy & -2\int x & -2\int xv_x & 2\int xu_x \\ -2\int xy & \int y^2 & \int y & \int yv_x & -\int yu_x \\ -2\int x & \int y & \int 1 & \int v_x & -\int u_x \\ -2\int xv_x & \int yv_x & \int v_x & \int v_x^2 & -\int u_x v_x \\ 2\int xu_x & -\int yu_x & -\int u_x & -\int u_x v_x & \int u_x^2 \end{bmatrix} \mathcal{B}_1 = \begin{bmatrix} -2\int x(xv - yu)_x \\ \int y(xv - yu)_x \\ \int (xv - yu)_x \\ \int v_x(xv - yu)_x \\ -\int u_x(xv - yu)_x \end{bmatrix},$$

where the integrations are over the image plane R. We can replace several terms containing the motion field derivatives by the motion field by invoking Green's theorem in a plane [20]. We have:

$$\int xv_x = \int_C xv\,\mathrm{d}y - \int v,$$

$$\int yv_x = \int_C yv\,\mathrm{d}y,$$

$$\int v_x = \int_C v\,\mathrm{d}y,$$

$$\int xu_x = \int_C xu\,\mathrm{d}y - \int u,$$

$$\int yu_x = \int_C yu\,\mathrm{d}y,$$

$$\int u_x = \int_C u\,\mathrm{d}y,$$

$$\int x(xv - yu)_x = \int_C x(xv - yu)\,\mathrm{d}y - \int (xv - yu),$$

$$\int y(xv - yu)_x = \int_C y(xv - yu)\,\mathrm{d}y,$$

$$\int (xv - yu)_x = \int_C (xv - yu)\,\mathrm{d}y,$$

where C is the boundary of the image plane R and all the integrals are over the image plane unless otherwise indicated.

6.2. *Bias in the least squares estimate*

Least squares offers a simple technique to estimate the basic parameters. However, it suffers from a severe problem; LS is asymptotically biased.

For an overdetermined system $M\mathcal{B}_1 = \boldsymbol{w}$ in n variables, least squares gives an unbiased estimate only if matrix M is noise free, i.e., all the measurement errors are in $\boldsymbol{w}$. For our problem, the measurement errors affect both M and $\boldsymbol{w}$. That is, the problem that is to be solved is

$$(M_o + \Delta M)\mathcal{B}_1 = \boldsymbol{w}_o + \Delta \boldsymbol{w},$$

where $[M_o; \boldsymbol{w}_o]$ is the true data, $[\Delta M; \Delta \boldsymbol{w}]$ is the error in the data, and $\mathcal{B}_{1o}$ is the actual solution. If the measurement error has zero mean it is easily demonstrated [33, Lemma 30] that under quite general conditions

$$\mathcal{B}_1 = (M^{\mathrm{T}}M)^{-1}M^{\mathrm{T}}\boldsymbol{w}$$

does not converge to $\mathcal{B}_{1o}$ but ("plim" means probability limit):

$$\mathrm{plim}_{m\to\infty}\,\mathcal{B}_1 = \left[I_n - \left(\frac{1}{m}M^{\mathrm{T}}{}_oM_o + C_a\right)^{-1}C_a\right]\left[\mathcal{B}_{1o} + \left(\frac{1}{m}M^{\mathrm{T}}{}_oM_o\right)^{-1}C_{ab}\right],$$

where the error covariance matrix C of $[M; w]$ is being decomposed as

$$\begin{bmatrix} C_a & C_{ab} \\ C^{\mathrm{T}}{}_{ab} & \sigma^2 \end{bmatrix}$$

(C_a is $n \times n$ and C_{ab} is $n \times 1$).

Assuming that the errors are uncorrelated across rows and columns of data $[M; w]$, the bias in the solution converges to

$$\mathcal{B}_1 - \mathcal{B}_{1o} = -\sigma^2 \left(\mathrm{plim}_{m \to \infty} \frac{1}{m} M^{\mathrm{T}} M \right)^{-1} \mathcal{B}_{1o}.$$

This implies that least squares is asymptotically biased. The size of bias increases with error and the condition number of matrix M. Also note that the bias always has the effect of underestimating the absolute value of the parameters. That is, the bias will move the focus of expansion towards the optical axis. This kind of bias has been reported earlier by several authors and we observered it as well. The results of experiments with LS appear in [7,8].

6.3. Total least squares estimator

Least squares yields a biased estimate as it fails to recognize the error in the left-hand side matrix M. In case the errors are equally distributed through the entire measurement matrix $[M; w]$, the TLS [16] algorithm can be used to solve the overdetermined system of equations. The basic principle of the TLS is to find $[\Delta M; \Delta w]$ such that

$$(M - \Delta M)\mathcal{B}_1 = w - \Delta w$$

has an exact solution and that $\| \Delta M; \Delta w \|_2$ is minimized. Compare this to least squares, which entails finding Δw such that

$$M\mathcal{B}_1 = w - \Delta w$$

has an exact solution and $\| \Delta w \|_2$ is minimized. A closed form solution can be derived for the TLS problem and is given by:

$$\mathcal{B}_1 = (M^{\mathrm{T}} M - \lambda^2 I_n)^{-1} M^{\mathrm{T}} w, \tag{16}$$

where λ is the smallest singular value of the matrix $[M; w]$. It has been proved in [16] that under very general conditions, the TLS yields an unbiased estimate with the residual error at least as small as the LS solution. Several other results regarding the stability, performance and computational requirements have been obtained.

6.4. Constrained total least squares estimator

The solution presented in the previous section yields an unbiased estimate when the noise is uniformly (uncorrelated, with covariance matrix I) distributed across matrix $[M; w]$. However, for the system of equations (13), the noise in the first three data

elements $(2x, -y, -1)$ is zero as the image coordinates are known perfectly (it is easy to see that the analysis is similar even if there is error in the optical axis) and the noise is only in the other three coefficients. In this section we present a technique to obtain an unbiased solution for the system of equations (13) under the assumption that the noise in the other three data elements $(-v_x, u_x, w_x)$ is similarly distributed and is uncorrelated.[2]

The system of equations (13) can be rewritten as:

$$[M_1; M_2]\mathcal{B}_1 = \boldsymbol{w},$$

where the $XY \times 3$ matrix M_1 is noise free. The problem is to find $[\Delta M_2; \Delta \boldsymbol{w}]$ such that

$$[M_1; M_2 - \Delta M_2]\mathcal{B}_1 = \boldsymbol{w} - \Delta \boldsymbol{w}$$

has an exact solution and $\| \Delta M_2; \Delta \boldsymbol{w} \|_2$ is minimized. A detailed solution of the above system of equations known as *mixed LS-TLS* is given in [16]. We give a brief description here. First, a 3 Householder transformation Q on the matrix $[M_1; M_2; \boldsymbol{w}]$ is performed to obtain:

$$Q^{\mathrm{T}}[M_1; M_2; b] = \begin{bmatrix} R_{11} & R_{12} & R_{1b} \\ 0 & R_{22} & R_{2b} \end{bmatrix},$$

where R_{11} is a 3×3 upper triangular matrix. The closed form solution for $\mathcal{B}_1$ is now given by:

$$\mathcal{B}_1 = \left(M^{\mathrm{T}}M - \lambda^2 \begin{bmatrix} 0 & 0 \\ 0 & I_2 \end{bmatrix} \right)^{-1} M^{\mathrm{T}}\boldsymbol{w},$$

where I_2 is a 2×2 identity matrix, and λ is the smallest singular value of the matrix $[R_{22}; R_{2b}]$. The proof regarding the unbiased nature of the solution is given in [16].

6.5. *Unmodeled noise structure*

The above solution with CTLS was developed under the assumption of uniform uncorrelated noise in the measurement vector (v_x, u_x, w_x). However, it is not satisfied even for very simple noise models. This results in biases in the estimates. If we assume that the noise in (v_x, u_x) is spatially uncorrelated and is zero mean Gaussian, the covariance matrix of the measurement vector C reduces to

$$C = \sigma^2 \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ x & y & x^2 + y^2 \end{bmatrix},$$

a nondiagonal matrix. Moreover, C is a function of image coordinate (x, y) and hence is different across rows of matrix M. A simple variable substitution can be used to reduce the covariance matrix to a diagonal form if it is the same for all rows. But,

[2] This is obviously not true and we observe some bias in the solution.

reducing the nondiagonal varying covariance matrix to a diagonal form is nontrivial. Some techniques to reduce the ill effect of varying nondiagonal covariance matrix are described in [18] and can be used.

Some other noise structures which are not considered but will effect the motion parameter estimates are the following:

- We assumed the Gaussian noise in the derivatives of the flow field. However, what is given to us is the motion field and its derivatives have to be computed using numerical differentiation. Therefore, while modeling the noise in the derivatives this must be taken into consideration.
- The noise in the flow field is almost always spatially correlated with significant outliers.
- There may be errors in the location of the optical axis.

Remark. We imposed the constraint, $t_3 = 1$, to obtain a nontrivial solution. Instead, if we require $\| \mathcal{B}_1 \| = 1$, the LS and the TLS yield identical solutions. This solution is given by the singular vector associated with the smallest singular value of the matrix $[M; w]$. The CTLS solution is given by the eigenvector associated with the largest eigenvalue of matrix $([M; w]^T [M; w])^{-1} P$, where P is a 6×6 matrix given by

$$P = \begin{bmatrix} 0 & 0 \\ 0 & I_3 \end{bmatrix}.$$

7. Robustness of motion parameter

In the last three sections we developed the solution for estimating the motion parameters from the motion field. We proved the correctness and the completeness of our approach when the data is noise free. The results guarantee that whenever the motion field renders a unique motion and shape interpretation, the correct motion parameter is obtained. We then developed the solution for the noisy motion field. In this section we investigate the stability of the motion parameter. In particular, we want to obtain an upper bound on the error in the motion parameter as noise is introduced in the motion field.

The motion parameter is recovered by first estimating the basic parameters and then recovering the motion parameter from them. Thus, the stability of the whole system is guaranteed only when both the steps involved are stable. We first consider the stability of the basic parameters and hence the translation parameters. The stability of the rotation—which is recovered from the basic parameters—is discussed subsequently.

Recovering the basic parameters from the motion field as well as recovering motion from the basic parameters entails solving a linear system of equations. A linear system of equations is easy to analyze for noise sensitivity. In particular, we need to address the following two issues:

- How stable is the system of equations (13)? We need an estimate of the error in the recovered basic parameters as a function of the error in the input data, or at least an upper bound on the error in the basic parameters with respect to the error in the input data.

- How much noise do we expect in the BaP constraints? Eqs. (10) and (11) relate the motion field, its derivatives, image coordinates and the basic parameters. What is then needed is an estimate of the noise level we expect in the motion field, its derivatives and the image coordinates.

7.1. Stability of the basic parameters

In this section we investigate the error introduced in the estimate of $\mathcal{B}_1$ as the error $[\Delta M; \Delta \boldsymbol{w}]$ is introduced in the data matrix $[M; \boldsymbol{w}]$. For the TLS solution (16), a *nonachievable* upper bound on the relative error in the estimate $\mathcal{B}_1$ can be obtained under reasonable restrictions on the noise and is given by [16, pp. 200–207]:

$$\frac{\|\Delta \mathcal{B}_1\|}{\|\mathcal{B}_1\|} < \frac{\kappa(M)}{1 - \kappa(M)(\|\Delta M\|/\|M\|)} \left(\frac{\|\Delta \boldsymbol{w}\|}{\|\boldsymbol{w}\|} + \frac{\|\Delta M\|}{\|M\|} \right), \tag{17}$$

where $\kappa(M)$ is the generalized condition number of the nonsquare matrix M and is defined as

$$\kappa(M) = \|M\|_2 \, \|M^{\dagger}\|_2 = \frac{\sigma_1}{\sigma_5},$$

the ratio of the two extreme singular values of M.

The condition number essentially is a measure of linear dependence among the columns of a matrix. For a singular matrix the columns are linearly dependent and the condition number is infinity, whereas, for an orthonormal matrix, the condition number is 1. For our case, the problem is reduced to estimating the linear dependence among the higher order terms (of order 2 and more) of $(h(x,y)(x-t_1))_x$ and $(h(x,y)(y-t_2))_x$. It is easy to see that the two functions are highly dependent if $h(x,y)$ has only lower order terms in its power series expansion. As a matter of fact, a planar surface yields a linear $h(x,y)$, and the two functions are linearly dependent.

The estimate in inequality (17) states that if the matrix is well conditioned, that is, $\kappa(M)$ is not too large, then small changes in M and $\boldsymbol{w}$ produce correspondingly small changes in the solution $\mathcal{B}_1$. On the other hand, if M is ill conditioned, then small changes in M and $\boldsymbol{w}$ may produce large changes in $\mathcal{B}_1$.

Condition number computation involves computing the singular values of the matrix. Though we can obtain a closed form expression for the condition number of M, it is rather complicated and hence not too informative. However, some empirical analysis can be performed rather easily. In most of our experiments, we found the condition number of M to be large (around hundred thousand), though the system was not sensitive to noise (noise was added to the motion field and its derivatives separately).

As we mentioned, the upper bound on the relative error given above cannot be achieved by any noise distribution. We can obtain a tighter bound on the instability of a linear system by transforming the linear system, such that the resulting coefficient matrix has a lower condition number. Obtaining the original basic parameters from the transformed variables involves a matrix multiplication and hence this step does not increase the relative errors. We point out though that this transformation does not improve the actual stability of the system, rather the tighter error bounds imply the

overall stability of the system. However, since the computation is performed on a finite precision machine, some stability is gained as truncation error does not adversely effect the system.

The rationale behind transforming the linear system is to equilibrate the errors in different terms of the data matrix $[M; \boldsymbol{w}]$. Expression (17) is obtained for the maximum noise in matrix $[M; \boldsymbol{b}]$. If the relative error for the different data elements is the same, the absolute error in the small elements will be relatively smaller than that in the larger elements. The idea then is to transform the system such that all the elements of the data matrix have almost similar absolute noise. The linear system is transformed by post-multiplying it by another matrix H and by change of variable $\mathcal{B}_1 = H\hat{\mathcal{B}}_1$. Here one replaces the problem $M\mathcal{B}_1 = \boldsymbol{w}$ by the problem $\hat{M}\hat{\mathcal{B}}_1 = \boldsymbol{w}$, where $\hat{M} = MH$. The matrix H is a square matrix of the order of M. If H is diagonal, this transformation may be interpreted as a column scaling operation applied to M. If H is nonsingular, this transformation does not change the problem mathematically. However, unless H is orthonormal, the condition number of $\hat{M}$ will generally be different from that of M. We use the diagonal matrix H as a column scaling matrix to balance the Euclidean norms of the columns of M. It has been proved by van der Sluis [37] that with this choice of H, $\kappa(MH)$ does not exceed the minimal condition number obtainable by column scaling by more than a factor of $n^{1/2}$.

The overall relative error in $\hat{\mathcal{B}}_1$ is dependent on $\kappa(MH)$ and is given by an expression similar to (17). The *relative error* in $\mathcal{B}_1$ is the same as the relative error in $\hat{\mathcal{B}}_1$ as it is obtained from $\hat{\mathcal{B}}_1$ by multiplication with a constant matrix H (note that H is constant for a given problem).

We found that with column scaling, the error bounds were small (condition number of $\hat{M}$ is in the hundreds most of the times and at most a few thousand), if the image plane is not small and the surface is not very close to being a planar. Low condition number means the system (16) is stable against random perturbations in the input data (again, noise being added to the motion field and its derivatives separately). The following remarks are offered:

- The condition number of the matrix M in (13) is inversely proportional to the size of the image plane. For a very small region the condition number was large. This shows that recovering 3-D motion from a small patch is unstable and we need data from a significant spatial and temporal (to compute stable optical flow [7]) support for robustness.
- Another interesting observation is that the condition number of the system of equations is inversely proportional to the complexity of the surface in motion. As the surface became more complicated (by any measure, for example, as the contribution of the higher order terms in the Taylor series expansion increases) the system of equations gained stability. Though surprising, the reason for this is not difficult to find. As the surface becomes complicated, u_x and v_x become dissimilar from each other and from x and y. The result is that the BaP constraints at various image points become increasingly dissimilar and hence the stability.
- The condition number of M is minimum when the object is at the center of the image plane. That is, the best 3-D motion parameters are estimated if the object is in the center of the visual field.

- Several attempts have been made in the past to show that the 3-D motion recovery from the motion field is inherently unstable by showing that the condition number of the Jacobian matrix of a nonlinear system of equations is large. It is clear that the condition number is dependent on the field of view and the surface structure and no general remarks can be made regarding the general stability of the 3-D motion estimation problem.

In the above analysis we considered the noise which was added to the motion field and its derivatives separately. We ignored the fact that what is given is the motion field and the derivatives have to be computed via numerical differentiation. The noise in the motion field and its derivatives is related and more often than not, the noise in the motion field derivatives will be significantly more than that in the motion field. We address this issue in the next section.

7.2. *Stability of BaP constraint*

The BaP constraints relate the motion field, its spatial derivatives, and the image coordinates (x, y) with the basic parameters. The stability of the constraints is thus dependent on the stability of the motion field and its derivatives. What is given to us is the motion field and the derivatives have to be computed via numerical differentiation—an extremely unstable and ill-posed operation. We quote from [3]

> *It should be remembered that, as an approximation method, numerical differentiation is not stable, since small values of h (step) can lead to large errors. . . ., and in actual practice this operation is avoided whenever possible.*

Instability of the numerical differentiation amplifies the noise in the flow field; a small noise in the flow field can produce large errors in the derivatives. The problem is compounded by the fact that we will always have significant noise in the flow field (recall our discussion about the fact that the motion field cannot be recovered and only optical flow is recoverable from a sequence of images). The result is that significantly large errors are introduced in the motion field derivatives resulting in unreliable BaP constraints. Unreliable BaP constraints mean that though the system of equations is well conditioned, the constraints are too unreliable to yield any meaningful answer.

One of the ways to get around this problem is to use several image points and solve the overdetermined system. This can solve the problem for cases where the derivatives can be estimated with a small noise (say 10%). But, in some of our experiments, the noise was much more than the signal (derivative) itself. The CTLS solution of the overdetermined system also failed to give any meaningful answer in these cases.

Another way to get around this problem is to somehow compute the spatial derivatives of the flow field directly from the image sequence, rather than computing them from the flow field numerically. We have not come across any such effort and we believe that computing the derivatives of the flow field directly from the image sequence will be much more difficult and unstable than computing the flow field itself. The optical flow field is related to the spatial and temporal derivatives of the image intensity. The derivatives of the optical flow will therefore be related to the image sequence through the second order partials. When computing the first order derivatives of the discrete,

noisy image sequence is itself unstable, it is rather evident that obtaining second order derivatives robustly will be infeasible.

In the next section we show how to get around this problem. We reformulate the solution so as to obviate the need for the flow derivatives. The central idea is to use the available information in the spatial domain in a useful way.

7.3. Stability of the rotation vector

In the previous subsection we investigated the stability of the basic parameters and hence the translation vector. The rotation vector $\boldsymbol{\omega}$ is recovered from the basic parameters by solving a linear system of equations. In this section we discuss the stability of the rotation vector $\boldsymbol{\omega}$ recovered from the basic parameter $\mathcal{B}_1$.

The rotation vector $\boldsymbol{\omega}$ is recovered by solving

$$\begin{bmatrix} 0 & t_2 & t_3 \\ t_2 & t_1 & 0 \\ t_3 & 0 & t_1 \end{bmatrix} \boldsymbol{\omega} = \begin{bmatrix} a \\ c \\ d \end{bmatrix}. \tag{18}$$

The stability of the above linear system of equations is given by the condition number of the matrix. It is easy to see that the condition number of the above matrix is large only if t_1 is small or both t_2 and t_3 are small; with t_1 zero or both t_2 and t_3 zero, the matrix is singular. The stability of the constraints for the rotation vector is dependent on the stability of the basic parameters. If $\mathcal{B}_1$ can be recovered reliably, the error in the constraints is small and bounded. Therefore, if $\mathcal{B}_1$ can be recovered reliably and t_1 and $t_2^2 + t_3^2$ are not small then the rotation parameter can be estimated robustly.

8. Stabilizing the BaP constraints

Numerical derivative computation is an unstable operation and any noise in the input data is amplified. We mentioned earlier that what is available to us is optical flow—a very crude approximation to the motion field. Hence, it becomes all the more important for our purposes to avoid numerical differentiation. Below we present a way to eliminate the flow derivatives from the constraints (10) and (11). The central idea is to use non-local constraints—constraints which use information from a significant spatio-temporal domain—for stability. We first expand on the idea of nonlocal constraints and then present a way to stabilize the BaP constraints by eliminating the motion field derivatives from the formulation.

8.1. Nonlocal constraint

Several vision problems have been formulated as solving a system of equations or minimizing an energy function. The energy function or the system of equations consist of the domain constraints and the regularization terms. The domain constraints capture the idea of the domain whereas the regularization terms capture the heuristic knowledge

about the physical world. Local constraints, which capture the domain knowledge have been used extensively to solve a variety of problems in computer vision. Local constraints relate various known and unknown quantities at a point in the space-time continuum. Some examples of the systems based on local constraints are given below:

- *Optical flow computation*: Optical flow constraint

$$I_x u + I_y v + I_t = 0$$

(where I_x, I_y, I_t are the image intensity gradients and (u, v) the optical flow) relates the instantaneous optical flow velocity with the various partial derivatives of the image intensity at a point in time and space. The optical flow constraint, along with the smoothness constraint, which imposes the assumption about the smoothness of the optical flow field, are used to estimate the optical flow field from the local constraints only.

- *Shape from shading*: For recovering shape from shading [14], local constraints like

$$\boldsymbol{n} \cdot \boldsymbol{l} = I$$

(where $\boldsymbol{n}$ is the surface normal, $\boldsymbol{l}$ is lighting direction and I is the image intensity at a point in space and time) capture the relationship between the reflectance and the image intensity at a point. Constraints like these, along with the regularization terms which stipulate that the normals change slowly in a local neighborhood are used to obtain the 3-D shape information from shading.

There are several reasons why systems based on local constraints became so popular. Some, albeit not all are:

- Local constraints along with the additional heuristic knowledge about the physical world yield a highly overdetermined and a well-conditioned system.
- Approaches based on local constraints lead to a neat, structured and elegant mathematical formulation.
- Most important of all is the ease with which the local constraints based systems can be implemented on a neural network. It was *assumed* that the biological visual system—a neural network—must be doing local processing, and hence a system performing only local operations was preferred.

In spite of their neat mathematical structure, these local constraints suffer mainly due to an overload of "work". By this we mean that these constraints try to do too much locally. In this noise corrupted world there is not enough reliable information to allow the local solution of the problem in its most general form. The result was that though enough constraints were obtained, the constraints themselves were extremely noise sensitive. The problem of noise handling was not tackled at the constraint level, but was deferred till very late and was passed on to the least squares to provide noise immunity to the system. Least squares handles noise only if a random noise has corrupted the system and hence fails to alleviate much of the problem in the presence of highly correlated and structured noise.

If we hope to have any realistic chance to solve the difficult problems in computer vision we have to use robust constraints. The main reason for the instability of the constraints is that they use only local information and hence a way to achieve stability

is to utilize the information available in time and space in an effective way. We must use constraints which invoke the support of a significant spatio-temporal domain to achieve stability, so as to tolerate a realistic level of noise.

8.2. *Elimination of flow derivatives*

The BaP constraints are local constraints and use information only at a point in time and space. They are unstable as they use the local variations of the motion field, which are impossible to estimate in a reliable way. The idea then is to use the large amount of available information to impart stability to BaP constraints. Integrating (10) and (11) over a region R, with boundary C, in the image plane we get

$$-2a\int_R x + c\int_R y + d\int_R 1 + t_1\int_R v_x - t_2\int_R u_x - \int_R xv_x + v - yu_x = 0, \tag{19}$$

$$-2b\int_R y + c\int_R x + e\int_R 1 + t_1\int_R v_y - t_2\int_R u_y - \int_R xv_y - u - yu_y = 0. \tag{20}$$

If the motion field (u, v) is continuous and has continuous partial derivatives in region R, from Green's theorem in the plane [20], we have

$$\int_R v_x\,dx\,dy = \oint_C v\,dy, \qquad \int_R v_y\,dx\,dy = -\oint_C v\,dx,$$

$$\int_R u_x\,dx\,dy = \oint_C u\,dy, \qquad \int_R u_y\,dx\,dy = -\oint_C u\,dx,$$

$$\int_R (xv_x + v - yu_x)\,dx\,dy = \oint_C (xv - yu)\,dy,$$

$$\int_R (xv_y - u - yu_y)\,dx\,dy = -\oint_C (xv - yu)\,dx.$$

Substituting the above in Eqs. (19) and (20) we get

$$-2a\int_R x + c\int_R y + d\int_R 1 + t_1\oint_C v\,dy - t_2\oint_C u\,dy = \oint_C (xv - yu)\,dy, \tag{21}$$

$$-2b\int_R y + c\int_R x + e\int_R 1 - t_1\oint_C v\,dx + t_2\oint_C u\,dx = -\oint_C (xv - yu)\,dx, \tag{22}$$

as linear constraints for the basic parameters in terms of various integrals of the flow field. If the flow field is known at the boundaries of five regions, the BaPs can be obtained by solving a linear system of equations as in Section 4.3.

To recapitulate, we have been able to eliminate the instability in the local constraints (10) and (11) by using the motion field in a region. The local constraints (10) and (11) are unstable due to the need to obtain the first order variations in the flow field. When the

flow field itself is noisy, the task of finding local variations becomes extremely difficult. By seeking large spatial support, with the additional assumption that the underlying surface is smooth and differentiable, we are able to eliminate the need for derivatives. To emphasize the advantage of substituting the flow derivatives by various integrals of the flow field, we quote from [3] again

> ..., *consequently, the numerical procedure (integration) is stable as* $h \to 0$. *Recall that this was not true in the case of numerical differentiation procedures* ...

Remarks. (1) The additional condition that the motion field be continuous along with continuous partial derivatives is not necessary to obtain the constraints (21) and (22). The same constraints can be obtained with a much relaxed requirement, that the region R must be moving with the same 3-D velocity. In the case of a static environment and moving camera, this condition is true for the entire image. In case of multiple objects in motion, we require that the entire region R be on a single object. Differencing Eq. (4) at points (x_1, y) and (x_2, y) we get

$$a\bar{x}\delta_x - cy\delta_x - d\delta_x - t_1(v(x_1, y) - v(x_2, y)) + t_2(u(x_1, y) - u(x_2, y))$$
$$= y(u(x_1, y) - u(x_2, y)) - (x_1 v(x_1, y) - x_2 v(x_2, y)),$$

where $\delta_x = x_1 - x_2$ and $\bar{x}$ is the mean of x_1 and x_2. Integrating the above equation along a contour C of the region R we get the constraint (19). Similarly, constraint (20) can be obtained without the need for imposing the continuity and differentiability on the motion field.

(2) At this point it is instructive to look at the algorithm proposed by Jepson and Heegar [17], which is very similar to our approach; in several ways our work extends their results. We derive the constraint obtained by Jepson and Heegar in a different way. For a patch of $N \times N$, Eq. (3) can be written as

$$Pq + Rt = 0,$$

where P is an $N^2 \times 6$ matrix consisting of $(x^2, y^2, -xy, -x, -y, 1)$, $q = (a, b, c, d, e, f)$, and matrix R is an $N^2 \times 3$ matrix consisting of $(-v, u, w)$. Multiplying the above equation with $p(x, y)$, such that $p(x, y)$ is in the null space of P yields

$$p(x, y)Rt = 0,$$

or

$$t_1 \int p(x, y)u(x, y) + t_2 \int p(x, y)v(x, y) + t_3 \int p(x, y)w(x, y) = 0,$$

the constraint obtained in [17]. The kernel $p(x, y)$ requires that

$$\int p = \int xp = \int yp = \int x^2 p = \int y^2 p = \int xyp = 0,$$

where all the integrals are over the $N \times N$ image patch. From the description in [18] it seems that the authors have used a Hermite polynomial for the function p, although

they do not specify the order of the polynomial. The lowest order Hermite polynomial that will satisfy the above constraints is of order three. Computing the convolutions of the flow field with this p is akin to computing the weighted third order difference of the flow field in a small neighborhood (N is 31). We saw earlier that computing the first order difference of the flow field in a small neighborhood is extremely noisy. Therefore, it is easy to see that the third order differences would be highly noise sensitive.

9. Combining motion parameters

In the previous sections a CTLS solution was obtained for the basic parameter $\mathcal{B}_1$. We have used only one of the BaP constraints so far. The other constraint (22) can be used to reduce the effect of the noise even further. One way would be to define a error function for the two constraints and minimize them together. Unfortunately, the error function does not yield a closed form solution and iterative techniques have to be used to solve the nonlinear system.

Instead, we use CTLS to estimate the basic parameters $\mathcal{B}_1$ and $\mathcal{B}_2$ separately. For noiseless data the two solutions will be identical (we proved this earlier) but noise in the motion field may lead to slightly different solutions. The two sets of motion parameters thus estimated can be combined to yield a better estimate of the motion parameter. One way would be to take the average of the two estimates, though a better technique would be to select the motion parameter which better fits the given motion field. The measure of fitness of the solution can be easily estimated by computing how well the motion parameters satisfy Eq. (4). Specifically, if $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ are the two solutions, we estimate (function $\mathcal{P}$ is defined by Eq. (14)):

$$E_1 = \int_R \mathcal{P}^2(x, y, \boldsymbol{\alpha}_1)\, \mathrm{d}R, \qquad E_2 = \int_R \mathcal{P}^2(x, y, \boldsymbol{\alpha}_2)\, \mathrm{d}R,$$

and select the motion parameters based on one which yields the smaller error. A further post-processing may be performed (if the residuals E_1 and E_2 are large) which minimizes the above energy function for motion parameter $\boldsymbol{\alpha}$ using iterative techniques. The solution obtained from basic parameters is used as the starting state. This post-processing improved the results slightly for the real images in a few iterations (less than 5).

10. Discrete case

The algorithm and the analysis in the preceding sections was for the continuous flow field. Though essential for analyzing the behavior of the algorithm, it should be adapted so as be useful for the discrete motion field, since what is given to us is a sparse flow field at discrete points.

We use numerical integration to obtain the coefficients in Eqs. (21) and (22). Though we can use a more sophisticated numerical integration scheme, Simpson's composite integration algorithm or even the trapezoidal composite integration technique is sufficient for our purposes. The errors introduced by the trapezoidal rule are of the order of the

Algorithm Compute-motion(u, v).

Input: noisy, sparse, discrete flow fields.

Output: translation and rotation vector.

Step 1: Choose regions such that reliable image flow is available at most points on the boundary which are not parallel to the x, y axis.

Step 2: Compute M and $\boldsymbol{w}$.

Step 3: Solve the CTLS problem via solving

$$M\mathcal{B}_1 = \boldsymbol{w}.$$

Step 4: Compute rotation and translation from $\mathcal{B}_1$.

Step 5: Compute M and $\boldsymbol{w}$ for $\mathcal{B}_2$.

Step 6: Solve the CTLS problem via solving

$$M\mathcal{B}_2 = \boldsymbol{w}.$$

Step 7: Compute rotation and translation from $\mathcal{B}_2$.

Step 8: Compute the error in (4) for the motion parameters estimated from $\mathcal{B}_1$ and $\mathcal{B}_2$.

Step 9: Select the motion parameter which yields the smaller error.

Fig. 1. Algorithm for computing motion parameters.

third derivative of the motion field, whereas Simpson's technique introduces errors of the order of the fourth derivative. We use Simpson's composite integration technique which involves obtaining the weighted sum along the contour.

We would like to draw attention to the fact that by computing integrals this way we assume that the surface is locally planar. To see this, just consider that Simpson's rule fits a second order polynomial between three points and second order flow is generated only if the surface is planar [38]. Also note that we do not need the flow to be dense for our scheme to work. We just have to define regions such that flow field is available for most points on the segments of the boundary which are not parallel to the x, y axis.

A top level sketch of the algorithm is given in Fig. 1.

10.1. Data requirements

As we mentioned earlier, the motion field is an abstract concept and cannot be computed from an image sequence. What is computable is optical flow—an approximation of the motion field. Moreover, reliable optical flow can be estimated only at few points. In this context, it is important that the algorithm be useful even for a very sparse optical flow field. In this section we analyze the data requirements for the algorithm.

Eq. (21) is a constraint in five variables. Therefore, for a unique solution, the BaP constraint should be obtained for five different regions. The issue is: how much data is needed for each constraint. As is evident from Eq. (21), the motion field is required only at the boundaries of the selected regions. Moreover, we do not require the data over the entire boundary. We can select regions such that the data is required for some segments of the boundary only.

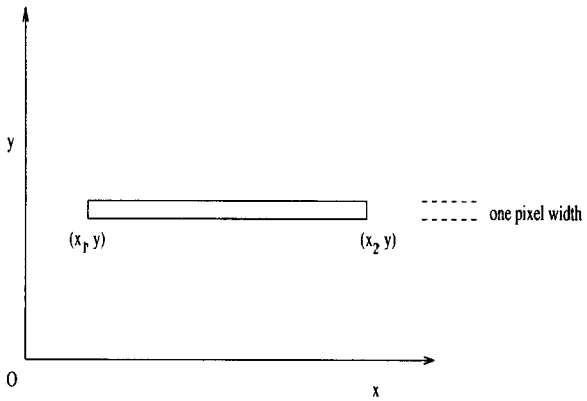

Fig. 2. Region such that data is required only for two points.

The motion field in Eq. (22) is integrated along the y axis. Therefore, the segments of C, along which the y coordinate does change, always give zero for the integral. Thus, if the regions are chosen such that the significant portion of the boundary is parallel to the x axis, the motion field will be needed only along small segments.

The minimum data is required if a rectangular region is selected with the width in the y dimension being a single pixel. The rectangular region is oriented along the coordinate axes and the width along the x axis can be suited according to the given data. For the region shown in Fig. 2, it is easy to see that Eq. (22) reduces to:

$$\begin{aligned} &-2a\delta\bar{x} + cy\delta + d\delta + t_1(v_2 - v_1) - t_2(u_2 - u_1) \\ &\quad = x_2v_2 - x_1v_1 - y(u_2 - u_1) \end{aligned} \tag{23}$$

which requires the motion field at exactly two points (x_1, y) and (x_2, y) ($\delta = x_2 - x_1$ and $\bar{x} = 0.5(x_1 + x_2)$). This means that the algorithm is applicable even if the optical flow field is extremely sparse. It is easy to see that a unique motion parameter can be obtained, given the motion field and its derivatives at five points. Moreover, if five of these points lie on a straight line and two other points lie on a line parallel to this line (but not the same) then the motion field data at these seven points is sufficient for a unique solution. A unique solution can also be obtained from six points, if they lie on a straight line.

Remark. Note that the above constraint (23) can also be obtained if Eq. (10) was integrated along the x axis from point (x_1, y) to (x_2, y).

11. Implementation issues

The algorithm works for sparse as well as dense flow fields. In case of a dense flow field, with flow at each point equally reliable, a closed form solution can be obtained by selecting the regions of predefined size and shape. In this case various integrals of the flow field will have to be computed which can be easily done in parallel or on a

neural network. The resulting algorithm is extremely fast even on a Sparc Station and takes less than a few seconds for a 512×512 image.

In case when the flow field is sparse or has a reliability factor associated with each point, choosing the regions become slightly involved. As is evident, we need only five regions to obtain a unique motion parameter. Also, we need the flow field only on the boundaries of these regions. We have two contrasting demands here:

- More regions lead to the stability of the system.
- For each region a reliable estimate of the flow field is required at the boundaries.

For stability we need a small condition number which can be achieved by having a large number of regions. However, we need reliable motion field estimates for each region. A good compromise is to fix a bound on the condition number and select the regions with the most reliable flow field till the required stability is achieved. This results in using the best available information in the flow field to achieve the required stability. As we mentioned before, the regions can be overlapping and even one region can be completely inside another one. This suggests that if we have an extremely reliable flow along a contour, we should select several regions having the contour as part of their boundary.

In case the flow field is sparse or has an associated reliability function, the regions have to be selected depending on the data and hence no closed form solution can be obtained a priori. In these cases, the constraints (21) and (22) have to be obtained separately for each region and motion can be obtained via least squares. The resulting algorithm is significantly slower than the one for predefined regions as we need to achieve the required stability by adding new regions incrementely. A lot of book-keeping is required to avoid computing the same partial integrals over and over again. With some book-keeping, our implementation takes about a minute for a 512×512 image on a Sparc Station.

12. Results and discussion

We implemented the algorithm described in Sections 6, 8, 9, and 10 with the translation scaling $|t| = 1$. Since t_3 is nonzero for all the experiments, for the ease of reporting we scale the results such that t_3 is unity. The algorithm was tested extensively for a simulated motion field and for optical flow obtained from a sequence of images. Both real and simulated image sequences were experimented with. We first describe the experiments with the simulated motion field and then the experiments with the computed optical flow field.

12.1. Simulated motion field

The motivation for experimenting with the simulated motion field was to test the algorithm for the noise tolerance and to assess the effects of varying grid sizes, spatial resolution, field of view, and the type of motion. Working with the simulated motion field allowed for an easier control of the above parameters, which would be difficult to control precisely for the optical flow obtained from an image sequence. The motion field

Table 1
Covariance matrix ($\times 10^{-4}$) for Experiment 1 with last noise level

0.000559	−0.000136	−0.000461	0.091938	−0.004281
−0.000136	0.000317	0.000109	−0.064692	0.004835
−0.000461	0.000109	0.000777	−0.120080	0.052371
0.091938	−0.064692	−0.120080	47.176479	−1.651347
−0.004281	0.004835	0.052371	−1.651347	9.633154

was generated for spheres, ellipsoids and other regular surfaces undergoing a general motion. Various noise models and varying noise levels were tried, the motivation being to test the stability of the algorithm in the presence of different kinds of structured noise. We experimented with the following noise models:

- Uniform noise in the range $[-\sigma, \sigma]$.
- Gaussian noise with zero mean and specified variance.
- Gaussian noise followed by global smoothing in the spatial domain.
- Gaussian noise followed by fitting a model to the flow field in a small region. We experimented with the constant and the linear flow field models. The model was fitted to regions varying from 14 to 20 pixels in size.

We report the results of our experiments with a rigid ellipsoid in motion. The motion field is generated analytically as the motion and the depth map are known. The noise model used is the fourth listed above and is given as

$$u(x,y) = u(x,y) + G(0.0, p|u(x,y)|),$$

$$v(x,y) = v(x,y) + G(0.0, p|v(x,y)|),$$

where G(mean, variance) is a Gaussian random number generator with a specified mean and variance and p is the specified fractional noise level. The two columns under "Noise" in Tables 2–7 report the percent noise before and after the model fitting. From the "After Fit" column, we note that though fitting a model reduces the noise level it makes the noise correlated, which is closer to the kind of noise we expect in the computed optical flow field. In the next four columns we report the average translation and rotation found over twenty trials along with the respective standard deviation.

Experiment 1. The objective of this experiment was to test the algorithm for noise tolerance. The image size is 595 × 595 pixels and the resolution is 512 pixels per unit focal length; the field of view is 60 degrees. Maximum and mean flow velocities are 5.79 pixels and 2.14 pixels, respectively. The regions are 161 × 161 pixels in size. The average motion parameters and the variance are shown in Table 2. The covariance matrix of the estimate for (A, B, C, t_1, t_2) is shown in Table 1 for the last noise level. Note that the algorithm is quite stable for even a large amount of noise (the noise level is more than a magnitude greater than what has been tried for other 3-D motion estimation algorithms, e.g. [18] introduces 10–15% noise whereas we introduce 200% noise with comparable results) and the error in the motion parameter increases gradually with the noise in the motion field. Note the small bias in the estimates of the motion parameters which can be attributed to the unmodeled correlation between u_x, v_x, w_x and the spatial correlation of noise.

Table 2
Results for Experiment 1

Noise		Translation				Rotation					
Before	After	True		Standard		True $\times 10^{-3}$			Standard		
fit	fit	0.80	0.60	deviation		0.00	3.20	−5.30	deviation		
as %	as %	Estimate		$\times 10^{-2}$		Estimate $\times 10^{-3}$			$\times 10^{-5}$		
5.0	1.0	0.80	0.60	0.17	0.22	0.00	3.20	−5.30	0.51	0.57	1.16
20.0	3.2	0.80	0.59	0.65	0.66	−0.01	3.20	−5.33	1.77	2.67	3.31
35.0	7.6	0.80	0.59	1.67	1.16	−0.02	3.20	−5.32	2.73	5.32	4.38
50.0	10.1	0.80	0.58	2.31	1.73	−0.02	3.21	−5.40	4.33	7.50	6.46
70.0	14.1	0.81	0.55	2.84	2.54	−0.08	3.18	−5.51	10.88	10.24	13.31
100.0	19.2	0.79	0.51	4.52	3.35	−0.20	3.20	−5.59	14.61	15.66	18.27
200.0	32.2	0.76	0.40	6.87	4.10	−0.73	3.25	−5.74	23.65	17.80	27.87

Table 3
Results for Experiment 2

Noise		Translation				Rotation					
Before	After	True		Standard		True $\times 10^{-3}$			Standard		
fit	fit	0.80	0.60	deviation		0.00	3.20	−5.30	deviation		
as %	as %	Estimate		$\times 10^{-2}$		Estimate $\times 10^{-3}$			$\times 10^{-5}$		
5.0	1.4	0.80	0.60	0.38	0.32	0.00	3.20	−5.30	0.83	0.69	2.00
20.0	4.5	0.80	0.59	1.19	1.31	−0.01	3.21	−5.34	2.70	2.47	5.44
35.0	7.6	0.80	0.58	2.31	2.12	−0.01	3.21	−5.39	7.18	4.80	11.72
50.0	10.5	0.81	0.54	3.17	2.67	−0.02	3.19	−5.57	9.90	6.68	14.37
70.0	16.2	0.80	0.51	3.82	2.91	−0.07	3.23	−5.65	11.89	11.15	21.11
100.0	21.3	0.79	0.45	5.58	4.57	−0.09	3.32	−5.91	19.88	9.05	27.61

Experiment 2. The objective of this experiment was to assess the noise tolerance of the algorithm as we change the field of view of the camera. We have all the parameters the same as in Experiment 1, except for a smaller field of view. The FOV for this experiment is 45 degree (425 × 425 image) and the maximum flow velocity is 4.02 pixels and the average flow velocity is 2.0 pixels. The results are shown in Table 3. Compare the results with the results for Experiment 1. For the same noise level, the bias in the estimates is more with larger variances. The added stability of the algorithm for Experiment 1 can be easily attributed to the larger field of view.

Experiment 3. The objective of the experiment was to study the stability of the algorithm as the grid size was changed. The grid size in this experiment is 81 × 81 pixels, compared to 161 × 161 for Experiments 1 and 2. The field of view is 60 degree and is the same as in Experiment 1. The results are shown in Table 4. Compare these results to one obtained in Experiment 1. The bias in the estimates is larger with comparable variances. It is clear that the tolerance to noise is reduced.

Table 4
Results for Experiment 3

Noise		Translation				Rotation					
Before	After	True		Standard		True $\times 10^{-3}$			Standard		
fit	fit	0.80	0.60	deviation		0.00	3.20	−5.30	deviation		
as %	as %	Estimate		$\times 10^{-2}$		Estimate $\times 10^{-3}$			$\times 10^{-5}$		
5.0	1.0	0.80	0.60	0.17	0.15	−0.00	3.20	−5.30	0.47	0.38	0.73
20.0	3.2	0.80	0.58	0.62	0.51	−0.04	3.22	−5.35	2.08	1.87	1.76
35.0	7.3	0.79	0.55	1.10	1.38	−0.14	3.28	−5.40	5.29	3.39	5.77
50.0	10.4	0.78	0.51	1.39	1.30	−0.26	3.32	−5.50	8.72	6.22	8.32
70.0	14.6	0.75	0.46	2.78	2.61	−0.48	3.43	−5.50	10.76	8.14	12.62
100.0	21.1	0.72	0.40	3.56	2.89	−1.02	3.53	−5.37	19.47	11.82	14.54

Table 5
Results for Experiment 4

Noise		Translation				Rotation					
Before	After	True		Standard		True $\times 10^{-3}$			Standard		
fit	fit	0.80	0.60	deviation		0.00	3.20	−5.30	deviation		
as %	as %	Estimate		$\times 10^{-2}$		Estimate $\times 10^{-3}$			$\times 10^{-5}$		
5.0	1.0	0.80	0.60	0.14	0.16	−0.01	3.20	−5.31	0.43	0.40	0.74
20.0	3.2	0.79	0.54	0.50	0.61	−0.12	3.27	−5.44	2.29	1.42	1.91
35.0	7.3	0.76	0.46	0.78	0.76	−0.42	3.40	−5.48	4.87	2.89	3.99
50.0	10.4	0.70	0.38	2.31	1.46	−0.97	3.59	−5.31	10.59	7.28	7.81
70.0	14.6	0.47	0.30	3.42	1.38	−2.43	4.18	−4.49	22.91	9.31	11.85
100.0	20.7	0.20	0.24	3.78	1.21	−3.89	5.07	−4.11	39.23	14.68	10.13

Experiment 4. The objective of the experiment was to study the stability of the algorithm as the grid size was changed. The grid size in this experiment is 161×1 pixels, compared to 161×161 for Experiments 1 and 2 and 81×81 for Experiment 3. The grid size is such that motion field velocity at only two points is required (as in Eq. (23)) to obtain a constraint for the basic parameter. This is significant as a reliable flow field cannot be obtained over long contours for the real images. The field of view is 60 degrees and is the same as in Experiment 1. The results are shown in Table 5. Compare these results to those obtained in Experiment 1. The bias in this case is significantly more with larger variances.

Experiment 5. The objective of the experiment was to assess the performance of the algorithm in the presence of predominant rotational motion. The ellipsoid is rotating about the z axis at a rate of approximately 1 degree per frame. The rotation about the other two axes is also significant. The translational motion is only along the x and the z axis and induces only 10 percent of the image motion. The results are shown in Table 6.

Table 6
Results for Experiment 5

Noise		Translation				Rotation					
Before	After	True		Standard		True $\times 10^{-3}$			Standard		
fit	fit	0.50	0.00	deviation		5.00	−3.20	−15.30	deviation		
as %	as %	Estimate		$\times 10^{-2}$		Estimate $\times 10^{-3}$			$\times 10^{-5}$		
5.0	1.0	0.50	−0.00	0.56	0.35	5.00	−3.20	−15.30	0.43	1.37	0.66
20.0	3.7	0.48	−0.00	1.51	1.40	5.01	−3.22	−15.31	2.11	3.78	3.62
35.0	6.4	0.45	−0.01	3.46	2.13	5.04	−3.27	−15.34	3.66	10.54	5.49
50.0	9.2	0.43	−0.02	3.12	3.40	5.08	−3.30	−15.38	4.00	10.90	7.15
70.0	12.8	0.39	−0.07	2.92	4.23	5.11	−3.39	−15.48	7.39	16.32	8.80
100.0	18.4	0.37	−0.10	4.85	5.98	5.22	−3.41	−15.59	14.78	17.33	10.54

Table 7
Results for Experiment 6

Noise		Translation				Rotation					
Before	After	True		Standard		True $\times 10^{-3}$			Standard		
fit	fit	−3.00	−3.00	deviation		0.00	3.20	−15.30	deviation		
as %	as %	Estimate		$\times 10^{-2}$		Estimate $\times 10^{-3}$			$\times 10^{-5}$		
5.0	0.9	−3.01	−3.01	3.33	2.93	−0.01	3.21	−15.30	3.28	3.01	1.48
20.0	3.8	−3.01	−3.04	12.10	12.73	−0.18	3.37	−15.27	15.05	14.92	6.72
35.0	6.6	−2.82	−3.02	24.13	25.16	−0.48	3.73	−15.08	29.35	34.64	16.99
50.0	9.3	−2.70	−3.06	25.60	20.96	−0.85	4.07	−14.96	44.38	54.38	22.38
70.0	13.0	−2.55	−3.26	31.70	31.18	−1.54	5.31	−14.50	67.26	104.87	38.88
100.0	19.0	−2.21	−3.73	30.82	57.94	−3.51	9.36	−13.41	106.01	253.88	64.78

Experiment 6. The objective of the experiment was to assess the performance of the algorithm in the presence of significant rotational motion and inplane translation—the most difficult situation for motion estimation algorithms. The ellipsoid is rotating about the z axis at a rate of approximately 1 degree per frame. The translational motion along the x and the y axis is significant. The results are shown in Table 7.

Based on these and many other experiments performed with a simulated motion field, our general observations are:

- The algorithm is stable against random perturbation in the input data.
- Noise introduces bias. The bias does not always move the focus of expansion towards the optical axis as reported in [18]. As we mentioned earlier, the bias of moving FOE towards center is due to LS and is eliminated with TLS and CTLS. The remaining bias can be attributed to the correlated errors across the last three columns and different rows of the matrix M in Eq. (13).
- The bias increases with the noise level.
- A larger grid size enables the system to be tolerant to larger noise.
- Better resolution improves the noise tolerance.
- A larger field of view enables the system to be tolerant against larger noise.

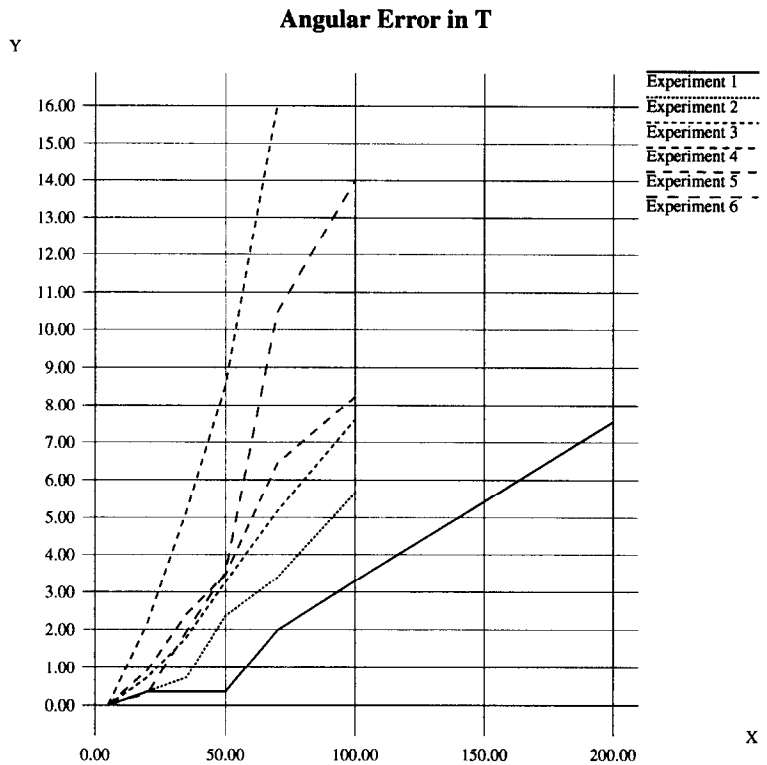

Fig. 3. Angular error in computation of T (in degree) for the six experiments with the simulated motion field. The x axis reports the percent noise in the motion field and the y axis reports the angular error in the computation of the translational vector.

12.2. Computed optical flow

In this section we describe our experiments with real image sequences, for what is available as input are the image sequences; the motion field, or its approximation the optical flow, has to be estimated from the image sequences. The motivation is to see how well the algorithm performs for a wide range of image sequences, and also, to assess the noise stability of the algorithm in the presence of a realistic noise—both in image sequences and in the computed optical flow field. Optical flow—an approximation to the motion field—was computed using the algorithm described in [9], which provides a dense flow field with an associated reliability. The same parameters are used for computing the optical flow as in [9].

The regions over which the BaP constraints were integrated were preselected and were 200 × 200 pixels in size. The regions were highly overlapped with a new region starting every 2nd pixel. The image size varied from 400 × 400 pixels for simulated imagery to 574 × 652 for the real image sequence.

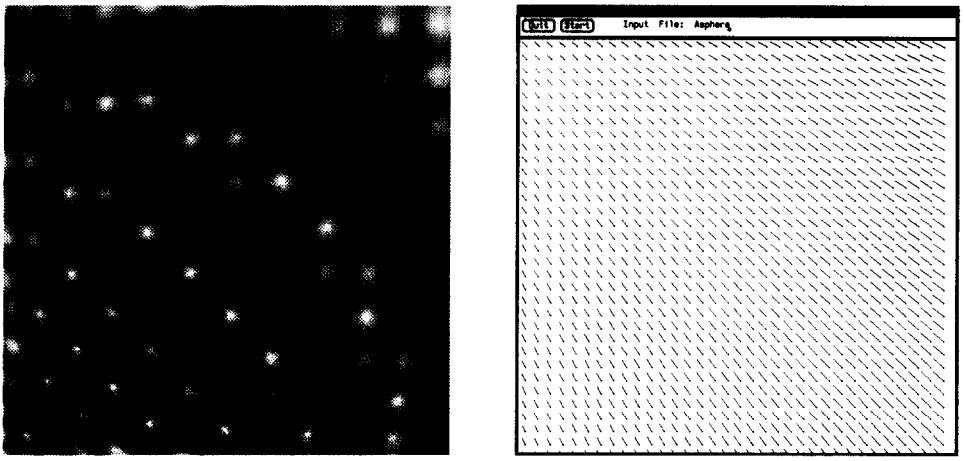

Fig. 4. Left: One frame of the sphere sequence. A sphere with a sinusoidal pattern is translating with velocity $(-2.0, 2.0, 1)$. Right: Computed optical flow field.

Sphere sequence

A sphere with a sinusoidal pattern is translating in an arbitrary direction. Images were generated using a simple ray tracing program and intensity values are quantized into 256 levels. The first frame of the sequence is shown in Fig. 4. The image sequence is 400×400 pixels in size and the average motion per frame is about 4 pixels. The focal length is 256 pixels in both x and y directions and the field of view is 76 degree in either direction. Discretization effects are severe due to the poor resolution. The signal to noise ratio for the computed flow field was around 50 dB i.e. the power of error was around 0.1% of the power of the flow field. Results are reported in Table 8. Note that the motion estimates with $\mathcal{B}_1$ are more stable than with $\mathcal{B}_2$. This is due to a large velocity gradient in the x direction compared to the y direction. This effect is clearer in the next experiment.

Cylinder sequence

The image sequence has a cylinder with random texture and is undergoing translation. The cylinder is colored with random patterns, as in a random dot cinematogram. The sizes of the patches are such that the projection of each patch is approximately of the same size. We show the first frame of the sequence in Fig. 5. The image sequence is of size 400×400 pixels. The focal length is 256 pixels in both x and y directions. The field of view is 76 degrees in either direction. Discretization effects are severe due to the poor resolution. The signal to noise ratio for the computed flow field is around 25 dB i.e. the power of error is around 4.0% of the power of the flow field. Results are reported in Table 8. This experiment clearly demonstrates the need for computing motion via both $\mathcal{B}_1$ and $\mathcal{B}_2$. The motion estimates using $\mathcal{B}_2$ are totally wrong. The reason is that in case of a cylinder oriented parallel to the image plane with axis along the y direction, there is no gradient in u in the y direction. That is, u_y is zero for the entire image plane. Because of noise, the situation that the system of equations is not of full

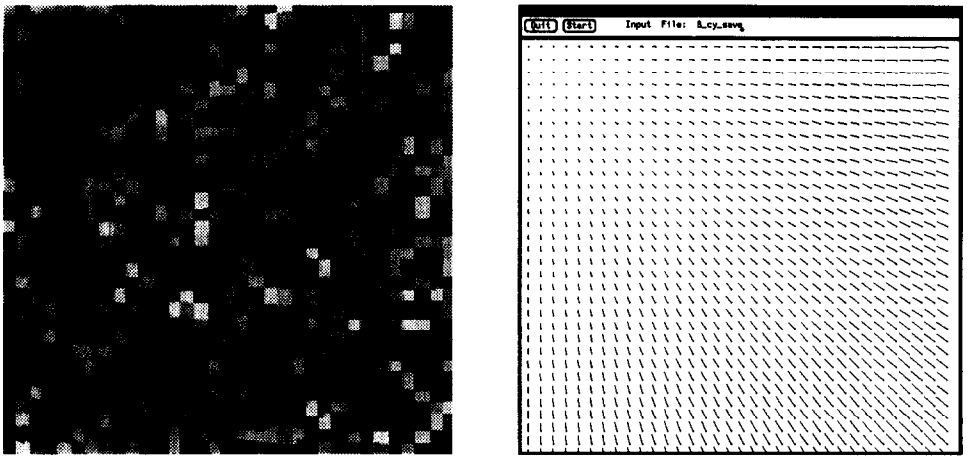

Fig. 5. Left: One frame of the cylinder sequence. A cylinder with a random pattern is translating with velocity $(-0.4, 0.4, 0.5)$. Right: Computed optical flow field.

rank cannot be detected resulting in a totally erroneous motion parameter. The fact that a wrong motion parameter has been recovered can be easily inferred from the condition number of the system of equations and by looking at how well the motion parameters satisfy Eq. (3).

Robot arm sequences

The robot arm sequences were made available to us by the Computer Vision Laboratory at the University of Maryland. There are six sequences with varying degree of motion complexities. The prominent motion being the translation along the optical axis and hence the flow fields appear to be diverging about the focus of expansion. One frame of the first image sequence is shown in Fig. 6. The image sequences are 652×574 pixels in size. The focal length of the camera is 1162.7 pixels in the x direction and 1315.6 pixels in the y direction. The field of view is 52.5 degrees in either direction. The camera is undergoing a general motion in these sequences. In the first sequence the camera is moving into the optical axis. The true motion in the case of the second and the third sequences is not known. In the fourth and the sixth sequence the camera is undergoing both translation and rotation. The fifth sequence has different rotation between successive frames.

The computed flow field for one sequence is shown in Fig. 6. We cannot comment on the quality of the optical flow obtained because the true motion fields are not available. The 3-D motion parameters are reported in Table 8. Only two components of the translation vector are reported, for t_3 is set to one. Note that the focus of expansion is computed with small error. The rotation parameters are not very reliable. The reason for that seems to be that the rotation parameters are fairly small (about 0.072 degree per frame) and hence the rotational component in the flow field is relatively small compared to the translational component.

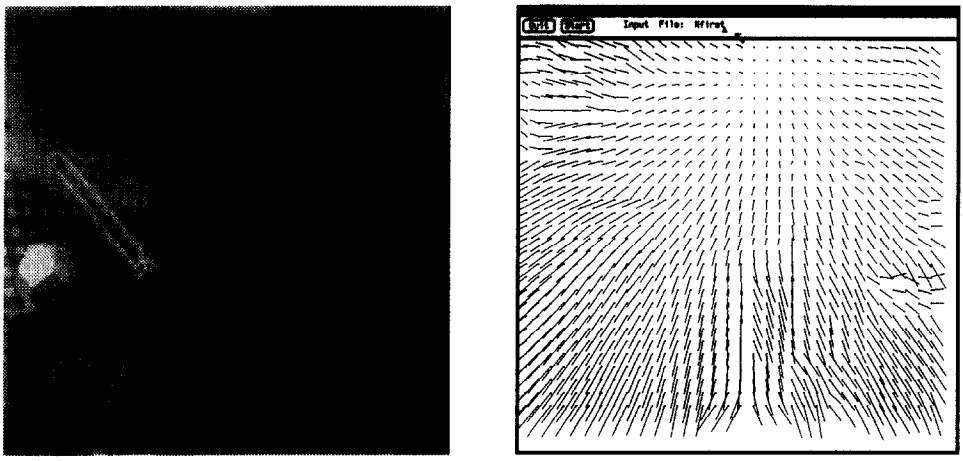

Fig. 6. Left: One frame of the robot arm 1 sequence. A real world scene in which camera is undergoing a translation. Right: Computed flow field for the first robot arm sequence.

Coke sequence

A frame of the coke can sequence and the computed optical flow are shown in Fig. 7. The camera is zooming into the center of the coke can, and hence the optical flow field is diverging about the center of the image frame. The quality of the computed optical flow suffers in the top left corner of the image. This is due to the lack of texture. In fact, for this region the condition number of the system of equations for obtaining the flow parameters is very large (20000+), denoting the unreliability of the computed optical flow. We ignore the optical flow from these regions for the 3-D motion and structure estimation. The average image motion for this image sequence is less than a pixel. The results are reported in Table 8.

PUMA2 sequence

The 30-frame long PUMA2 sequence was taken by connecting a camera to the end of a PUMA robot arm, and the robot arm was then rotated for a total of 120 degrees. The rotation between successive frames is about 4 degrees (0.0698 radian). The first frame of the sequence and the computed optical flow are shown in Fig. 8. The flow field seems to be correct and the rotational motion of the camera is clearly evident. From the optical flow field it is clear that the rotation is not about the center of the image, as reported earlier in several other articles, rather the center of rotation is shifted little to the left. However, the motion parameters were computed with the center of the image (121, 128) as the optical center.

In case of only rotational motion, the basic parameters will be all zero. The rotation vector cannot be computed from Eqs. (7) and (9), however, Eq. (3) yields a linear constraint for rotation in this case. The rotation vector can be computed easily.

Due to the error in the optical axis, the 3-D motion of the camera about the optical axis passing through the center of the image is not pure rotation. If the optical axis is shifted by $\boldsymbol{r}$, then the translation about the new axis is $\boldsymbol{\omega} \times \boldsymbol{r}$, where $\times$ indicates the

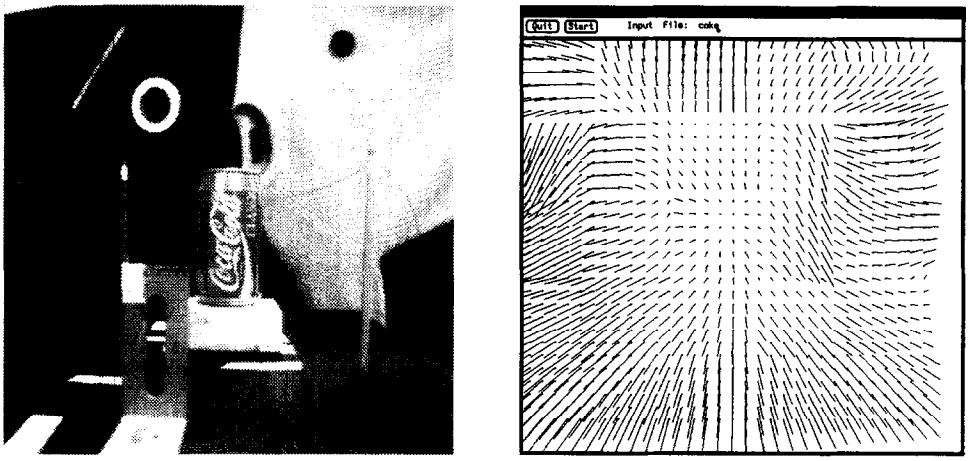

Fig. 7. Left: First frame of the coke can sequence. The camera is translating along the optical axis. Right: Computed optical flow.

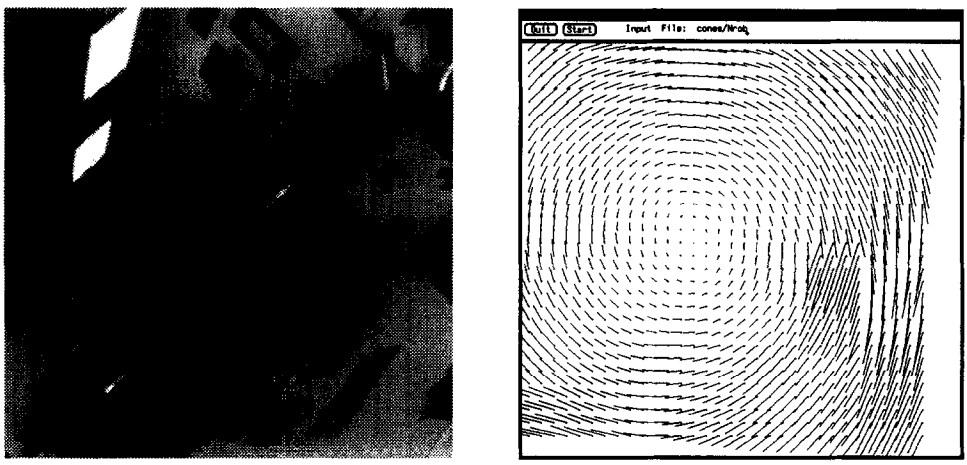

Fig. 8. Left: First frame of the PUMA2 sequence. Right: Computed optical flow field.

vector cross product. The recovered rotation vector about the y axis is around 0.647 radian which is very close to the actual rotation of 0.698 radian.

13. Conclusions

In this paper we have presented an algorithm to estimate the 3-D motion of objects from the motion field. Motion estimation has been one of the most difficult problems in computer vision. The reasons for this are the nonlinearity of the equations involved and the sensitivity of the motion parameters to noise. Introducing two sets of basic parameters, we obtained a system of linear equations for estimating the motion parameter

Table 8
Results for experiments with the computed optical flow

Seq.	True T $\times 10^{-2}$		True Rot $\times 10^{-4}$			Parm.	Comp T $\times 10^{-2}$		Error T deg	Comp Rot $\times 10^{-4}$		
						$\mathcal{B}_1$	−200	204	0.4	0.4	1.0	−1.67
Sph.	−200	200	0	0	0	$\mathcal{B}_2$	−195	181	5.5	−1.47	0.47	−3.9
						Final	−199	201	0.2	0.22	0.92	−1.9
						$\mathcal{B}_1$	−61	57	8.8	−15	−12	10.4
Cyl.	−80	80	0	0	0	$\mathcal{B}_2$	−98	−41	48.6	362	−62	36.4
						Final	−80	71	3.0	−6.0	0.48	0.51
						$\mathcal{B}_1$	2.8	6.9	4.3	2.2	−11	0.81
Rob 1	0	0	0	0	0	$\mathcal{B}_2$	−2.1	−3.9	2.5	13.7	−0.9	0
						Final	−2.0	1.6	1.5	3.31	−0.8	−0.3
						$\mathcal{B}_1$	19	6.3	NA	1.34	−30	−0.3
Rob 2	NA	NA	NA	NA	NA	$\mathcal{B}_2$	14	11	NA	−13	2.2	9.8
						Final	16	10	NA	0.33	2.11	0.59
						$\mathcal{B}_1$	11	2.2	NA	2.40	−17	1.60
Rob 3	NA	NA	NA	NA	NA	$\mathcal{B}_2$	2.0	−17	NA	43.5	−3.3	−2.4
						Final	11	1.9	NA	2.60	−1.0	0.94
						$\mathcal{B}_1$	15	−13	2.5	4.46	−11	0.53
Rob 4	11	−11	−1.3	−2.5	0	$\mathcal{B}_2$	15	−10	2.3	38.1	5.28	0.80
						Final	14	−12	1.8	−4.6	−6.3	−0.4
						$\mathcal{B}_1$	18	−13	12.5	2.46	−21	−0.3
Rob 5	0	0	variable		0	$\mathcal{B}_2$	−4	8.4	5.3	18.9	−1.8	0.39
						Final	3	3	2.5	1.27	−0.6	1.18
						$\mathcal{B}_1$	22	6.8	8.6	2.15	−5.4	0
Rob 6	6.7	6.7	1.3	1.3	0	$\mathcal{B}_2$	8.8	−15	12.3	50	2.78	−1.4
						Final	15	5.5	4.8	2.49	4.41	3.47
						$\mathcal{B}_1$	5.2	0.3	3.0	0.2	−2.1	1.0
Coke	0	0	0	0	0	$\mathcal{B}_2$	1.8	−1.5	1.4	10.0	2.70	−2.4
						Final	3.5	−1.0	2.1	7.34	2.21	0.47
						$\mathcal{B}_1$	25.8	3.54	NA	80.46	92.7	−683
Puma	NA	NA	0	0	−698	$\mathcal{B}_2$	14.4	−25.3	NA	113.2	13.73	−604
						Final	20.5	−9.49	NA	95.4	55.26	−647

from the motion field. The system of equations is stabilized by utilizing the information available in space. We proved the correctness, completeness and robustness of the approach and in that sense the problem of recovering 3-D motion from the motion field is "solved".

No assumptions are made about the nature of the surface or the motion. This makes the algorithm general and widely applicable. However, since the optical flow field can be obtained reliably only in the case of smooth surfaces undergoing a smooth motion, the applicability of the motion estimation algorithm is restricted to these cases for the lack of a more stable technique for computing optical flow. But, the stability of the motion estimation improves as the local structure of the object becomes complicated.

Specifically, we have been able to show that the condition number of the system of equations to obtain the basic parameters decreases as the contribution of higher order terms increases in the description of the inverse depth map h.

Acknowledgement

We are grateful to the anonymous reviewers and the editor of this special issue for their comments which have been very helpful in improving the contents and the form of this paper.

Appendix A. Completeness of the basic parameters

In this section we prove that the basic parameters cannot be obtained uniquely only if a certain type of quadratic surface is being viewed and the observer is on the surface itself. We prove that the system of equations

$$-2ax + cy + d + t_1 v_x - t_2 u_x - t_3 w_x = 0, \tag{A.1}$$

$$cx - 2by + e + t_1 v_y - t_2 u_y + t_3 w_y = 0 \tag{A.2}$$

for obtaining the basic parameters do not yield a unique solution only if the surface being viewed belongs to a class of seven parameter surfaces described by

$$s_1 X^2 + s_2 Y^2 + s_3 Z^2 + s_4 XY + s_5 XZ + s_6 YZ - X - Y - s_7 Z = 0.$$

The motion parameter can be obtained from either $\mathcal{B}_1$ or $\mathcal{B}_2$, hence the cases when no motion parameter can be recovered are the ones when both the system of equations (A.1) and (A.2) have rank 4 or less. The system of equations (A.1) and (A.2) has rank less than 5 if five of the coefficients of the basic parameters are linearly dependent. That is, we do not obtain basic parameters if

$$a_1 u_x + a_2 v_x + a_3 x + a_4 y + a_5 = 0, \tag{A.3}$$

$$b_1 u_y + b_2 v_y + b_3 x + b_4 y + b_5 = 0 \tag{A.4}$$

for all (x, y) in the image plane. In the above equations not all $(a_1, a_2, a_3, a_4, a_5)$ or $(b_1, b_2, b_3, b_4, b_5)$ should be zero. Our objective then is to obtain the condition on the surface structure when the above equations are satisfied simultaneously.

From the motion field equations we obtain:

$$u_x = (-t_1 + t_3 x) h_x + h + Ay - 2Bx,$$

$$u_y = (-t_1 + t_3 x) h_y + Ax + C,$$

$$v_x = (-t_2 + t_3 y) h_x - Bx - C,$$

$$v_y = (-t_2 + t_3 y) h_y + h + 2Ay - Bx.$$

Substituting the above equations into Eqs. (A.3) and (A.4), we get

$$a_1 h + (a_1 x t_3 + a_2 y t_3 - a_1 t_1 - a_2 t_2) h_x + (a_3 - 2Ba_1 - Ba_2)x + y(Aa_1 + a_4) + a_5 - a_2 C = 0,$$

$$b_2 h + (b_1 x t_3 + b_2 y t_3 - b_1 t_1 - b_2 t_2) h_y + (b_3 - b_2 B + Ab_1)x + y(b_4 + 2b_2 A) + b_1 C + b_5 = 0.$$

Rewriting the above differential equations we get

$$h_x + \frac{1}{x - \eta y + \gamma} h = \frac{\alpha x + \beta y + \delta}{x - \eta y + \gamma},$$

$$h_y + \frac{1}{y - \tilde{\eta} x + \tilde{\gamma}} h = \frac{\tilde{\alpha} x + \tilde{\beta} y + \tilde{\delta}}{y - \tilde{\eta} x + \tilde{\gamma}},$$

where the various Greek symbols are used for conciseness. We assumed that the parameters (a_1, b_2) are nonzero; them being zero results in a planar surface. That is, the cases when no basic parameters can be obtained are the set of planar surfaces and the surfaces which simultaneously satisfy the above set of differential equations.

The solution for the above differential equations is straightforward and can be shown to be

$$h = \frac{1}{x - \eta y + \gamma} \left(\alpha \frac{x^2}{2} + \beta x y + \delta x + c^*(y) \right),$$

$$h = \frac{1}{y - \tilde{\eta} x + \tilde{\gamma}} \left(\tilde{\alpha} \frac{y^2}{2} + \tilde{\beta} x y + \tilde{\delta} y + \tilde{c}^*(x) \right),$$

where $c^*(y)$ is any function of y only and $\tilde{c}^*(x)$ is a function of x only. We are interested in surfaces which belong to the above two classes of surfaces simultaneously. This class is given by

$$h = \frac{1}{x + y + \gamma} \left(\alpha \frac{x^2}{2} + \beta x y + \delta x + \tilde{\alpha} \frac{y^2}{2} + \tilde{\delta} y + c^* \right),$$

and this happens when η and $\tilde{\eta}$ are equal to -1, $\gamma = \tilde{\gamma}$, $\beta = \tilde{\beta}$, and

$$c^*(y) = \tilde{\alpha} \frac{y^2}{2} + \tilde{\delta} y + c^*,$$

$$\tilde{c}^*(x) = \alpha \frac{x^2}{2} + \delta x + c^*.$$

This is the class of surfaces for which no basic parameters and hence no motion parameter can be obtained. The surface is however represented in terms of the image coordinate (x, y) and is not very easy to comprehend. To make it clear we obtain the surface equation in terms of the global coordinate (X, Y, Z). Substituting the projection equations in the above we get

$$\alpha \frac{X^2}{2} + \tilde{\alpha} \frac{Y^2}{2} + c^* \frac{Z^2}{2} + \beta XY + \delta XZ + \tilde{\delta} YZ - X - Y - \gamma Z = 0,$$

which represents a class of seven parameter quadratic surfaces and is a subset of all the quadratic surfaces. Note that there is no constant term in the above expression, which means that the surface should pass through the origin or in other words, the viewer should be on the surface itself.

References

[1] G. Adiv, Determining three-dimensional motion and structure from optical flow generated by several moving objects, *IEEE Trans. Pattern Anal. Mach. Intell.* **7** (1985) 384–401.

[2] A.R. Bruss and B.K.P. Horn, Passive navigation, *Comput. Vision Graph. Image Process.* **21** (1983) 3–20.

[3] R.L. Burden, J.D. Faires and A.C. Reynolds, *Numerical Analysis* (Prindle, Weber and Schmidt, Boston, MA, 1978).

[4] J.J. Gibson, *The Perception of the Visual World* (Riverside Press, Cambridge, 1950).

[5] J.J. Gibson, *The Senses Considered as Perceptual Systems* (Houghton-Mifflin, Boston, MA, 1966).

[6] J.J. Gibson, On the analysis of change in the optic array, *Scand. J. Psych.* **18** (1977) 161–163.

[7] N.C. Gupta, Recovering shape and motion from a sequence of images, Ph.D. Dissertation, Department of Computer Science, University of Maryland, College Park, MD (1993).

[8] N.C. Gupta and L.N. Kanal, On recovering shape from motion, in: *Proceedings ACCV-95* (to appear).

[9] N.C. Gupta and L.N. Kanal, Recovering robust image motion, *Int. J. Comput. Vision* (to appear).

[10] K.J. Hanna, Direct multi-resolution estimation of ego-motion and structure from motion, in: *Proceedings IEEE Workshop on Visual Motion* (1991) 156–162.

[11] J.C. Hay, Optical motions and space perception: an extension of Gibson's analysis, *Psych. Rev.* **73** (1960) 550–565.

[12] D.J. Heegar and A.D. Jepson, Subspace methods for recovering rigid motion I: algorithm and implementation, *Int. J. Comput. Vision* **7** (1992) 95–117.

[13] J. Heel, Dynamic motion vision, *Rob. Autonom. Syst.* **6** (1990) 297–314.

[14] B.K.P. Horn, *Robot Vision* (MIT Press, Cambridge, MA, 1986).

[15] B.K.P. Horn, Motion fields are hardly ever ambiguous, *Int. J. Comput. Vision* **1** (1987) 259–274.

[16] S.V. Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis* (SIAM, Philadelphia, PA, 1991).

[17] A.D. Jepson and D.J. Heegar, A fast subspace algorithm for recovering rigid motion, in: *Proceedings IEEE Workshop on Visual Motion* (1991) 124–131.

[18] A.D. Jepson and D.J. Heegar, Linear subspace methods for recovering translational direction, Tech. Rept. RBCV-TR-92-40, University of Toronto, Toronto, Ont. (1992).

[19] J.J. Koenderink and A.J. van Doorn, Invariant properties of the motion parallax field due to the movement of rigid bodies relative to the observer, *Optica Acta* **22** (1975) 773–791.

[20] E. Kreyszig, *Advanced Engineering Mathematics* (Wiley, New York, 1967).

[21] D.T. Lawton, Processing translational motion sequences, *Comput. Vision Graph. Image Process.* **22** (1983) 116–144.

[22] H.C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Nature* **293** (1981) 133–135.

[23] H.C. Longuet-Higgins, The visual ambiguity of a moving plane, *Proc. Roy. Soc. London Ser. B* **223** (1984) 165–175.

[24] H.C. Longuet-Higgins and K. Prazdny, The interpretation of a moving retinal image, *Proc. Roy. Soc. London Ser. B* **208** (1980) 385–397.

[25] S.J. Maybank, The angular velocity associated with the optical flow field arising from motion through a rigid environment, *Proc. Roy. Soc. London Ser. A* **401** (1985).

[26] S.J. Maybank, The angular velocity associated with the optical flow field due to a single moving rigid plane, in: *Proceedings ECAI-84*, Pisa (1984) 641–644.

[27] L. Matthies and T. Kanade, Kalman filter-based algorithms for estimating depth from image sequence, *Int. J. Comput. Vision* **3** (1989) 209–236.

[28] S. Negahdaripour, Direct passive navigation, Ph.D. Dissertation, Department of Mechanical Engineering, MIT, Cambridge, MA (1986).
[29] S. Negahdaripour and B.K.P. Horn, Direct passive navigation, *IEEE Trans. Pattern Anal. Mach. Intell.* **9** (1987) 168–176.
[30] S. Negahdaripour and S. Lee, Motion recovery from image sequences using only first order optical flow information, *Int. J. Comput. Vision* **9** (1992) 163–184.
[31] K. Prazdny, Determining the instantaneous direction of motion from optical flow generated by curvilinearly moving observer, *Comput. Vision Graph. Image Process.* **17** (1981) 238–248.
[32] J.H. Reiger and D.T. Lawton, Determining the instantaneous axis of translation from optical flow generated by arbitrary sensor motion, in: *Proceedings ACM Interdisp. Workshop Motion*, Toronto, Ont. (1983).
[33] P. Schmid, *Econometrics* (Marcel Dekker, New York, 1976).
[34] V. Sundareswaran, Egomotion from global flow field data, in: *Proceedings IEEE Workshop on Visual Motion* (1991) 140–145.
[35] R.Y. Tsai and T.S. Huang, Estimating 3-D motion parameters of a rigid planar patch, *IEEE Trans. ASSP* **29** (1981) 1147–1152.
[36] R.Y. Tsai and T.S. Huang, Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces, *IEEE Trans. Pattern Anal. Mach. Intell.* **6** (1984) 13–27.
[37] Van der Sluis, Stability of solutions of linear algebraic systems, *Numer. Math.* **14** (1970) 246–251.
[38] A.M. Waxman and S. Ullman, Surface structure and three-dimensional motion from image flow kinematics, *Int. J. Rob. Res.* **4** (1985) 95–108.