

# The Feasibility of Motion and Structure from Noisy Time-Varying Image Velocity Information

JOHN L. BARRON

*Department of Computer Science, University of Western Ontario, London, Canada, N6A 5B7*

ALLAN D. JEPSON AND JOHN K. TSOTSOS

*Department of Computer Science, University of Toronto, Toronto, Canada, M5S 1A4*

## Abstract

This research addresses the problem of noise sensitivity inherent in motion and structure algorithms. The motion and structure paradigm is a two-step process. First, we measure image velocities and, perhaps, their spatial and temporal derivatives, are obtained from time-varying image intensity data and second, we use these data to compute the motion of a moving monocular observer in a stationary environment under perspective projection, relative to a single 3-D planar surface. The first contribution of this article is an algorithm that uses time-varying image velocity information to compute the observer's translation and rotation and the normalized surface gradient of the 3-D planar surface. The use of time-varying image velocity information is an important tool in obtaining a more robust motion and structure calculation. The second contribution of this article is an extensive error analysis of the motion and structure problem. Any motion and structure algorithm that uses image velocity information as its input should exhibit error sensitivity behavior compatible with the results reported here. We perform an average and worst case error analysis for four types of image velocity information: full and normal image velocities and full and normal sets of image velocity and its derivatives. (These derivatives are simply the coefficients of a truncated Taylor series expansion about some point in space and time.) The main issues we address here are: just how sensitive is a motion and structure computation in the presence of noisy input, or alternately, how accurate must our image velocity information be, how much and what type of input data is needed, and under what circumstances is motion and structure feasible? That is, when can we be sure that a motion and structure computation will produce usable results? We base our answers on a numerical error analysis we conduct for a large number of motions.

## 1 Introduction

### 1.1 The Problem and Solution Methodology

This article outlines a motion and structure algorithm that uses time-varying image velocity information to compute the motion of a monocular observer moving relative to a stationary 3-D planar surface under perspective projection. Due to the depth-speed ambiguity experienced by a monocular observer, his translation  $\vec{U}$  and the 3-D coordinates of environmental points of the planar surface,  $\vec{P}$ , cannot be recovered. Figure 1 illustrates this. Hence, we can only compute  $\vec{u}$ , the depth-scaled observer translation and  $\vec{\alpha}$ , the normalized surface gradient of the 3-D planar surface. Since the

rotational component of image velocity is independent of 3-D depth we can fully recover the observer's rotation,  $\vec{\omega}$ . To determine these motion and structure parameters we derive nonlinear equations relating image velocity and its derivatives at some image point and time,  $\vec{V}(\vec{P}', t')$  to the underlying motion and structure parameters at another image point and time,  $\vec{V}(\vec{P}, t)$ . Hence, we are able to use time-varying image velocity information in the computation of motion and structure. This is in sharp contrast with most previous approaches (for references, see [Barron 1984, 1988] and section 1.2) to motion and structure where image velocity information is measured at one time only. As we shall see in what follows, the use of time-varying image velocity information can often reduce error

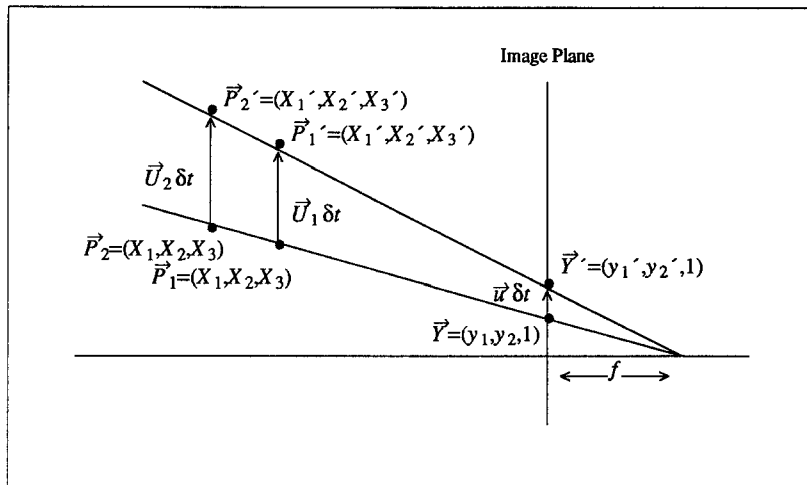


Fig. 1. The depth-speed ambiguity for a monocular observer. If  $\vec{U}_1$  and  $\vec{U}_2$  are the 3-D translational velocities of  $\vec{P}_1$  and  $\vec{P}_2$  respectively, both with the same image point  $\vec{Y}$ , then both 3-D translational velocities project onto the image plane with the same image velocity proportional to depth-scaled translation  $\vec{u} = \vec{U}_1 / \|\vec{P}_1\|_2 = \vec{U}_2 / \|\vec{P}_2\|_2$ . Hence, neither 3-D translation or 3-D depth can be recovered for a monocular observer, only their ratio.

sensitivity in motion and structure calculations. Our solution technique then simply involves solving non-linear systems of equations (often in the least-squares sense) relating time-varying image velocity information to the appropriate motion and structure parameters. Due to the aperture problem [Horn and Schunk 1981] (see figure 2) it may be easier to compute normal image velocity and its derivatives rather than full image veloc-

ities and its derivatives. Hence, we investigate the use of both full and normal image velocities and full and normal Taylor series coefficients in the recovery of motion and structure. We consider the use of Taylor series coefficients computed from (normal) image velocity fields [Waxman and Ullman 1983, 1985] or measured directly from time-varying image intensities [Eagleson 1989].

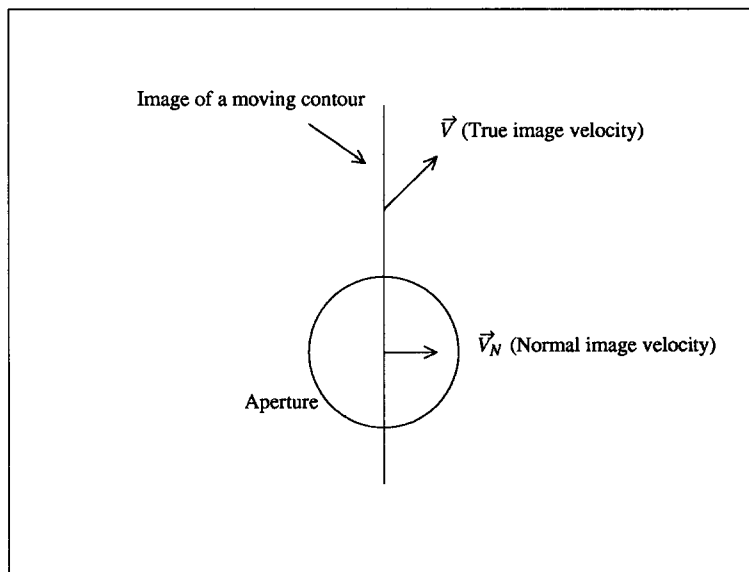


Fig. 2. The true image velocity,  $\vec{V}$  cannot be distinguished from the image velocity normal to a moving contour,  $\vec{V}_N$ , when viewed through an aperture.

## 1.2 Literature Survey

The most common approach to monocular reconstruction involves solving (generally nonlinear) systems of equations relating image velocity (or image displacement) to a set of motion and structure parameters [Adiv 1984; Barron 1988; Bruss and Horn 1983; Buxton et al. 1984; Dreschler and Nagel 1982; Fang and Huang 1984; Kanatani 1985; Lawton 1983; Prazdny 1979; Roach and Aggarwal 1980; Tsai et al. 1982; Tsai and Huang 1984; Webb and Aggarwal 1981]. Some of the issues that arise for these algorithms are the need for good initial guesses of the solution, the possibility of multiple solutions, and the need for accurate input. The latter is by far the most important issue if a motion and structure algorithm is to be judged a success. As Waxman and Ullman [1983; 1985] and others have noted, motion and structure algorithms that use image velocities of neighboring image points require accurate differences of these similar velocities. That is, solving systems of equations effectively requires subtraction of very similar quantities: the error in the quantities themselves may be quite small but since the magnitudes of their differences are quite small, the relative error in them can be quite large. Hence, such techniques can be expected to be sensitive to input errors. Adiv [1984] and Bruss and Horn [1983] suggest this sensitivity might be reduced if a least-squares minimization was computed over the entire image.

A second approach to reconstruction involves solving nonlinear systems of equations relating local image velocity information (one image velocity and its first- and second-order spatial derivatives) to the underlying motion and structure parameters [Longuet and Higgins 1981; Waxman and Ullman 1983, 1985; Subbarao 1986]. The rationale is that using local information about one image point means that the problem of similar neighboring image velocities can be averted. However, this is replaced with the problem of computing these first- and second-order spatial derivatives. Waxman and Wohn [1984, 1985] propose that these derivatives be found by solving linear systems of equations, where each equation specifies the normal component of image velocity on a moving nonoccluding contour in terms of a Taylor series expansion of the  $x$  and  $y$  components of full or normal image velocity. They call this the Velocity Functional Method. In effect, their motion and structure algorithm divides the computation into two steps: use a normal velocity distribution to compute image velocity and its first- and second-order spatial

derivatives at an image point and then use these as input to an algorithm that solves the nonlinear equations relating motion and structure to the image velocity and its first- and second-order derivatives. More recently, Subbarao [1986] and Waxman et al. [1987] have proposed closed-form solutions for motion and structure. This basically involves solving a cubic equation and a set of decoupled nonlinear equations.

A third approach to reconstruction involves using spatiotemporal intensity derivatives directly as input to a motion and structure algorithm [Aloimonos and Brown 1984; Horn and Weldon 1988; Negahdaripour and Horn 1987]. The claimed advantage is that the computation of optic flow as an intermediate step is bypassed. However, the error sensitivity behavior of this approach has not yet been fully analyzed.

Only recently, have researchers begun to address the use of temporal information, such as temporal derivatives, in reconstruction [Subbarao 1986; Bandyopadhyay and Aloimonos 1985]. We note that others' use of temporal derivative information and our use of time-varying image velocity information differ in one important respect: we measure our information, including spatial and temporal derivative information at many different times rather than at one just one time. The time interval between the measurement of adjacent image velocity fields can be arbitrarily large, provided our acceleration model (see below) is satisfied. Only if the time intervals are small can we accurately approximate temporal image velocity derivatives (as a finite difference) from the image velocities. Otherwise, image velocities measured at different times have no relationship with temporal image velocity derivatives. We note that, via a Taylor series expansion, image velocity fields (at least locally) can be approximated from one image velocity and its first and/or second spatial and temporal derivatives and vice versa. Indeed, image velocity fields are often used in the derivation of spatial and temporal image velocity information [Waxman and Ullman 1985; Waxman and Wohn 1985].

Some other work [Broida and Chellappa 1986; Shariat and Price 1986a, b] has been concerned with the use of image displacements over multiple frames in motion and structure computations. As well, Matthies et al. [1989] and Wu et al. [1989] have used a Kalman-filter approach to integrate information over time in the computation of 3-D information from a sequence of images. Spetsakis and Aloimonos [1988] also showed the effectiveness of using least squares in image displacement motion techniques.

It is somewhat disappointing that almost none of these reconstruction techniques have been successfully applied to flow fields calculated from realistic scenes. Primarily, the problem is the difficulty in computing accurate flow fields. There has been little or no error analysis in previous monocular reconstruction work, although some researchers (see, e.g., Waxman and Ullman [1983, 1985], Buxton et al. [1984], Aloimonos and Rigoutsos [1986], Subbarao [1986], and Snyder [1986]) have begun to consider the inherent sensitivity of their algorithms to random noise in the input. See Barron [1984; 1988] for a more detailed survey of motion and structure algorithms and their problems.

### 1.3 Contributions of This Work

Most approaches to reconstruction use an image velocity field at one time instance only. We propose that time-varying image velocities arising from a moving surface be used. By using image velocities distributed over both space and time, the range of image velocity magnitudes may be much larger than the range of image velocity magnitudes distributed over space at one time only. Hence, the problem of similar neighboring velocities may be reduced. Intuitively, this makes sense, as using image velocities over a time interval is, in effect, simply extending the spatial extent of the surface (and, generally, the larger the spatial extent, the greater the range of image velocity magnitudes). As well, the use of time, may allow the recovery of motion and structure parameters from flow fields that are not analyzable at one time only. A couple of other advantages result from using a spatiotemporal distribution of image velocity; namely, sparse flow fields may be used (often accurate image velocity can only be computed at a few points in each image) and occlusion of parts of a surface at different times can be handled as we can use image velocities on the visible parts of the surface at any given time.

In order to relate a spatiotemporal distribution of image velocities to the motion and structure parameters at some image point we need to make some assumptions:

1. The use of perspective transformations requires the assumption of *rigidity*; objects are assumed to be rigid. This ensures that the image velocity of an object's point is due entirely to the point motion with respect to the observer and not due to changes in the object's shape. There is much psychological evidence (e.g., Hay [1966] and Ullman [1979]) to support the premise that humans have a tendency to use the rigidity assumption to analyze world scenes.
2. The 3-D surfaces of objects can be described locally as a plane. The *local planarity* assumption means curved surfaces are treated as collections of adjacent planes.
3. The observer rotates with a constant angular velocity for some small time interval (with respect to the observer coordinate system). Webb and Aggarwal [1981] call this the *fixed-axis* assumption.
4. The spatiotemporal distribution of image velocity results from 3-D points on the same planar surface. We call this the *same-surface* assumption.
5. The observer's type of motion is known a priori. This *known-motion-type* assumption is necessary for relating  $\vec{u}$  values at various image locations on a planar surface at different times to each other and will be explained in more detail below. We also refer to this as the *acceleration model*.

The use of a spatiotemporal distributions of image velocity means motion and structure can be computed using local spatiotemporal data; thus we are not necessarily restricted to stationary environments as we can treat each independently moving surface as stationary relative to the moving observer.

The reader may question the validity of using these kinds of assumptions, especially when real-world imagery is used as the input. We treat violation of these assumptions as one type of error in the input data. We believe that artificial situations such as the one created by our assumptions need to be fully analyzed before the even harder problem of using real-image input can be addressed. We show below that the use of separate image velocities measured at separate individual times generally increases the robustness of the motion and structure calculation in most cases (especially in the least squares case). As well, it allows the recovery of motion and structure in situations not possible using only one flow field.

The use of the local planarity and fixed-axis assumptions means that the point-to-point correspondence problem does not have to be solved, that is, we do not have to use velocities of the same 3-D points at different time intervals, as it is now mathematically possible to relate image velocities distributed in space and time at any point and time on a 3-D planar surface to the motion and structure parameters of any other point on the planar surface at any other time (where these assumptions are reasonably satisfied).<sup>1</sup> Other researchers, such

as Kanatani [1985] and Aloimonos and Rigoutsos [1986], have also advocated a correspondenceless approach. Direct methods for computing motion and structure [Aloimonos and Brown 1984; Horn and Weldon 1988; Negahdaripour and Horn 1987] may also be classified as correspondenceless. The computation of image velocity may require solving the correspondence problem, although there is a group of techniques based on the relationship between spatial and temporal grayvalue distributions, (see, e.g., [Horn and Schunck 1981; Fleet and Jepson 1990; Waxman et al. 1988; Heeger 1987]) for determining image velocity without the need to compute correspondence.

The algorithm presented here involves solving nonlinear systems of equations that relate a spatiotemporal distribution of image velocity to a set of motion and structure parameters at some image point at a particular time. Newton's method is used to solve the equations. As a result, an initial guess to the actual solution is needed to start the convergence calculation. This article investigates two important questions: how good does the initial guess have to be and how accurate must the time-varying image velocity input be? Some preliminary results were presented by Barron et al. [1987a, b, c].

We note that only the first question is algorithm dependent. The second question involves the inherent sensitivity of the exact solution of the solution and structure equations to perturbations in the image velocity data. Any algorithm based solely on solving these types of equations will exhibit this sensitivity.

## 2 Mathematical Preliminaries

In this section we present a brief description of our algorithm. Complete details are in [Barron 1988]. We use notation  $\vec{P}(t; \tau)$  to indicate a 3-D point measured at time  $t$  with respect to a coordinate system  $\vec{X}(\tau)$ . Similarly,  $X_3(\vec{P}, t; \tau)$  is the depth of  $\vec{P}(t; \tau)$ .  $\vec{Y}(\vec{P}, t)$  is the image of  $\vec{P}(t; t)$ . We adopt a right-handed coordinate system as in Longuet-Higgins and Prazdny [1980] and shown in figure 3.  $\vec{U} = (U_1, U_2, U_3)$  is the translational velocity of the observer, centered at the origin of the coordinate system  $\vec{X}(t)$  and  $\vec{\omega} = (\omega_1, \omega_2, \omega_3)$  is the angular velocity of the observer. The image of  $\vec{P}$  is located at  $\vec{Y} = (y_1, y_2, 1)$ . The origin of the image plane is  $(0, 0, 1)$ , that is, the focal length,  $f$ , is 1. The  $X_3$  axis is the line of sight.

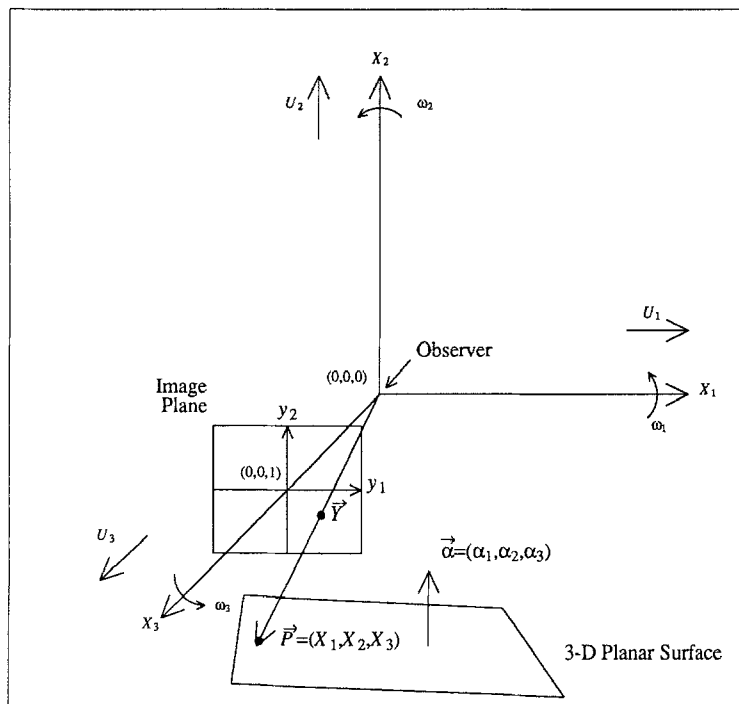


Fig. 3. The observer-based coordinate system.  $\vec{U} = (U_1, U_2, U_3)$  is the observer's 3-D translational velocity while  $\vec{\omega} = (\omega_1, \omega_2, \omega_3)$  is his 3-D rotational velocity. The image plane is at depth 1. The image of  $\vec{P}$  is located at  $\vec{Y} = (y_1, y_2, 1)$ .  $\vec{\alpha} = \alpha_1, \alpha_2, \alpha_3$  is the normalized surface gradient of the 3-D planar surface that  $\vec{P}$  belongs to. The origin of the image plane is  $(0, 0, 1)$ . The  $X_3$  axis is the line of sight

2.1 The General Monocular Image Velocity Equation

We can write an equation relating image velocity at some image point  $\bar{Y}(\bar{P}_j, t')$  to the monocular motion and structure parameters at some image point  $\bar{Y}(\bar{P}_i, t)$  as

$$\begin{aligned} \bar{v}(\bar{Y}(\bar{P}_j, t'), t') &= A_1(\bar{Y}(\bar{P}_j, t')) h(\bar{Y}(\bar{P}_i, t)) \\ &\quad Q_i(\bar{\omega}, t, t') \bar{u}(\bar{Y}(\bar{P}_i, t), t; t) \\ &\quad S_M(\bar{Y}(\bar{P}_i, t), \bar{Y}(\bar{P}_j, t'), t') \\ &\quad T_M(\bar{Y}(\bar{P}_i, t), t, t') \\ &\quad + A_2(\bar{Y}(\bar{P}_j, t')) \bar{\omega}(t; t). \end{aligned} \quad (1)$$

where  $\bar{P}_i$  and  $\bar{P}_j$  are 3-D points on the same planar surface and generally  $\bar{Y}(\bar{P}_i, t) \neq \bar{Y}(\bar{P}_j, t')$ . In the above equation

$$A_1(\bar{Y}(\bar{P}, t)) = \begin{pmatrix} -1 & 0 & y_1 \\ 0 & -1 & y_2 \\ 0 & 0 & 0 \end{pmatrix}, \quad (2)$$

$$A_2(\bar{Y}(\bar{P}, t)) = \begin{pmatrix} y_1 y_2 & -(1 + y_1^2) & y_2 \\ (1 + y_2^2) & -y_1 y_2 & -y_1 \\ 0 & 0 & 0 \end{pmatrix}, \quad (3)$$

$h(\bar{Y}(\bar{P}, t))$  is the *perspective correction function* that specifies the ratio between the depth of  $\bar{P}(t; t)$ ,  $X_3(\bar{P}, t; t)$  and its 3-D distance from the observation point,  $\|\bar{P}(t; t)\|_2 = (\bar{P}(t; t) \cdot \bar{P}(t; t))^{1/2}$ , that is,

$$h(\bar{Y}(\bar{P}, t), t; t) = \frac{\|\bar{P}(t; t)\|_2}{X_3(\bar{P}, t; t)} = \|\bar{Y}(\bar{P}, t)\|_2 \quad (4)$$

and  $\bar{u}(\bar{Y}(\bar{P}, t), t; t)$  is the depth-scaled translational velocity of the observer,

$$\bar{u}(\bar{Y}(\bar{P}, t), t; t) = \frac{\bar{U}(t; t)}{\|\bar{P}(t; t)\|_2} \quad (5)$$

Before we define  $Q$ ,  $S_M$  and  $T_M$  below we need to introduce the observer's acceleration model. One of the advantages of using a single instantaneous image velocity field is that no assumptions about the observer's motion, for example his acceleration, have to be made. However, the use of a spatiotemporal distribution of image velocities requires that we relate the motion and structure parameters at one time to those at another time. Hence, we need an observer acceleration model. In this paper, we consider two specific types of motion, although we emphasize that our treatment can be generalized to other motions as well. The two types of motion considered are:

**Type 1 (Linear Motion, Rotating Observer):** A vehicle is moving with constant translational velocity and has a camera mounted on it that is rotating with constant angular velocity. The translation and rotation are constant with respect to the inertial coordinate system,  $\bar{X}(0)$ .

**Type 2 (Circular Motion, Fixed Observer):** A vehicle with a fixed mounted camera is moving with constant translational and angular velocity. The translation and rotation are measured in the current coordinate system,  $\bar{X}(t)$ .

Figures 4 and 5 show top-down views of a vehicle undergoing type 1 and type 2 motion respectively.

$$Q_1(\bar{\omega}, t, t') = R^T(\omega, t')R(\bar{\omega}, t)$$

and

$$Q_2(\bar{\omega}, t, t') = I \quad (\text{the identity matrix}),$$

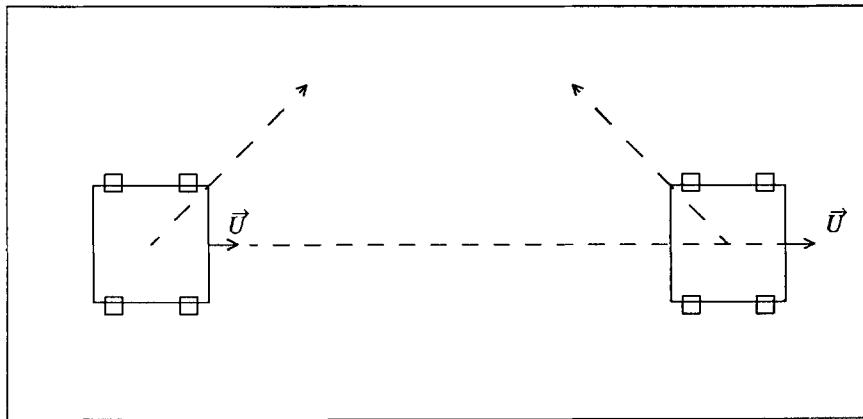


Fig. 4. A top-down view of type 1 motion. A vehicle undergoes pure constant translation while the camera mounted on it rotates about some fixed axes.

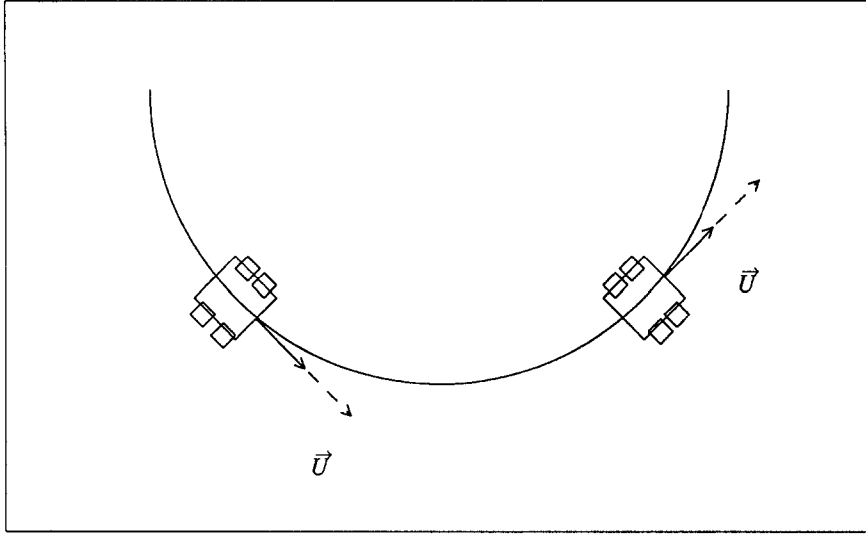


Fig. 5. A top-down view of type 2 motion. A vehicle moving with constant speed,  $\|\vec{U}\|_2$ , has a camera rigidly mounted on it. The camera and the vehicle are rotated together as they translate, resulting in the curvilinear motion shown above. The view is top-down.

for types 1 and 2 motion respectively.  $R(\vec{\omega}, t)$  is an orthogonal matrix specifying the rotation  $\|\vec{\omega}\|_2 t$  of  $\vec{X}(t)$  with respect to  $\vec{X}(0)$ .  $S_M$ , the *monocular spatial scaling function*,

$$\begin{aligned} S_M(\vec{Y}(\vec{P}_i, t), \vec{Y}(\vec{P}_j, t), t) &= \frac{\vec{\alpha}(\vec{P}_j, t; t) \cdot \vec{Y}(\vec{P}_j, t)}{\vec{\alpha}(\vec{P}_i, t; t) \cdot \vec{Y}(\vec{P}_i, t)} \\ &= \frac{X_3(\vec{P}_j, t; t)}{X_3(\vec{P}_i, t; t)} \end{aligned} \quad (6)$$

specifies the depth ratio of two 3-D points,  $\vec{P}_i$  and  $\vec{P}_j$  on the same planar surface at the same time.  $T_M$ , the *monocular temporal scaling function*,

$$\begin{aligned} T_M(\vec{Y}(\vec{P}_i, t), t, t') &= \frac{X_3(\vec{P}_i, t; t)}{X_3(\vec{P}_k, t'; t')} \\ &= \{ \vec{\alpha}(\vec{P}_i, t''; t'') \cdot \vec{Y}(\vec{P}_i, t) \} \div \\ &\quad \{ \vec{\alpha}(\vec{P}_k, t''; t'') \cdot R^T(\vec{\omega}, t'')R(\vec{\omega}, t) \\ &\quad \times [\vec{Y}(\vec{P}_i, t) - \vec{\Delta}d(\vec{P}_i, t, t'; t) \\ &\quad \times h(\vec{Y}(\vec{P}_i, t))] \} \end{aligned} \quad (7)$$

specifies the depth ratio of two 3-D points,  $\vec{P}_i$  and  $\vec{P}_k$  at times  $t$  and  $t'$ , where  $\vec{Y}(\vec{P}_i, t) = \vec{Y}(\vec{P}_k, t')$ .  $\vec{\Delta}d(\vec{P}, t, t'; t)$  is the depth-scaled observer displacement from time  $t$  to  $t'$  in  $\vec{X}(t)$  coordinates,

$$\vec{\Delta}d(\vec{P}, t, t'; t) = \frac{\vec{\Delta}D(t, t'; t)}{\|\vec{P}\|_2} \quad (8)$$

$\vec{\Delta}D(t, t'; t)$  is the absolute observer displacement from time  $t$  to  $t'$ . For type 1 motion

$$\vec{\Delta}D_1(t, t'; t) = \vec{U}_1(t; t)(t' - t)$$

yielding

$$\vec{\Delta}d_1(\vec{P}, t, t'; t) = \vec{u}_1(\vec{Y}(\vec{P}, t), t; t)(t' - t). \quad (9)$$

For type 2 motion

$$\vec{\Delta}D_2(t, t'; t) = R^T(\vec{\omega}, t) \left[ \int_t^{t'} R(\vec{\omega}, s) ds \right] \vec{U}_2(t; t)$$

yielding

$$\begin{aligned} \vec{\Delta}d_2(\vec{P}, t, t'; t) &= \\ R^T(\vec{\omega}, t) \left[ \int_t^{t'} R(\vec{\omega}, s) ds \right] \vec{u}_2(\vec{Y}(\vec{P}, t), t; t). \end{aligned} \quad (10)$$

The integral in (10) can be computed using adaptive quadrature. However, since type 2 motion involves the observer moving on a path that traces out a circular helix, a closed-form expression is possible. The particular expression is messy and therefore omitted.

Equation (1) reduces to the standard image velocity equation [Longuet-Higgins and Prazdny 1980] when  $t = t' = 0$ , and image velocity is measured at the same image point the motion and structure parameters are measured at, that is,  $Q = I$ ,  $S_M = 1$ , and  $T_M = 1$ . Given eight distinct components of image velocity

distributed over space and time, but on the same 3-D planar surface, we can construct and solve a nonlinear system of equations to determine the motion and structure parameters.

In addition to recovering motion and structure from full image velocity data, we can also use normal image velocity data directly in the computation of motion and structure. We can write

$$\|\bar{v}_n(\bar{Y}(\bar{P}, t), t)\|_2 = \bar{v}(\bar{Y}(\bar{P}, t), t) \cdot \hat{n}(\bar{P}, t, t) \quad (11)$$

where  $\hat{n}$  is the unit-normal vector against which  $\bar{v}_n$  was measured. Equation (11) means we can use 8 normal image velocities measured in space and time to recover motion and structure.

Lastly, we can differentiate (1) and (11) to obtain first- and second-order spatial derivatives and first-order temporal derivatives of  $\bar{v}$  and  $\bar{v}_n$ . These derivatives can be estimated via an approximate Taylor series expansion from full/normal image velocity flow fields or measured directly from the time-varying intensity data by some as yet unspecified filtering technique. These derivatives can be used directly in a motion and structure computation.

## 2.2 The Nonuniqueness of the Solutions

Because we are solving nonlinear systems of equations we need to be concerned about the uniqueness of our solution. Hay [1966] was the first to investigate the inherent nonuniqueness of the visual interpretation of a stationary planar surface. He showed that for any planar surface there are at most two sets of motion and structure parameters that give rise to the same image velocity field for that surface. Hay also showed that given two views of such a surface only one unique set of motion and structure parameters was capable of correctly describing the image velocity field. Waxman and Ullman [1983, 1985] carried this result one step further by showing the *dual* nature of the solutions: given one set of motion and structure parameters it is possible to derive a second set in terms of the first analytically. If this second solution is then substituted back into the equations specifying the duality, the first solution is obtained. Given one set of motion and structure parameters,  $\bar{u}_1$ ,  $\bar{\alpha}_1$ , and  $\bar{\omega}_1$  at  $\bar{Y}(\bar{P}, t)$ , we can derive expressions for the dual solution,  $\bar{u}_2$ ,  $\bar{\alpha}_2$ , and  $\bar{\omega}_2$ , at  $\bar{Y}(\bar{P}, t)$  as

$$\begin{aligned} & \bar{u}_2(\bar{Y}(\bar{P}, t), t; t) \\ &= \bar{\alpha}_1(\bar{P}, t; t) \frac{\bar{u}_1(\bar{Y}(\bar{P}, t), t; t) \cdot \bar{Y}(\bar{P}, t)}{\bar{\alpha}_1(\bar{P}, t; t) \cdot \bar{Y}(\bar{P}, t)}, \quad (12a) \end{aligned}$$

$$\bar{\alpha}_2(\bar{P}, t; t) = \frac{\bar{u}_1(\bar{Y}(\bar{P}, t), t; t)}{\|\bar{u}_1(\bar{Y}(\bar{P}, t), t; t)\|_2} \quad (12b)$$

and

$$\begin{aligned} & \bar{\omega}_2(t; t) = \bar{\omega}_1(t; t) \\ & + \begin{pmatrix} 0 & -\alpha_{13}(\bar{P}, t; t) & \alpha_{12}(\bar{P}, t; t) \\ \alpha_{13}(\bar{P}, t; t) & 0 & -\alpha_{11}(\bar{P}, t; t) \\ -\alpha_{12}(\bar{P}, t; t) & \alpha_{11}(\bar{P}, t; t) & 0 \end{pmatrix} \\ & \times \frac{\bar{u}_1(\bar{Y}(\bar{P}, t), t; t)h(\bar{Y}(\bar{P}, t))}{\bar{\alpha}_1(\bar{P}, t; t) \cdot \bar{Y}(\bar{P}, t)} \quad (12c) \end{aligned}$$

$\bar{\alpha}_1 = (\alpha_{11}, \alpha_{12}, \alpha_{13})$  in (12c). Obviously, when

$$\bar{\alpha}(\bar{P}, t; t) = \frac{\bar{u}(\bar{Y}(\bar{P}, t), t; t)}{\|\bar{u}(\bar{Y}(\bar{P}, t), t; t)\|_2}$$

the solution is unique as the dual solution reduces to the first solution. Subbarao and Waxman [1985] have also shown the uniqueness of the motion and structure parameters over time.

These theoretical results suggest that the possibility of multiple (nondual in the spatial case) solutions is nonexistent. However, they hold only when the whole flow field is considered. It is possible for two distinct sets of four image velocity fields to have four common image points at the four times with the same image velocity values. Hence, the analysis of the four image velocities may give rise to any of the sets of motion and structure parameters having those four image velocities in common. See [Barron 1984 and 1988] and below for examples of multiple solutions. It is unlikely this phenomenon would be observed as the amount of input data is increased. In this case, we can be reasonably confident that the solution obtained in time is unique.

## 2.3 Singularity

In cases where there is no translation it is essential that this be known; the reduced system of equations for  $\bar{\omega}$  is well conditioned and robust ( $\bar{\alpha}$  cannot be recovered unless there is translation). However, we are concerned with the general system of equations for  $\bar{u}$ ,  $\bar{\alpha}$ , and  $\bar{\omega}$ , which is singular if  $\bar{u} = (0, 0, 0)$  and is treated as an unknown. In fact, when  $\bar{u} \ll \bar{\omega}$ , its condition number becomes very large; a relatively small input error can cause large changes in the output. Also, Fang and Huang [1984] and others have shown that the solution does not exist if image velocities at three or more collinear 3-D points are used (as the determinant of the Jacobian



matrix,  $J$ , is 0). We have also observed that the solution cannot be obtained when two pairs of time-varying image velocities resulting from the 3-D motion of two 3-D points on the same planar surface are used. The image motion of the points can be caused by an infinite number of motion and structure combinations. As well, there are particular motion and structure combinations that cannot be recovered at one time. For example, if  $\vec{U} = (0, 0, a)$ ,  $\vec{\alpha} = (0, 0, 1)$  and  $\vec{\omega} = (0, b, 0)$  at time 0, then for an arbitrary set of constants  $(a, b)$  the motion is singular.

### 3 Experimental Preliminaries

In this section we present details about the solution technique and sensitivity analysis.

#### 3.1 The Solution Technique

We solve the nonlinear system of equations

$$\vec{f} = \vec{f}_C - \vec{f}_M = 0 \quad (13)$$

where  $\vec{f}_M$  are the measured input (full/normal) image velocities or (full/normal) Taylor series coefficients and  $\vec{f}_C$  are the corresponding (full/normal) image velocities or (full/normal) Taylor series coefficients computed using the current estimate of the motion and structure parameters. We use Newton's method to solve equation (13), hence we need to compute a Jacobian matrix

$$J = \begin{bmatrix} \frac{\partial f_i}{\partial s_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_{Ci}}{\partial s_j} \end{bmatrix} \quad i = 1, \dots, m, \quad j = 1, \dots, n \quad (14)$$

where  $n$  is 8 as  $\vec{s} = (\vec{u}, \vec{\alpha}, \vec{\omega})$  consists of nine components but only eight of them are independent—that is, since  $\vec{\alpha}$  is normalized to 1,  $\|\vec{\alpha}\|_2 = 1$ , only two of its components are independent; and  $m \geq n$  specifies the number of inputs. To enforce the  $\vec{\alpha}$  normalization constraint we replace the three  $\vec{\alpha}$  parameters with two  $\vec{a}$  parameters. We compute three mutually perpendicular normalized vectors,  $\vec{g}_1$ ,  $\vec{g}_2$ , and  $\vec{g}_3$ , where, for convenience only,  $\vec{g}_3$  is set to the correct  $\vec{\alpha}$  value. We then replace  $\vec{\alpha}$  by  $\vec{a} = (a_1, a_2)$ :

$$a_i = \frac{\vec{\alpha} \cdot \vec{g}_i}{\vec{\alpha} \cdot \vec{g}_3} \quad i = 1, 2 \quad (15)$$

Given  $\vec{a}$ , we can reconstruct  $\vec{\alpha}$ :

$$\vec{\alpha} = a_1 \vec{g}_1 + a_2 \vec{g}_2 + \vec{g}_3 \quad (16)$$

Elements of  $J$  corresponding to  $a_i$  are computed as

$$\begin{pmatrix} \frac{\partial f_i}{\partial a_1} & \frac{\partial f_i}{\partial a_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial f_i}{\partial \alpha_1} & \frac{\partial f_i}{\partial \alpha_2} & \frac{\partial f_i}{\partial \alpha_3} \end{pmatrix} \cdot (\vec{g}_1, \vec{g}_2) \quad (17)$$

If we assume pure translation, we compute  $J_{m \times 5}$ , that is, we do not compute  $\partial f_i / \partial \omega_i$ ,  $i = 1, \dots, 3$ . Newton's method also requires an initial guess,  $\vec{s}_0$ , to start the convergence calculation. That is, we iterately solve

$$J \vec{h} = \vec{f} \quad (18)$$

where, at the  $i$ th iteration  $\vec{h} = \vec{s}_i - \vec{s}_{i-1}$ , until convergence is obtained.

#### 3.2 The Sensitivity Analysis

It is well known in the computer vision community that motion and structure computation is very sensitive to noisy input. The problem is further compounded by the fact that image velocity may not be entirely due to the 3-D velocity field [Horn 1986; Verri and Poggio 1987], since variation in the image brightness patterns can be due to other phenomena as well, such as changes in lighting conditions and specular points. It is possible to discriminate between the 2-D motion field that is a direct result only of the 3-D velocity and the optic flow field which is a measure of the image intensity variation in space and time. As a result of these problems, some authors, for example, Thompson and Kearney [1986] and Verri and Poggio [1987], argue against the quantitative use of image velocity information. In this article the discrepancy between a motion field and the measured optic flow field is considered as another source of input error.

Recently, Waxman et al. [1987] and Subbarao [1986] have presented closed-form solutions to motion and structure (at one time) even if the surface is nonplanar. As Subbarao [1986] notes, any such algorithm's error behavior can be predicted analytically. Hence, a numerical analysis is not needed because it is redundant. However, since we are solving for motion and structure in space and time we use  $S_M$  and  $T_M$  in our image velocity equation. As a result, this equation is highly nonlinear and as of yet we have not been able to derive a closed-form solution for it. Thus, an analytic analysis is not possible and we instead conduct a numerical analysis. Since error behavior for a particular set of motion and structure parameters is inherent in the environment under which recovery is being attempted and not the particular algorithm being used, results of our error

analysis should (roughly) hold for all similar motion and structure algorithms.

Previous error analysis has consisted almost entirely of running particular motion and structure algorithms with random noise in the input, see for example, Waxman and Ullman [1983, 1985]. Here we present *average* and *worst* case error results for both minimal and least squares input. Rather than compute the percent output error given  $X\%$  input error we compute error amplification factors: given a certain size input error, what is the size of the output error? If  $\vec{\Delta}f$  is the input error and  $\vec{\Delta}s$  is the output error, then the  $L_2$  error amplification factor is simply

$$\rho = \frac{\|\vec{\Delta}s\|_2}{\|\vec{\Delta}f\|_2} \quad (19)$$

If  $\Gamma = \|\vec{f}\|_2/\|\vec{s}\|_2$ , where  $\vec{f}$  is the correct input vector and  $\vec{s}$  is the correct output (the solution) then  $X\%$  input error causes  $\rho\Gamma X\%$  output error.

We also considered *best* case error for minimal input in [Barron 1988]. These results, which are not reported here, are quite good, with error amplifications typically 0.2 or less for most motions. Of course, in the least squares case best case error corresponds to a residual vector of size 0, which occurs for both error-free input and erroneous input in some unspecified direction.

**3.2.1 What Constitutes a Feasible Motion and Structure Computation?** At this point in the article we state the criteria by which we judge whether a motion and structure computation is viable. We believe a feasible motion and structure algorithm should consistently produce worst case error amplification factors of 3 or less. 3 may seem arbitrarily chosen but higher  $\rho$  values such as 10 would be too big (yielding no useful information) while we may not be able to obtain lower  $\rho$  values for a large range of motions. For  $\Gamma \approx 1$ , the 10% worst-case error in  $\vec{f}$  would then produce at most 30% error in  $\vec{s}$ . Typically,  $\Gamma$  ranges from 0.5 to 2.0 for the motions we have examined in this article. We have chosen 10% error in the input as the maximum allowable. Heeger [1987] and Fleet and Jepson [1990] have presented some encouraging results in this regard for the measurement of image velocities. Still, 30% output error is not that useful, except to give a general idea about the motion. For an autonomous vehicle we probably need  $\rho \leq 1$  if the computed motion and structure parameters are to be useful. As well, we desire a robust calculation: the condition number of the Jacobian matrix,  $\kappa$ , should be small, in the thousands or less, at most (an arbitrary

cutoff). Obviously, there is a strong correlation between  $\kappa$  and  $\rho$ . Lastly, we note that the use of worst case input in our investigation is important: if we can show that the motion and structure algorithm is feasible for worst case input error then it is also feasible for all types of other error. Worst case error will seldom occur in the real world.

**3.2.2 SVD Analysis: Computing Average and Worst Case Input Error.** Given a Jacobian  $J_{m \times n}$ ,  $m \geq n$  we can compute its singular value decomposition (SVD) as

$$J_{m \times n} = U_{m \times n}^T D_{n \times n} V_{n \times n} \quad (20)$$

where  $U$  and  $V$  are unitary matrixes and  $D$  is a diagonal matrix. Given the SVD of  $J$  we can compute *worst* case SVD error amplification factors as simply the inverse of the smallest diagonal element of  $D$ . If  $d_i$  is that element, then  $U^T \hat{e}_i$  gives the worst case error direction. We note that this is actually the worst case error direction for the linear system of equations  $J\vec{h} = \vec{f}$  at one iteration (we always compute the best, average, and worst case error using the correct  $\vec{s}$  values) and is only an approximation for the nonlinear system in general: the smaller the input error, the smaller the nonlinear effects and the better the approximation. (When  $m = n$  the inverse of the largest diagonal element of  $D$  gives the *best* case SVD error amplification factors.)

We can also conduct *average* case error analysis if we make enough random error nonlinear simulations. We can compute average case SVD error amplification factors as

$$\frac{\sum_{i=1}^{1000} \|J^+ \hat{n}_i\|_2}{1000} \quad (21)$$

where  $J^+$  is the pseudoinverse of  $J$  and  $\hat{n}_i$  is a random unit vector. The largest random error amplification factor should approach worst case error amplification results.

Even though worst case SVD error is only an approximation to the actual nonlinear worst case error of a nonlinear system of equations experimental observations show it is still much worse than random (average) case error for all the motion and structure setups we examined in this article. As well, SVD analysis of  $J$  allows us to *predict* average and worst case performance for a nonlinear system of equations without performing nonlinear simulation. Hence, SVD analysis is a valuable tool in our sensitivity investigation.

We can analyze the feasibility of motion and structure using full image velocities, normal image velocities, full Taylor series coefficients and normal Taylor series coefficients; we compute the appropriate Jacobian  $J$  for each type of input. We can also estimate Taylor series coefficients from image velocities, that is,  $A\vec{g} = \vec{f}$ , where  $\vec{f}$  are the image velocities and  $\vec{g}$  are the estimated Taylor series coefficients. SVD analysis of  $J_{\text{est}} = A^{-1}J$  allows us to compare error amplification behavior with  $J$  computed for Taylor series coefficients.

In order to compare worst case predicted SVD error amplification factors with nonlinear simulation results we compute the worst case error direction as described above and then add  $X\%$  relative worst case error to the input by scaling the worst case error so that the image velocity or Taylor series coefficient pair with the largest ratio of error size to the magnitude of the pair has  $X\%$  error; hence all other pairs have  $\leq X\%$  error in them. In the case of normal image velocities and Taylor series coefficients we compute  $X\%$  relative error by scaling the error vector so that the largest ratio of error to normal magnitude is  $X\%$ .

**3.2.3 Computing Mean/Difference Error in Image Velocities.** Given four image velocities,  $\vec{f}$ , we can compute a matrix,  $T$ , that transforms the image velocities into their mean and differences, that is,  $T\vec{c} = \vec{f}$ , where  $\vec{c}$  is a vector consisting of two mean coefficients and six difference coefficients. By orthogonalizing  $T$ ,  $T^{-1}J$  and  $J$  produce the same best and worst case error amplification factors, although we use  $\vec{c}$  in the former case and  $\vec{f}$  in the latter case as the input. We perform SVD analysis on  $T^{-1}J$  to compute best and worst case mean and difference error amplification factors (see [Barron 1988] for full details).

**3.2.4 Computing Mean/Derivative Error in Taylor Series Coefficients.** Again, we can perform SVD analysis on the Taylor series  $J$  to obtain best and worst case error when error is added to the image velocity only or is added to image velocity derivatives only. Full details are in [Barron 1988].

### 3.3 Computing Initial Guess Error

We add random error to the correct solution  $\vec{s} = (\vec{u}, \vec{\alpha}, \vec{\omega})$  to obtain the initial guess  $\vec{s}_0$  required to start the convergence calculation in nonlinear simulation. We compute  $X\%$  initial guess error in  $\vec{s}$  by simply adding

$X\%$   $L_2$  random error for each of  $\vec{u}$ ,  $\vec{\alpha}$ , and  $\vec{\omega}$  separately. This means that if one set of parameters is much larger than another, the latter's initial-guess error is not dominated by the larger set of parameters.

### 3.4 Experimental Motions and Surfaces

In choosing what motion and structure parameters we should investigate we were guided by two principles: the parameters should be realistic given our assumptions or they should have been previously reported in other work.

Table 1 shows the parameters for eight realistic motions (and their duals at time 0) that we have devised. We show the  $\vec{\alpha}$  values as unnormalized to keep them as simple integers. When  $\vec{\omega} \neq (0, 0, 0)$ , we have analyzed the motions for both type 1 and type 2 observer motion; hence motions 5 and 7 are type 1 motion; motions 6 and 8 are type 2 motion. The solution point  $\vec{Y}_s = (20, 20)$  is the image point where  $\vec{s}$  is computed.<sup>2</sup> If  $\vec{Y}_s$  were changed from (20, 20) to (0, 0), then motions D3, D4, D7, and D8 would be singular as  $\vec{u} = (0, 0, 0)$ . Motions 1–4 are analyzed assuming pure translation (i.e.,  $\vec{\omega} = (0, 0, 0)$  is known). We can analyze motions 2–4 assuming general motion as well. However, motion 1 is singular when general motion is assumed, regardless of the time at which the inputs are measured. Motions 5 and 6 are also singular, if all the input is obtained at time 0, but can be analyzed when the input is measured over time.

The second set of motion and structure parameters considered here are the planar motions of Waxman and Ullman [1983, 1985]. Table 2 shows the 5 motions (labeled W6 to W10 to correspond to examples 6 to 10 in Waxman and Ullman [1983]) and their duals, DW6 to DW10. These are not realistic, everyday, motions one might expect an autonomous vehicle to undergo. We analyze these motions at time 0 only and use  $\vec{Y}_s = (0, 0)$ , as did Waxman and Ullman. Note that motions DW7 and DW9 are singular since  $\vec{u} = (0, 0, 0)$ . For all the motions described in tables 1 and 2,  $X_3$  is 2000. Using equation (5) we can always compute  $\vec{U}$  given  $\vec{u}$ ,  $Y$ , and  $X_3$ .

We examine motion and structure for various environmental setups. The quantities that describe the environmental setup include  $\vec{Y}_s$ , the solution point, the *spatial extent*, and the *temporal extent*. We compute the spatial extent of a set of image points where the input is measured as the diagonal angle of the smallest rectangle containing all the image points (see figure 6). We compute the temporal extent,  $0-t$ , of a set of image points

Table 1. Original and dual solutions when  $\vec{Y}_s = (20, 20)$ .

Motion	$\vec{s}$	$\vec{s}_D$	Motion
1	$\vec{u} = (0, 0, 0.496976)$ $\vec{\alpha} = (0, 0, 1)$ $\vec{\omega} = (0, 0, 0)$	$\vec{u}_D = (0, 0, 0.496976)$ $\vec{\alpha}_D = (0, 0, 1)$ $\vec{\omega}_D = (0, 0, 0)$	D1
2	$\vec{u} = (0, 0, 0.496976)$ $\vec{\alpha} = (1, 0, 1)$ $\vec{\omega} = (0, 0, 0)$	$\vec{u}_D = (0.460963, 0, 0.460963)$ $\vec{\alpha}_D = (0, 0, 1)$ $\vec{\omega}_D = (0, -0.463768, 0)$	D2
3	$\vec{u} = (0, 0.496976, 0)$ $\vec{\alpha} = (0, 0, 1)$ $\vec{\omega} = (0, 0, 0)$	$\vec{u}_D = (0, 0, 0.038826)$ $\vec{\alpha}_D = (0, 1, 0)$ $\vec{\omega}_D = (-0.5, 0, 0)$	D3
4	$\vec{u} = (0, 0.496976, 0)$ $\vec{\alpha} = (1, 0, 1)$ $\vec{\omega} = (0, 0, 0)$	$\vec{u}_D = (0.036013, 0, 0.036013)$ $\vec{\alpha}_D = (0, 1, 0)$ $\vec{\omega}_D = (-0.463768, 0, 0.463768)$	D3
5, 6	$\vec{u} = (0, 0, 0.496976)$ $\vec{\alpha} = (0, 0, 1)$ $\vec{\omega} = (0, 0.2, 0)$	$\vec{u}_D = (0, 0, 0.496976)$ $\vec{\alpha}_D = (0, 0, 1)$ $\vec{\omega}_D = (0, 0.2, 0)$	D5, D6
7, 8	$\vec{u} = (0, 0.496976, 0)$ $\vec{\alpha} = (1, 0, 1)$ $\vec{\omega} = (0.2, 0, 0)$	$\vec{u}_D = (0.036013, 0, 0.036013)$ $\vec{\alpha}_D = (0, 1, 0)$ $\vec{\omega}_D = (-0.263768, 0, 0.463768)$	D7, D8

Table 2. Waxman and Ullman's planar motions and their duals at  $\vec{Y}_s = (0, 0)$ .

Motion	$\vec{s}$	$\vec{s}_D$	Motion
W6	$\vec{u} = (6, 5, 4)$ $\vec{\alpha} = (-1, 0, -1)$ $\vec{\omega} = (3, 2, 1)$	$\vec{u}_D = (4, 0, 4)$ $\vec{\alpha}_D = (-1.5, -1.25, -1)$ $\vec{\omega}_D = (-2, 4, 6)$	DW6
W7	$\vec{u} = (6, 5, 0)$ $\vec{\alpha} = (-1, 0, -1)$ $\vec{\omega} = (3, 2, 1)$	$\vec{u}_D = (0, 0, 0)$ $\vec{\alpha}_D = (1.2, 1, 0)$ $\vec{\omega}_D = (-2, 8, 6)$	DW7
W8	$\vec{u} = (1, 2, 3)$ $\vec{\alpha} = (-1, -2, -1)$ $\vec{\omega} = (4, 5, 6)$	$\vec{u}_D = (3, 6, 3)$ $\vec{\alpha}_D = (-0.333, -0.667, -1)$ $\vec{\omega}_D = (8, 3, 6)$	DW8
W9	$\vec{u} = (1, 2, 0)$ $\vec{\alpha} = (-1, -2, -1)$ $\vec{\omega} = (4, 5, 6)$	$\vec{u}_D = (0, 0, 0)$ $\vec{\alpha}_D = (1, 2, 0)$ $\vec{\omega}_D = (2, 6, 6)$	DW9
W10	$\vec{u} = (6, 5, 4)$ $\vec{\alpha} = (0, 0, -1)$ $\vec{\omega} = (3, 2, 1)$	$\vec{u}_D = (0, 0, 4)$ $\vec{\alpha}_D = (-1.5, -1.25, -1)$ $\vec{\omega}_D = (-2, 8, 1)$	DW10

where the inputs are measured as the times  $0, t/3, 2t/3$ , and  $t$  for full image velocity or Taylor series coefficient input and as the times  $0, t/7, 2t/7, 3t/7, 4t/7, 5t/7, 6t/7$ , and  $t$  for normal image velocity or Taylor series coefficient input. In general, the temporal extent depends on the times we measure image velocity information from the image data. We use uniform time intervals here only for convenience.

### 3.5 Monocular Flow Field Examples

In this section we show the image velocity and its derivative flow fields for one of Waxman and Ullman's motions, W6. For a planar surface at depth  $X_3 = 2000$  with surface gradient  $\vec{\alpha} = (-1, 0, -1)$  the image velocity flow field generated for an observer moving with translation,  $\vec{U} = (12000, 10000, 8000)$  and rotation,

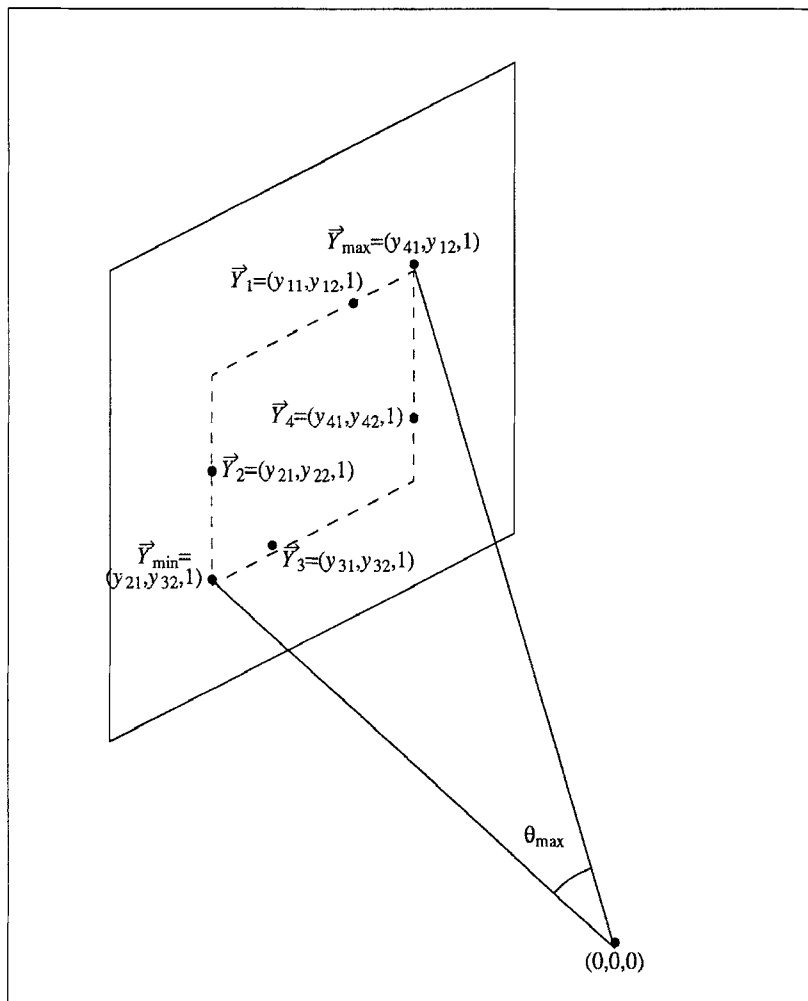


Fig. 6. We compute the spatial extent of the image points used as the maximum diagonal angle,  $\theta_{\max}$ , of the largest rectangle containing all the image points, i.e., the angle formed by the lines joining  $\vec{Y}_{\min}$  and  $\vec{Y}_{\max}$  at the lower left and upper right corners of the square respectively with the origin at  $(0, 0, 0)$ .  $\vec{Y}_{\min}$  is computed using the minimum  $y_1$  and  $y_2$  coordinates of  $\vec{Y}_1$ ,  $\vec{Y}_2$ ,  $\vec{Y}_3$ , and  $\vec{Y}_4$  while  $\vec{Y}_{\max}$  is computed using the maximum  $y_1$  and  $y_2$  coordinates of those same four image points.

$\vec{\omega} = (3, 2, 1)$ , is shown in figure 7. Figures 8 to 11 show the  $\partial \vec{v} / \partial y_1$ ,  $\partial \vec{v} / \partial y_2$ ,  $\partial^2 \vec{v} / \partial y_1 \partial y_2$ , and  $\partial \vec{v} / \partial t$  flow fields for W6 respectively.<sup>3</sup> Figure 12 shows the image velocity field for the flow field with the motion and structure parameters:  $\vec{U} = (1000, 0, 0)$ ,  $\vec{\alpha} = (1, 1, 1)$ ,  $\vec{\omega} = (0, -0.5, 0)$  and  $X_3 = 2000$ . This flow field is an example of Regan and Beverley's type of FOE [1982]. Figures 13 to 16 show the corresponding derivative flow fields. Finally, we present image velocity flow fields for motions 5 (type 1 motion) and 8 (type 2 motion) at four times, 0.0, 0.333, 0.67, and 1.0, that is, a temporal extent of 0–1 in figures 17 and 18 respectively. We note, that at time 0, motion 6 is the same as motion 5 and motion 8 is the same as motion 7. The corresponding

sets of flow fields for these two motions are very close to each other at the other times, becoming more different as the time becomes larger. For temporal extent 0–0 motions 5 and 6 are singular: we cannot analyze the flow field at this time because it is impossible to distinguish how much of the image velocity is due to 3-D observer translation and how much is due to the 3-D observer rotation since observer 3-D translation and 3-D rotation velocity are in the same direction. However, if the flow field is sampled for a nonzero temporal extent the motion can be analyzed. This is because the rotational image velocity field is constant over time due to the fixed-axis assumption. Hence, any changes in the flow field over time are due to observer translation

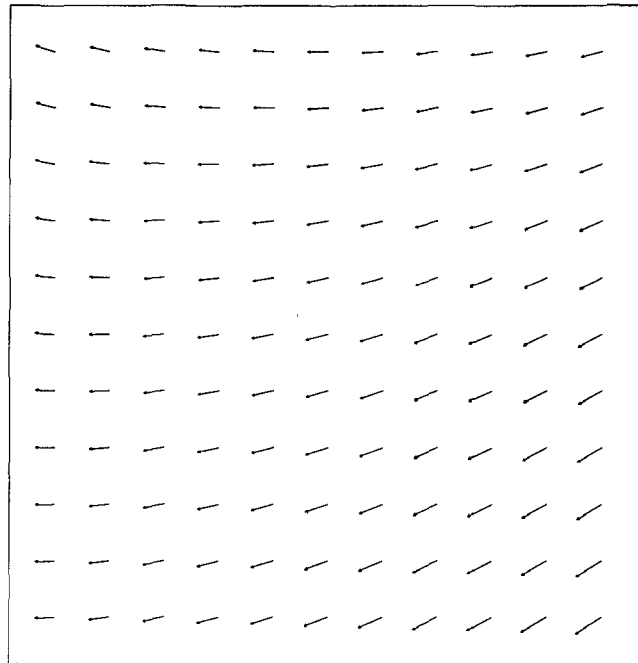


Fig. 7. The full image velocity field for Waxman and Ullman's motion, W6. The motion and structure parameters are  $\vec{u} = (6, 5, 4)$ , unnormalized  $\vec{\alpha} = (-1, 0, -1)$ ,  $\vec{\omega} = (3, 2, 1)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The observer velocity  $\vec{U}$  is (12000, 10000, 8000) and the flow field is generated at time 0. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The image velocities are scaled by a factor of 1.

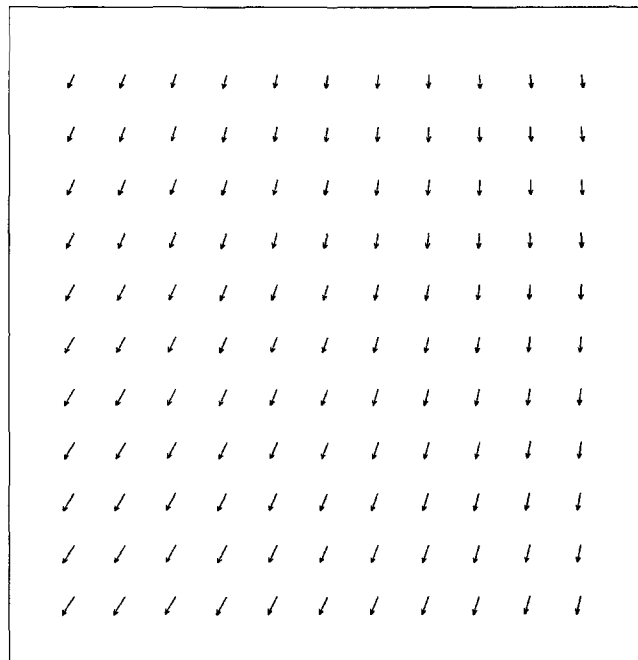


Fig. 8. The first-order spatial image velocity derivative field,  $\partial \vec{v} / \partial y_1$  for Waxman and Ullman's motion, W6. The motion and structure parameters are  $\vec{u} = (6, 5, 4)$ ,  $\vec{\alpha} = (-1, 0, -1)$ ,  $\vec{\omega} = (3, 2, 1)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is (12000, 10000, 8000). The flow field is generated at time 0. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The scale factor is 1.

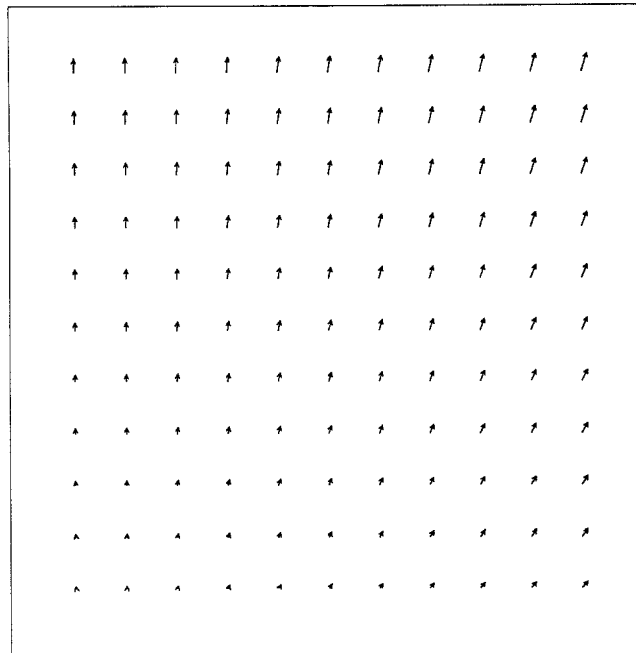


Fig. 9. The first-order spatial image velocity derivative field,  $\partial\vec{v}/\partial y_2$  for Waxman and Ullman's motion, W6. The motion and structure parameters are  $\vec{u} = (6, 5, 4)$ ,  $\vec{\alpha} = (-1, 0, -1)$ ,  $\vec{\omega} = (3, 2, 1)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is (12000, 10000, 8000). The flow field is generated at time 0. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The scale factor is 1.

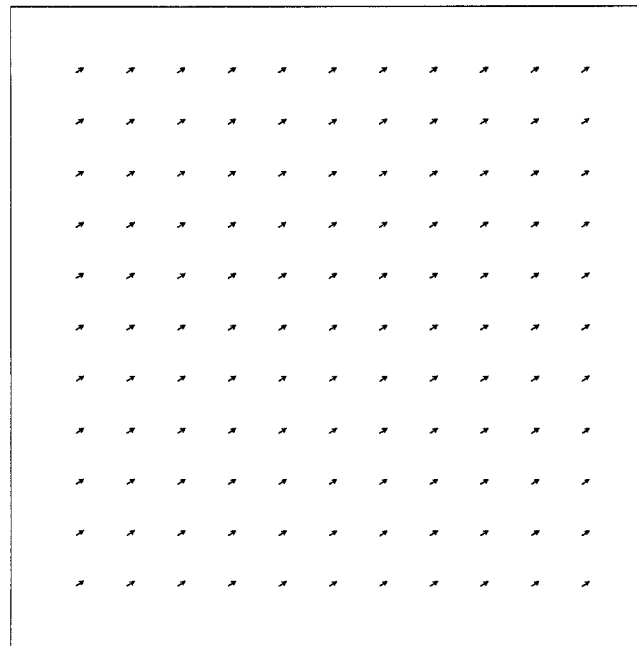


Fig. 10. The second-order spatial image velocity derivative field,  $\partial^2\vec{v}/\partial y_1\partial y_2$  for Waxman and Ullman's motion, W6. The motion and structure parameters are  $\vec{u} = (6, 5, 4)$ ,  $\vec{\alpha} = (-1, 0, -1)$ ,  $\vec{\omega} = (3, 2, 1)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is (12000, 10000, 8000). The flow field is generated at time 0. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The scale factor is 1.

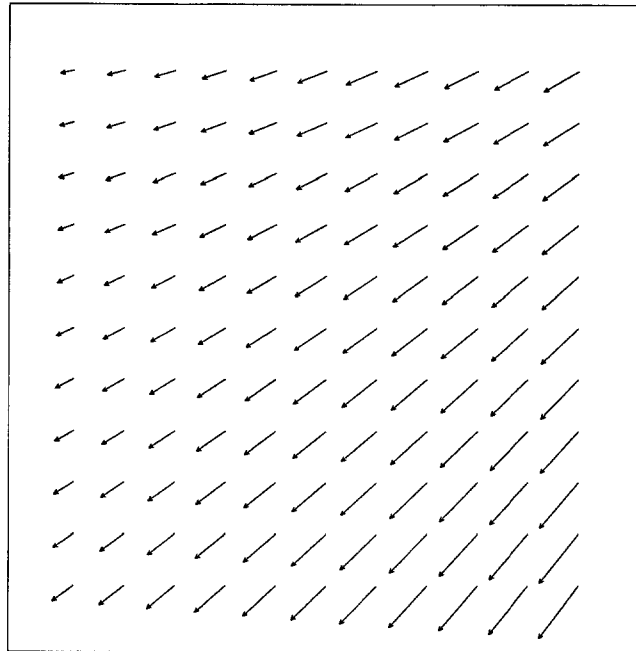


Fig. 11. The first-order temporal image velocity derivative field,  $\partial \vec{v} / \partial t$  for Waxman and Ullman's motion, W6. The motion and structure parameters are  $\vec{u} = (6, 5, 4)$ ,  $\vec{\alpha} = (-1, 0, -1)$ ,  $\vec{\omega} = (3, 2, 1)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is (12000, 10000, 8000). The flow field is generated at time 0. The corner points of the images (in pixels) are: (-100, -100), (-100, 100), (100, 100), and (100, -100). The scale factor is 0.25.

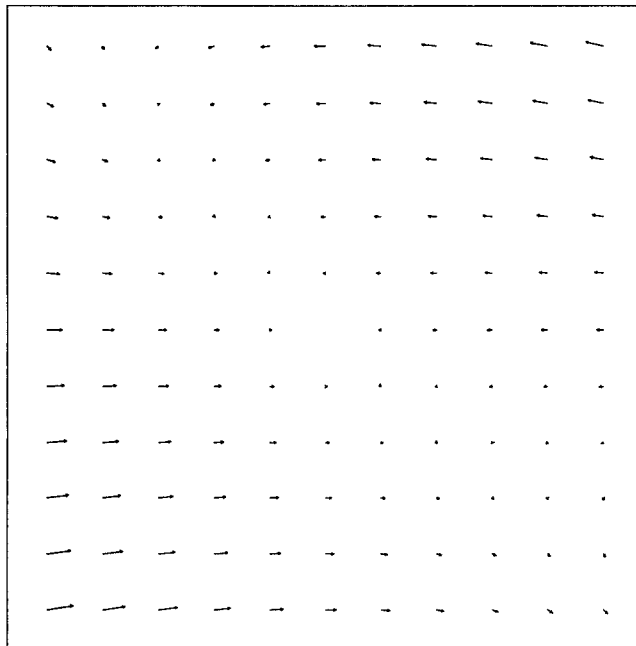


Fig. 12. The full image velocity field for the Regan and Beverley FOE example. The motion and structure parameters are  $\vec{u} = (0.5, 0, 0)$ , unnormalized  $\vec{\alpha} = (1, 1, 1)$ ,  $\vec{\omega} = (0, -0.5, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The observer velocity  $\vec{U}$  is (1000, 0, 0) and the flow field is generated at time 0. The corner points of the images (in pixels) are: (-100, -100), (-100, 100), (100, 100), and (100, -100). The image velocities are scaled by a factor of 20.



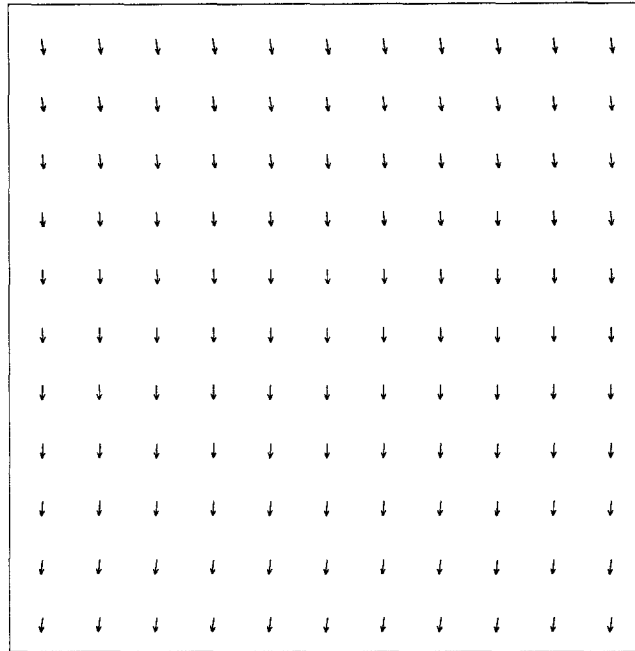


Fig. 13. The first-order spatial image velocity derivative field,  $\partial v/\partial y_1$ , for Regan and Beverley's FOE example. The motion and structure parameters are  $\vec{u} = (0, 0.5, 0)$ ,  $\vec{\alpha} = (1, 0, 1)$ ,  $\vec{\omega} = (0.2, 0, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is  $(0, 1000, 0)$ . The flow fields are for type 2 motion at time 0.333. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The scale factor is 10.

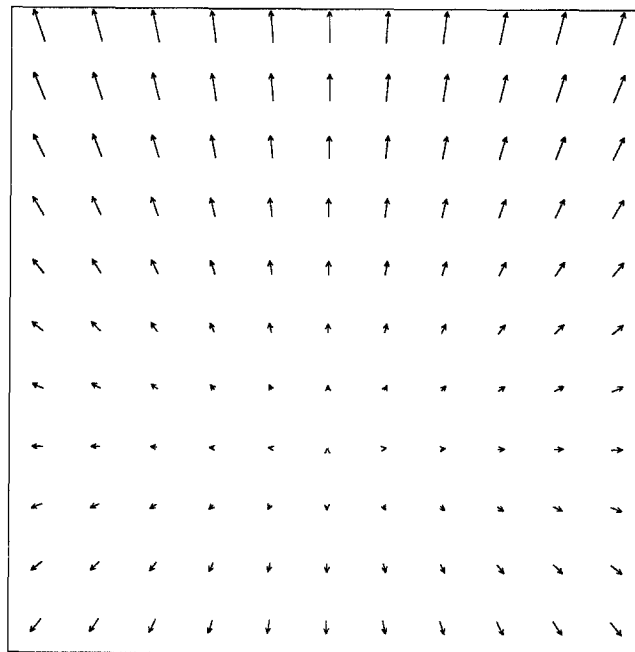


Fig. 14. The first-order spatial image velocity derivative field,  $\partial v/\partial y_2$ , for Regan and Beverley's FOE example. The motion and structure parameters are  $\vec{u} = (0, 0.5, 0)$ ,  $\vec{\alpha} = (1, 0, 1)$ ,  $\vec{\omega} = (0.2, 0, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is  $(0, 1000, 0)$ . The flow fields are for type 2 motion at time 0.667. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The scale factor is 50.

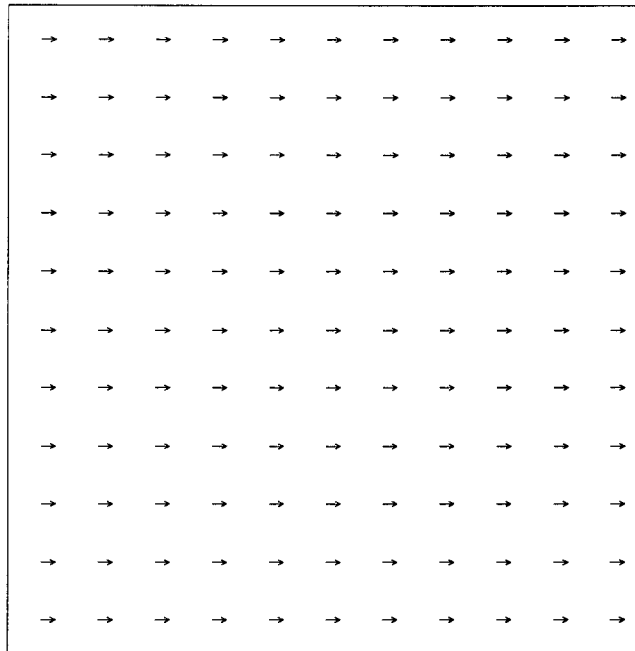


Fig. 15. The second-order spatial image velocity derivative field,  $\partial^2 v / \partial y_1 \partial y_2$  for Regan and Beverley's FOE example. The motion and structure parameters are  $\vec{u} = (0, 0.5, 0)$ ,  $\vec{\alpha} = (1, 0, 1)$ ,  $\vec{\omega} = (0.2, 0, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is  $(0, 1000, 0)$ . The flow fields are for type 2 motion at time 1.0. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The scale factor is 25.

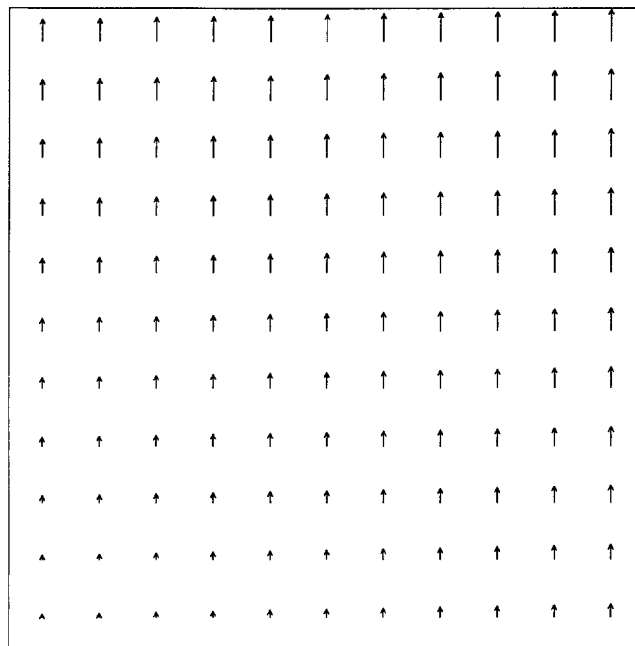


Fig. 16. The first-order temporal image velocity derivative field,  $\partial v / \partial t$  for Regan and Beverley's FOE example. The motion and structure parameters are  $\vec{u} = (0, 0.5, 0)$ ,  $\vec{\alpha} = (1, 0, 1)$ ,  $\vec{\omega} = (0.2, 0, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  in pixels. The correct observer velocity  $\vec{U}$  is  $(0, 1000, 0)$ . The flow fields are for type 2 motion at time 1.0. The corner points of the images (in pixels) are:  $(-100, -100)$ ,  $(-100, 100)$ ,  $(100, 100)$ , and  $(100, -100)$ . The scale factor is 100.

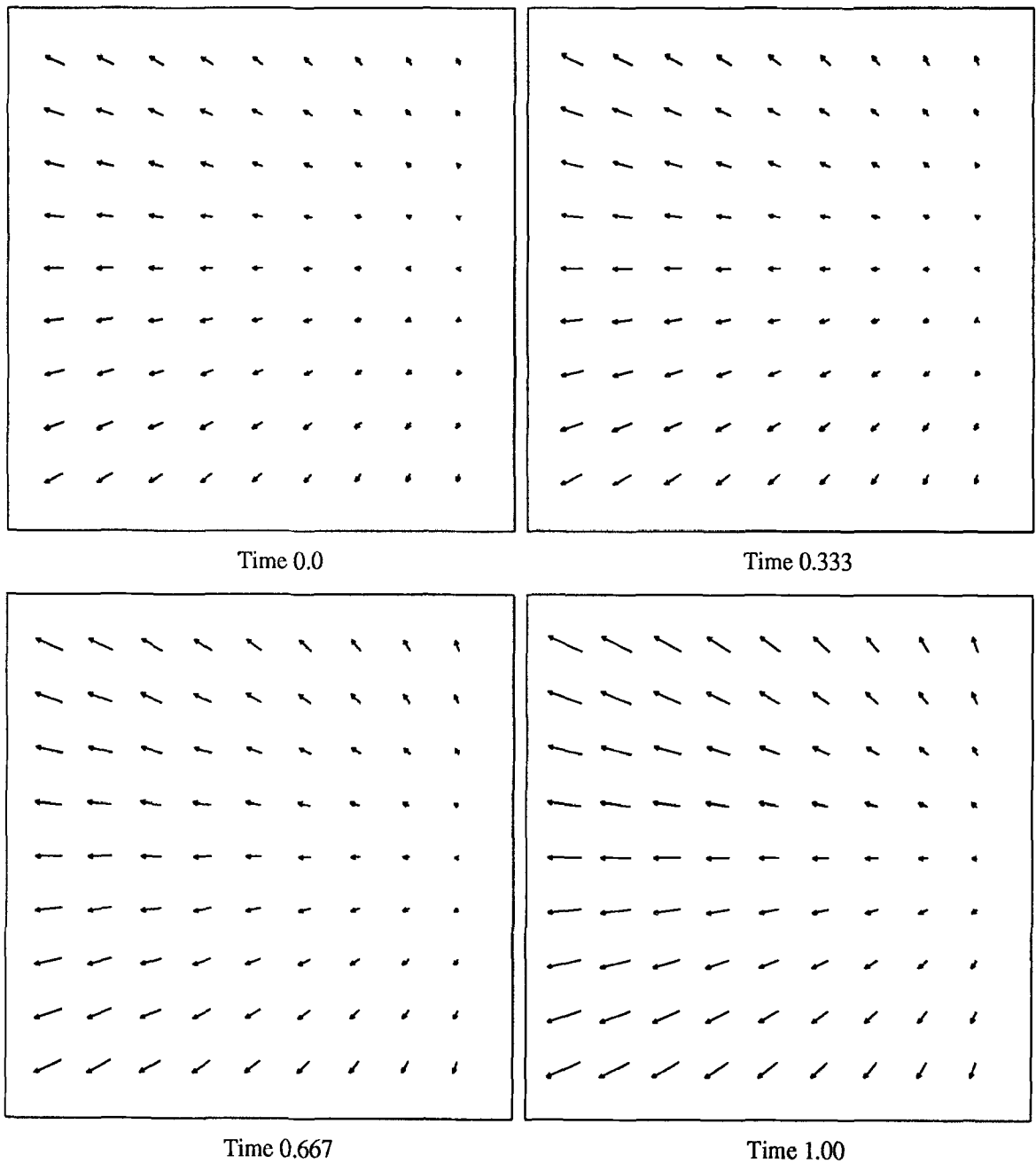


Fig. 17. The full image velocity fields for temporal extent 0-1 for motion 5. The motion and structure parameters are  $\bar{u} = (0, 0, 0.5)$ ,  $\bar{\alpha} = (0, 0, 1)$ ,  $\bar{\sigma} = (0, 0.2, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\bar{Y} = (0, 0)$  (in pixels). The correct observer velocity  $\bar{U}$  is  $(0, 0, 1000)$ . The flow fields are for type 1 motion at times 0.0, 0.333, 0.667, and 1.0. The corner points of the images (in pixels) are:  $(-80, -80)$ ,  $(-80, 80)$ ,  $(80, 80)$ , and  $(80, -80)$ . The image velocities are scaled by 25.

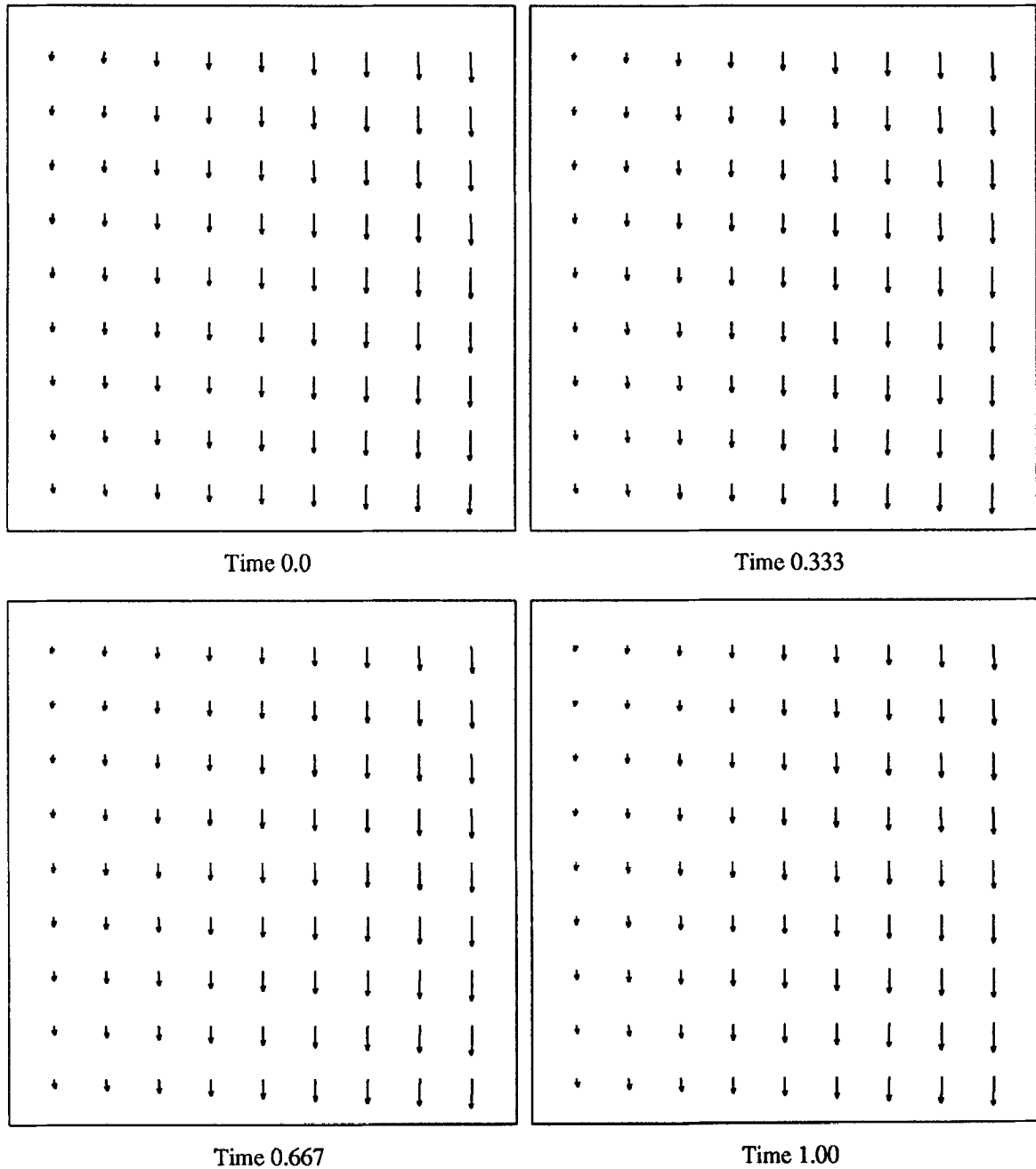


Fig. 18. The full image velocity fields for temporal extent 0-1 for motion 8. The motion and structure parameters are  $\vec{u} = (0, 0.500000, 0)$ ,  $\vec{\alpha} = (0.707107, 0, 0.707107)$ ,  $\vec{\omega} = (0.2, 0, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (0, 0)$  (in pixels). The correct observer velocity  $\vec{U}$  is  $(0, 1000, 0)$ . The flow fields are for type 2 motion at times 0.0, 0.333, 0.667, and 1.0. The corner points of the images (in pixels) are:  $(-80, -80)$ ,  $(-80, 80)$ ,  $(80, 80)$ , and  $(80, -80)$ . The image velocities are scaled by a factor of 25.

only. On the other hand, motions 7 and 8 are analyzable for temporal extent 0–0 as the translational and rotational image velocity components can be distinguished; the translational image velocity is dependent on the 3-D depth of points on the surface (as well as image location) as the surface is not frontoparallel while rotational image velocity just depends on the image location. Hence, the two components of image velocity can be distinguished if image velocities are sampled at four or more different 3-D points. Because  $\vec{\alpha}$  specifies a frontoparallel plane for motions 5 and 6, the 3-D depth of all points on the surface are constant. If  $\vec{\alpha}$  were changed to (1, 0, 1) the motion and structure could be recovered from a spatial distribution of image velocities at time 0.

## 4 Experimental Results

We present qualitative rather than quantitative results here; complete numerical results are in [Barron 1988]. We consider both *minimal* (the least amount of data needed to recover motion and structure) and *least squares* input.

### 4.1 Using Minimal Input

In this section we investigate the use of minimal full/normal image velocity or Taylor series coefficients data in the computation of motion and structure. The two main questions of concern are “How sensitive is the computation to noisy input?” and “Can anything be done to reduce the noise sensitivity?”

**4.1.1 Using Minimal Image Velocity Input.** Since four image velocities constitute minimal input for general motion we measure each image velocity at the four corners of a square centered at  $\vec{Y}_s$  at each of the four times specified by the temporal extent.<sup>4</sup> For each motion we use a temporal extent 0– $t$  where  $t$  is varied from 0 to 1 for motions 1–4, 7, and 8 and 0.3 to 1 for motions 5 and 6. For motions W6–W10, DW6, DW8, and DW10,  $t$  is fixed at 0. If  $\vec{Y}_s = (20, 20)$  the spatial extent is  $30.5^\circ$  while if it is (0, 0) the spatial extent is  $30.9^\circ$ .

Average and worst case SVD results show that the computation of motion and structure is untenable. Indeed, even if the feasibility criterion of a maximum worst case error amplification factor of 3 were increased to 10 most of the motions are still not viable. Worst case error amplification factors ranged from a low of 7 to a high of 317 for motions 1–8 and from 3.4 to 36

for motions W6–DW10. Some of the motions, especially motion 5, had average and worst case error amplification factors in the 100s! However, we are able to make two important observations:

1. The worst case error amplification factors observed from the SVD analysis and the corresponding nonlinear simulations showed good agreement: the SVD amplification factors are good predictors for nonlinear simulation behavior. We used 0% to 1.4% relative worst case image velocity error (because of the high error amplification factors, larger input error usually did not allow a convergence calculation) and 0% initial-guess error in the nonlinear simulations.
2. In general, worst case error amplification factors were at least 2–3 times higher than average case error amplification factors. However, the maximum of the random error amplification factors used in the computation of an average case error amplification factor for a particular motion was quite close to the worst case error amplification factor for that motion. Worst case error often occurred for minimal input. It seems that worst case analysis should be an important feature of any sensitivity analysis.

The second question we posed, “Can anything be done to reduce the noise sensitivity?” can be answered more positively. In particular, we demonstrate that time helps. As the temporal extent is increased, the average and worst case error amplification factors decrease for most of the motions. As well, increasing the spatial extent helps for all motions. Indeed, for some of the motions a spatial extent of  $70^\circ$  (the full image plane) yield error amplification factors less than 3 (or, more commonly, in the 4–6 range).

We are able to report numerous other results:

- (1) Worst case error amplification factors for full image velocities are only slightly larger than worst case difference error amplification factors where error is added to the image velocity differences only: worst case error is almost entirely due to error in the image velocity differences. On the other hand, best case error amplification factors for full image velocities are only slightly smaller than worst case mean error amplification factors where error is added to the single image velocity mean only. This means that large image velocity mean error can be handled with ease. Some types of autonomous vehicle motion, for example, hitting a rut in a road, could introduce this type of error into the image velocity input.

(2) It is reasonable to question the validity of our algorithm's underlying assumptions. As Thompson and Kearney [1986, p. 17] comment:

Unrealistic assumptions are only justifiable when it can be shown that useful answers can be obtained in realistic situations despite violations in the assumptions.

We test the violation of the planarity, rigidity, and same-surface assumptions by perturbing  $\vec{\alpha}$  values in one of three ways: (a) by measuring the image velocities on two halves of a hinged surface, (b) by measuring the image velocities on a spherical surface, and (c) by perturbing the  $\vec{\alpha}$  values randomly. We test violation of the fixed-axis assumption by adding randomly generated error to  $\vec{\omega}$  at each time an image velocity is measured while we test violation of the known-motion-type assumption by adding random error to  $\vec{u}$  and  $X_3$  at each time an image velocity is measured. We perform non-linear simulation, using assumption error scaled to be the same size as worst case image velocity error. The results show that the assumption error amplification factors are usually significantly less than worst case error amplification factors. It seems that we should be more concerned with image velocity accuracy rather than with violation of the algorithm's underlying assumptions.

(3) Another concern that may arise is the effect of conducting the motion and structure computation at various locations in the image plane. We use a  $50 \times 50$  pixel mask (measuring the image velocity at its four corners) centered at 17 image locations symmetrically distributed about the image plane. The results indicate that there is no preferred location for the best possible motion and structure computation. Theoretically, this makes sense as the image velocity accuracy is independent of image location for an ideal machine vision system. That is, we can measure the image velocities just as accurately (or more precisely, just as inaccurately) at the periphery as at the center of the image. Of course, this situation is not comparable to the human eye, where the concentration of processing cells is much more dense at the fovea than in the periphery. Lastly, we consider flow fields that exhibit an FOE (Focus of Expansion). For example, a flow field with the parameters given in table 3 exhibits a FOE at  $\vec{Y} = (0, 0)$  at time 0. This flow field is similar to Regan and Beverley's FOE example [1982, figure 1, p. 195] and is shown in figure 12. It appears that it is just as difficult to recover motion and structure in the flow field's periphery as it is near its FOE.

Table 3. FOE parameters.

$\vec{u} = (0.5, 0, 0)$
$\vec{\alpha} = (1, 1, 1)$
$\vec{\omega} = (0, -0.5, 0)$
$X_3 = 2000$

(4) Just as we have examined the sensitivity of pure translation we can also examine the sensitivity of pure rotation, in particular, when  $\vec{u} = (0, 0, 0)$  is known and  $\vec{\omega} = (0, 0.2, 0)$ . In this case neither  $\vec{u}$  or  $\vec{\alpha}$  can be recovered. Now we need only be concerned with a  $3 \times 3$  linear system of equations (or an  $8 \times 3$  system of equations if we use all four image velocities). Worst case error amplification factors for the  $3 \times 3$  system are less than 3 for spatial extents greater than about  $40^\circ$  while for the  $8 \times 3$  system they are less than 3 for spatial extents greater than about  $20^\circ$ . Recovery of  $\vec{\omega}$  under the pure rotation assumption is feasible. Even greater robustness can be obtained if larger rotational image velocity fields are available.

(5) We consider three situations where computing a restricted set of motion and structure may be realistic:

- (a) If  $\vec{\omega}$  is accurately known, say by some device that measures centrifugal force, then we just need to compute  $\vec{u}$  and  $\vec{\alpha}$ .
- (b) An autonomous vehicle may be able to measure its translation,  $\vec{U}$ , and rotation,  $\vec{\omega}$ , accurately enough to leave us with the problem of computing  $\vec{\alpha}$  from the image velocity field. In order to compute  $\vec{u}$  (using 2.1-5), we must also know  $X_3$  at the solution point  $\vec{Y}_s$ .  $X_3$  might be obtained by some active sensor, for example, a sonar or laser-beam device. We consider this a restricted form of the structure from motion problem (versus the more general motion and structure from motion problem). (Of course, if we know  $\vec{u}$  and  $\vec{\omega}$  for some 3-D planar surface then we can use a spatiotemporal distribution of image velocities on that surface to recover both  $X_3$  and  $\vec{\alpha}$  (effectively recovering  $X_3$  everywhere on the surface). This is an example of motion stereo, see (Barron, Jepson and Tsotsos (1987b, 1989) for more details.)
- (c) An autonomous vehicle's environment is restricted to the corridors of a building. In this case,  $u_2$ ,  $\omega_1$ , and  $\omega_3$  are known to be zero; hence, there are only five independent motion and structure parameters in the general case and only four when pure translation is assumed. Of course, the planarity assumption is satisfied for most of the vehicle's environment, as it is primarily composed of walls, floors, and ceilings.

Results are as follows: For (a) there is a significant reduction in error amplification factors when  $\vec{\omega}$  is assumed known for motions 5–8. Indeed, the error amplification factors range from 6 to 24 when all four image velocities are used in the input. In the case of motions W6–DW10 all motions are solvable, exhibiting worst case error amplification factors less than 3 when all four image velocities are used in the input. For (b), worst case error amplification factors for  $\vec{\alpha}$  are also much less than those for the corresponding sixteen general motions, ranging from 3 to 17 for motions 1–8 and 0.2 to 1 for motions W6 to DW10. Interestingly, best and worst case error amplification factors (for minimal input data) are close for most of the motions. For (c), we examine motions 1–2 and 5–6 under the assumption of planar translation, vertical rotation. Using the minimal amount of image velocity data 2 or 2½ image velocities needed yields worst case error amplification factors (from 16 to 220) that are actually larger than those observed in the initial sensitivity for the corresponding general motions. However, using all four image velocities resulted in lower worst case error amplification factors than for the corresponding general motions.

In all three cases, knowing something about the motion made the motion and structure calculation more viable (although not necessarily feasible).

(6) The last set of results that we report for minimal image velocity input is the effect of the initial guess on the output. In particular, we demonstrate the following properties of our algorithm:

(a) For 0% image velocity error we are able to solve motion and structure for large initial-guess error, up to 100% and more. More interestingly, we observed the presence of multiple (non-dual) solutions for motions 5–7. These multiple solutions produced flow fields distinct from those for the original solution but both sets of flow fields shared four common image velocities at the image locations and times at which the input was measured. Examples of multiple solutions are reported in [Barron 1984 and 1988]. Figure 19 shows the four flow fields generated for solution  $\vec{u} = (0, 0, 0.496976)$ ,  $\vec{\alpha} = (0, 0, 1)$ , and  $\vec{\omega} = (0, 0.2, 0)$  at four times 0, 0.166667, 0.333333, and 0.5 at  $X_3 = 2000$ . Rather than show the flow fields generated by the multiple solution  $\vec{u}_M = (0.303559, -0.527817, 0.532987)$ ,  $\vec{\alpha}_M = (0.087906, 0.163232, 0.982664)$ , and  $\vec{\omega}_M = (-0.550745, -0.115147, 0.020988)$  again at  $X_3 = 2000$ , for the same four times, we show their differences (amplified by a factor of 5) in figure 20.

(b) We also observed the presence of dual solutions for motions W6, W8, W10, and 2. No multiple solutions were observed for these motions; it is unlikely that two distinct nonsingular flow fields would have four common image velocities. Again 0% image velocity error and up to 100% initial-guess error was used in the nonlinear simulations.

(c) Finally, we performed nonlinear simulation, varying worst case image velocity error from 0% to 1.4% for fixed initial-guess errors of 50% and 100%. For most solved runs, we observed that image velocity and initial-guess error are independent of each other. The other solved runs produced multiple solutions in the presence of nonzero image velocity error; these solutions are due to the initial-guess error as discussed in (a).

4.1.2 Using Minimal Normal Image Velocity Input. The recovery of motion and structure from eight normal image velocities measured at eight times is also possible (or from five normal image velocities measured at five times if pure translation is assumed), provided that the flow field exhibits enough structure. Waxman and Ullman [1983, 1985] call this the “aperture problem in the large” versus the usual “aperture problem in the small” [Marr and Ullman 1981]. We measure eight normal image velocities at the four corners of a square and the four midpoints of the square’s sides. The normal image velocity data are generated in one of two ways: by either alternately using the horizontal and vertical components of the full image velocity at each of the eight image points and times or by computing eight unit random vectors,  $\hat{n}$ , and then computing  $\vec{v}_n = (\vec{v} \cdot \hat{n})\hat{n}$  for the  $\vec{v}$  values measured at each of the eight image points and times.

Again, we perform SVD analysis and nonlinear simulation for all 16 motions and find good correlation between predicted and observed worst case error amplification factors. Typically, normal image velocity error amplification factors are about twice as large as the corresponding error amplification factors for full image velocity input when the same spatiotemporal extent is used.

As in the previous section, we also observed multiple solutions (two distinct sets of eight normal image velocity flow fields shared eight common normal image velocities) for motions 1–2 and 5–8. Dual solutions were found for motions 7–8 and W6, W8, and W10 when the temporal extent is 0–0. Full details are in Barron [1988].

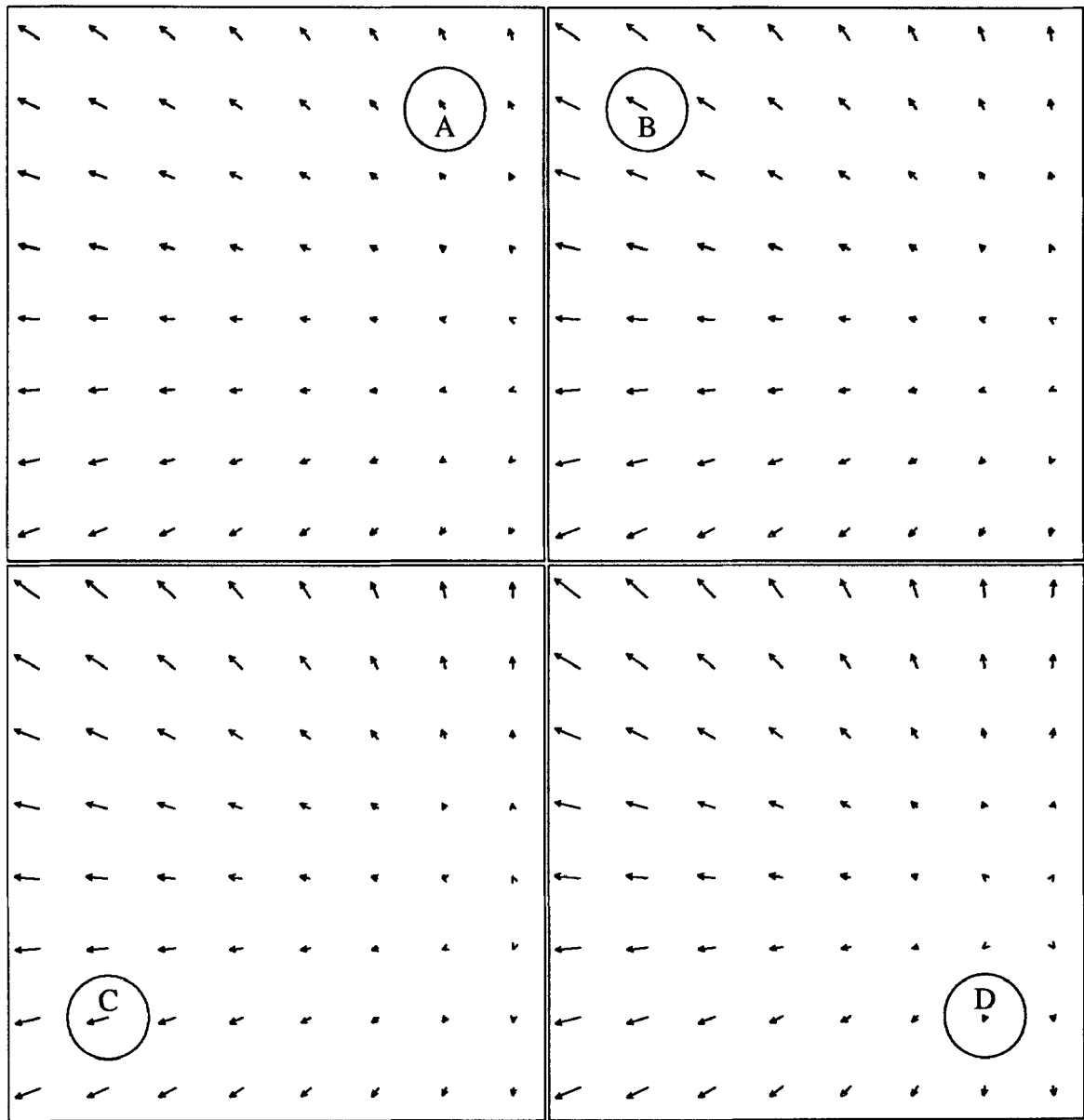


Fig. 19. The full image velocity fields for the correct solution at four different times. The motion and structure parameters are  $\vec{u} = (0, 0, 0.496976)$ ,  $\vec{\alpha} = (0, 0, 1)$ ,  $\vec{\omega} = (0, 0.2, 0)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (20, 20)$ . The correct observer velocity  $\vec{U}$  is  $(0, 0, 1000)$ . The flow fields are for type 2 motion. (a) shows the flow field for time 0 (top left corner), (b) the flow field at time 0.166667 (top right corner), (c) the flow field at time 0.333333 (bottom left corner), and (d) the flow field at time 0.5 (d) bottom right corner. The four common image velocities (with the multiple solution) are  $(-0.078235, 0.121765)$  at  $(70, 70)$  at time 0 (labeled A);  $(-0.266381, 0.154889)$  at  $(-30, 70)$  at time 0.166667 (labeled B);  $(-0.272344, -0.072344)$  at  $(-30, -30)$  at time 0.333333 (labeled C); and  $(-0.028700, -0.073414)$  at  $(70, -30)$  at time 0.5 (labeled D).



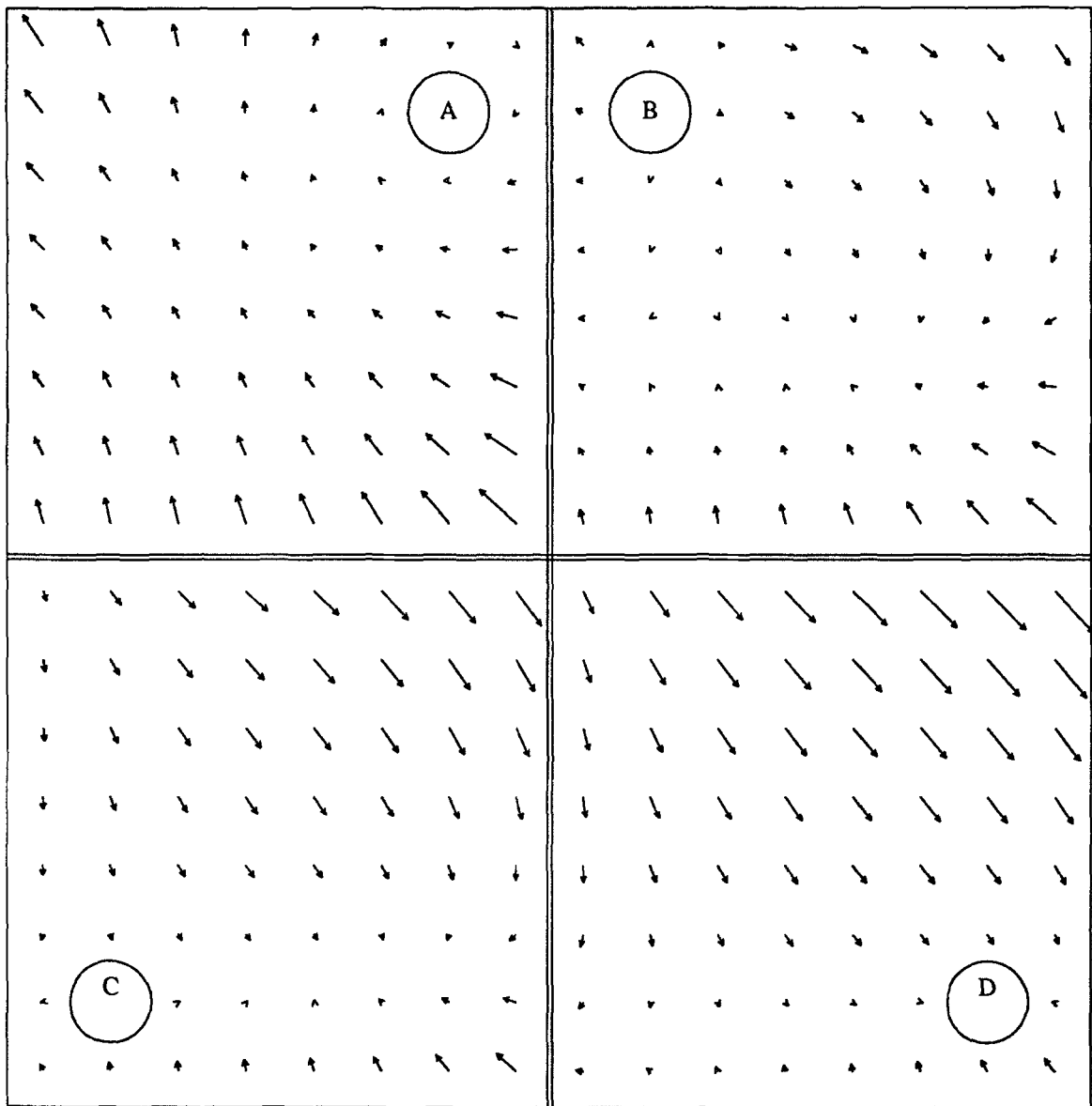


Fig. 20. The difference flow fields, (a) top left corner, (b) top right corner, (c) bottom left corner, and (d) bottom right corner, at the four times, 0, 0.166667, 0.333333, and 0.5 respectively. These flow fields were generated by subtracting the corresponding flow fields of the two solutions at each of the corresponding times. The motion and structure parameters for the second solution are  $\vec{u} = (0.303559, -0.527817, 0.532987)$ ,  $\vec{\alpha} = (0.087906, 0.163232, 0.982664)$ ,  $\vec{\omega} = (-0.550745, -0.115147, 0.020988)$ , and  $X_3 = 2000$  at time 0 for image location  $\vec{Y} = (20, 20)$ . The image velocity differences at (70, 70) in the top left flow field at time 0 (labeled A); (-30, 70) in the top right flow field at time 0.166667 (labeled B); (-30, -30) in the bottom left flow field at time 0.333333 (labeled C); and (70, -30) in the bottom right flow field at time 0.5 are all zero. The image velocity differences at all other points are nonzero. The magnitudes of the image velocity differences were scaled by 5, relative to the image velocity magnitudes shown in figure 19, to clearly show the distinctness of the two sets of flow fields.

#### 4.1.3 Using Minimal Taylor Series Coefficient Input.

In this section we report results obtained using minimal Taylor series coefficient input. Again, we emphasize that we are investigating the sensitivity of motion and structure to such input. Can techniques, such as the velocity functional method [Waxman and Wahn 1984, 1985] or the filters advocated by Egelson [1989] measure the data accurately enough? We can use either

$$\vec{g}_s = \left( \vec{v}, \frac{\partial \vec{v}}{\partial y_1}, \frac{\partial \vec{v}}{\partial y_2}, \frac{\partial^2 \vec{v}}{\partial y_1 \partial y_2} \right) \quad (22)$$

or

$$\vec{g}_t = \left( \vec{v}, \frac{\partial \vec{v}}{\partial y_1}, \frac{\partial \vec{v}}{\partial y_2}, \frac{\partial \vec{v}}{\partial t} \right) \quad (23)$$

to compute motion and structure. At any single time,  $\vec{g}_s$  fully specifies the flow field (see Waxman and Ullman 1983, 1985; and Barron 1988) for a planar surface.  $\vec{g}_s$  coefficients estimated from a single image velocity field are identical to those measured from the image intensity data at the same time. We also note that we can measure the  $\vec{g}_s$  coefficients in time and can measure  $\vec{g}_t$  at one time if we wish.

The first observation we make is that small image velocity error corresponds to much larger Taylor series coefficient error. This means that the error amplification factors resulting from the use of Taylor series coefficient input will be considerably smaller than from the use of image velocity input. In other words a motion and structure calculation can tolerate higher Taylor series coefficient error than image velocity error. This is intuitively pleasing because we shouldn't expect to be able to measure image velocity derivatives as accurately as the image velocities themselves.

Nonlinear simulation results for estimated and measured Taylor series coefficients agree reasonably well with predicted SVD error amplification factors. Unfortunately, except for isolated cases, the computation of motion and structure is still not feasible for minimal Taylor series coefficient input. These isolated cases consist of about half the runs for motions 1–2, all runs for motions 3–4, about a quarter of the runs for motions 7–8, and a few of Waxman and Ullman's planar motions. These feasible cases usually occurred for a spatial extent of  $0^\circ$  and for any of the temporal extents from 0–0 to 0–1 used in the analysis when  $\partial^2 \vec{v} / \partial y_1 \partial y_2$  was used and for large spatiotemporal extents when  $\partial \vec{v} / \partial t$  was used. Even though minimal Taylor series coefficient input is still not generally feasible it is very encouraging.

We define *estimation error* as the difference between estimated and actual Taylor series coefficients for a particular spatiotemporal extent (the spatial extent is always nonzero). For larger temporal extents, this error is quite large—the estimated and measured  $\partial \vec{v} / \partial t$  differ significantly. We obtain nonlinear convergence for estimation errors of up to 40%. The error amplification factors were about 1–2 but could be as high as 5. Of course, estimation error is not worst case error.

A comparison of the worst case error amplification factors that result when  $\partial \vec{v} / \partial t$  or  $\partial^2 \vec{v} / \partial y_1 \partial y_2$  is used to show that generally  $\partial^2 \vec{v} / \partial y_1 \partial y_2$  is better than  $\partial \vec{v} / \partial t$ . However, for these motions,  $\partial \vec{v} / \partial t$  was usually either much larger or much smaller than the other  $\vec{v}$ ,  $\partial \vec{v} / \partial y_1$ , and  $\partial \vec{v} / \partial y_2$  coefficients. The one exception is motion 5, where  $\partial^2 \vec{v} / \partial y_1 \partial y_2$  is much smaller than  $\partial \vec{v} / \partial t$  and is closer to the size of the other coefficients and, as a result, produces slightly smaller worst case error amplification factors than  $\partial \vec{v} / \partial t$ .

In the case of measured Taylor series coefficients we can collect all data at one image point and time (as suggested by Longuet-Higgins and Prazdny [1980] or Waxman and Ullman [1983, 1985]) or at four image points and times. Typically, the use of  $\vec{g}_s$  (see equation (22)) at one point and time produced the best results. Using  $\vec{g}_t$  (see equation (23)) measured for small spatiotemporal extents yields the largest error amplification factors but these are significantly reduced for larger spatiotemporal extents.  $\vec{g}_s$  can also be used for nonzero spatiotemporal extents, producing much better results than those observed from the use of  $\vec{g}_t$ .

We have also examined the feasibility of computing motion and structure using actual Taylor series coefficients at 17 symmetrically distributed image locations around the image. For a flow field arising from pure translation, motion and structure cannot be computed at the FOE if  $\partial \vec{v} / \partial t$  is used; this is because the image velocity remains 0 at this point over time, producing two corresponding rows in the Jacobian,  $J$ , with all zero values. Although calculation of motion and structure is still not feasible in general, for some motions it is. For example, Regan and Beverley's flow field example (described in table 3) produced worst case error amplification factors that were 3 or less when  $\vec{g}_s$  was used. Again, no conclusions about whether computation of motion and structure is better at the image periphery or at its center can be drawn when using Taylor series coefficients.

Two further comments concerning the use of measured Taylor series coefficients are in order. First, we

did not observe any multiple solutions even when 100% initial guess error was used, although all runs solved. Second, we performed a mean and derivative analysis similar to the mean and difference analysis reported above. Worst mean error amplification factors are quite small (less than 0.9 in all cases). This means that a motion and structure computation is insensitive to shifts in the image velocity fields. For small spatiotemporal extents, worst case derivative error amplification factors approach worst case overall error amplification factors. However, as the spatiotemporal extent increases, worst case derivative error amplification factors become significantly smaller than worst case overall error amplification factors. It seems that mean and derivative error must interact in some unspecified manner to produce worst case error for Taylor series coefficients.

*4.1.4 Using Minimal Normal Taylor Series Coefficient Input.* We measure normal Taylor series coefficients at the same times and image point offsets as earlier [Barron 1988]. In a similar manner as for normal image velocities, we measure horizontal/vertical (HV) and random normal Taylor series coefficients. A set of normal Taylor series coefficients consists of two normal coefficients for each of  $\vec{v}_n$ ,  $\partial\vec{v}_n/\partial y_1$ ,  $\partial\vec{v}_n/\partial y_2$ , and  $\partial\vec{v}_n/\partial t$  or  $\partial^2\vec{v}_n/\partial y_1\partial y_2$ . As before, there is a good correlation between predicted and observed worst case error amplification factors for minimal input data. Error amplification factors range from 1.7 to 552 for random normal Taylor series coefficient data and from 5 to 780 for horizontal/vertical normal Taylor series coefficient data. We also observed the presence of multiple solutions for motions 1, 5, 6, and 7 and dual solutions for some of Waxman and Ullman's planar motions.

#### 4.2 Using Least Squares Input

As we have seen above, calculation of motion and structure is not feasible for minimal input. In this section we report the results of our investigation when least squares input is used. The relevant question here is "How much least squares input is needed to obtain a feasible motion and structure calculation?" Many researchers, for example Waxman and Ullman [1983, 1985] have suggested that the use of least squares input can produce acceptable error amplification factors. A second related question is "With what kind of input and how much of it is needed?"

The main tool used in our analysis of least squares input is SVD analysis. We compute average and worst

case error amplification factors for various amounts of least squares input for full and normal image velocity and full and normal Taylor series coefficients. We construct least squares input by using multiple sets of minimal input data in our experiments. One advantage of using time-varying input is that it is easier to collect the volume of data needed for a feasible motion and structure calculation over a period on time rather than at one time only. Lastly, we emphasize that all the experimental results presented in section 4.1 for minimal input also hold for least squares input. For example, increasing the spatial or temporal extent for least squares image velocity data will result in lower error amplification factors for most motions.

*4.2.1 Using Least Squares Image Velocity Input.* In this and the next section we use a solution point  $\vec{V}_s = (0, 0)$ . We investigate least squares using 3, 6, and 24 sets of image velocity data. (Each set of image velocity consists of four image velocities.) The image points are distributed within the same spatiotemporal extents used for minimal input (see [Barron 1988] for details on the image point distribution). Thus, 24 sets of image velocity means we use 96 image velocities in total (we compute  $J_{192 \times 8}$ ). We believe this is an upper limit on the number of image velocities we can realistically hope to measure for a single surface patch over a small range of times.

While the use of least squares image velocity data is helpful in reducing worst case error amplification factors it still does not make motion and structure computations feasible in general, although motions 3, 4, and DW10 are now feasible. It would seem that the tradeoff of obtaining a worst case error amplification factor of 3 or less and the necessary increase in the amount of data and the spatial and temporal extents of the measurements required is unreasonable. Using more than 100 image velocities, a larger spatial extent than 30.9 degrees (already about 1/5 of the image) or a larger temporal extent (the underlying assumptions such as local planarity and the fixed-axis assumption may be violated) is not realistic for most motions. On the other hand, average SVD amplification results for 24 sets of image velocities are most encouraging; all motions except motion 5 are feasible (motion 5 is feasible when the temporal extent exceeds 0-0.5). In fact, except for motion 5 average case error amplification factors ranged from 0.1 to 1. If we examine the eight error amplification factors that result from the SVD of  $J$  we see that only one (or sometimes two) of the factors

are large, the others are often around 1 or less. It seems that worst case error directions occupy only a small part of the overall error direction space. This suggests that average error analysis rather than worst case error analysis may be a more appropriate type of analysis for least squares image velocity input. This is an important observation: all researchers, including [Waxman and Ullman 1983, 1985; Waxman and Wohn 1984, 1985; Subbarao 1986; Bandyopadhyay 1986] and others, used random noise in the input to their motion and structure algorithms. Least squares has another effect: local anomalies in error amplification factors that occur when minimal image velocity data are used to compute motion and structure around the image, that is, an error amplification factor much higher than its neighbors' error amplification factors, are smoothed out.

*4.2.2 Using Least Squares Normal Image Velocity Input.* As in the minimal input case we use two types of normal image velocity data: horizontal/vertical and random. One reason to use normal image velocities directly in the computation of motion and structure is because it is easier to measure normal rather than full image velocity data from most image sequences (due to the aperture problem [Horn and Shunk 1981; Marr and Ullman 1981]). Another of our goals in using normal image velocity data was to test the velocity functional method [Waxman and Wohn 1984, 1985]. The velocity functional method was introduced as a way of computing the "observables" from a normal image velocity field. These observables were then used in the subsequent motion and structure calculation. We carried the investigation one step further: we are interested in computing motion and structure directly from least-squares normal image velocity input. This combines the velocity functional method and the motion and structure computation from the "observables" into one computation. We found that even when 96 normal image velocities were used the motion and structure computation was only feasible for motions 3, 4, and DW10 (for worst case error in the input). Waxman and Wohn suggest the use of several hundred normal image velocities. This may be unreasonable, especially since the measurement is carried out at one time. We obtained worst case error amplifications of about 10–20 for their motions (with the exception of motion DW10 that exhibited an error amplification less than 2) when 96 normal image velocities were used. However, when average SVD analysis was performed the error amplification ranged from 0.1 to 2 or less for all motions, except motion 5 (where

the average error amplification ranged from just over 2 to just under 14). Waxman and Wohn report 5% output error given 10% random input error for typical runs. (Depending on the  $L_2$  of the input and output vectors, this is approximately equivalent to an error amplification factor ranging from 0.5 and 2.0).

*4.2.3 Using Least Squares Taylor Series Coefficient Input.* The use of worst case least squares Taylor series coefficient data yields feasible motion and structure computations for all the motions except motion 5. We use two-point formulations when computing motion and structure from actual Taylor series coefficients: the first is the same as for the least squares image velocity data and the second measures each set of Taylor series coefficients at a particular image point and time. The same spatiotemporal extents are used for both point formulations. Estimated Taylor series coefficients are computed using image velocities measured at the first point formulation. (See [Barron 1988] for complete details.) The error amplification factors for the two point formulations are very similar. All motions, with the exception of motion 5, are feasible using only 6 sets of Taylor series coefficients, a reasonable amount of data to measure, especially in time. Motion 5 becomes feasible if average case error is present in the input. It seems that actual Taylor series coefficients are a bit better than estimated Taylor series coefficients: the worst case error amplifications are slightly smaller and various actual Taylor series coefficient pairs can be measured at a large number of spatialtemporal extents or complete sets can be measured at individual image locations and times. Hence the use of actual Taylor series coefficients is more flexible than the use of estimated Taylor series coefficients. At the beginning of section 4.1.3 we described the relationship between error in a minimal image velocity set and its corresponding estimated Taylor series coefficients. Small image velocity error corresponds to much larger Taylor series coefficient error. Hence, it is not surprising that Taylor series coefficients are more robust than image velocities for motion and structure computations when there is noise in the input. The problem of image velocity computation has been addressed by many researchers (see section 1.2 for references), only now are there encouraging results [Anandan 1989; Fleet and Jepson 1990; Heeger 1988]. We suggest that a second approach might be to measure actual Taylor series coefficients directly from the raw time-varying data. The design of spatiotemporal filters to perform this task seems appropriate. There is some

progress toward this goal. Eagleson [1989] has designed filters that measure image translation, divergence, curl, and shear. The divergence, curl, and shear components can be shown to be equivalent to the first-order spatial image velocity derivatives [Longuet-Higgins and Prazdny 1980 and Barron 1984].

*4.2.4 Using Least Squares Normal Taylor Series Coefficient Input.* Again, because of the aperture problem it may be easier to measure normal Taylor series coefficients rather than full Taylor series coefficients. With the exception of motion 5, these are feasible for worst case input error, provided all 96 normal Taylor series coefficients are used. This includes both horizontal/vertical and random normal Taylor coefficient series data measured for both point formulations. Motion 5 is feasible if average Taylor series coefficient error is used. Since the spatiotemporal filters that would measure this type of data would do so locally, we believe that this type of data is the most practical for motion and structure calculations.

### 4.3 The Stability of Motion and Structure

We have observed that error amplification factors decrease for increasing spatiotemporal extents. In addition, as the amount of input data is increased not only is the error amplification reduced but the condition number also decreases. This means that least squares motion and structure computations become both more feasible and more robust than when the corresponding calculation is performed on a minimal set of input data. Thus, least squares not only produces small error amplification factors but also lower condition numbers. Our feasibility criterion is thus satisfied.

## 5 Conclusions

We reiterate the main contributions of this research:

(1) We have formulated a motion and structure algorithm that uses full/normal image velocity data and full/normal Taylor series coefficient data as its input. Measuring this input over time usually results in lower sensitivity to input error, allows us to analyze some motions that are singular at one time, and allows us to collect larger amounts of input data for a least squares calculation.

(2) We have conducted an extensive sensitivity analysis for both minimal and least squares input for the four types of input for best (in the minimal case), average, and worst case error. We found that in the least

squares case, both average/worst case normal image velocity data were about 20% worse than average/worst case full image velocity data, worst case normal Taylor series coefficient data were about 40% worse than worst case full Taylor series coefficient data, and average normal image velocity data are about the same as worst case normal Taylor series coefficient data. (Worst case normal image velocity data were 5 to 10 times worse than worst case normal Taylor series coefficient data.) We stated the conditions under which we consider a motion and structure computation to be feasible and conclude that:

- (a) Motion and structure can be feasibly recovered in the average case if full/normal least squares image velocity data are used.
- (b) Motion and structure can be feasibly recovered in the worst case if full/normal least squares Taylor series coefficient data are used.

Indeed, since most of the motions exhibit  $\rho$  values less than 1 for average or worst case input we can conclude that motion and structure may even be useful for autonomous vehicle navigation. Further, we recommend the following: use normal Taylor series coefficient data measured over time to compute motion and structure. The design of filters to measure this normal data locally from an image sequence is advocated as an area of future research.

## 6 Other Work

We are currently investigating the recovery of motion and structure from realistic images using image velocity data computed by the filtering techniques of Heeger [1988], Waxman et al. [1988], Anandan [1989], Fleet and Jepson [1990] and some others (on the Yosemite fly-through sequence, produced by Lynn Quam at SRI, on image sequences created by Fleet and Jepson and on image sequences obtained from an ALV [Dutta et al. 1989]). Also, spatiotemporal filters that measure image velocity and first-order spatial image velocity derivatives are currently being designed at the University of Western Ontario [Eagelson 1989]. Measurements from all these (and other) techniques can be used in testing our motion and structure algorithm. We have devised binocular motion and structure algorithms that use time-varying image velocity information from a binocular image sequence [Barron et al. 1987b] or use independently computed left and right monocular solutions [Barron et al. 1989]. We are currently performing a sensitivity analysis.

## Acknowledgments

We gratefully acknowledge financial support from the Natural Science and Engineering Council of Canada (NSERC) and the Information Technology Research Center (ITRC) of Ontario.

## References

- Adiv, G. 1984. Determining 3-D motion and structure from optical flow generated by several moving objects. COINS Tech. Rept. 84-07, University of Massachusetts, April. (Also *PAMI* 7 (4): 384-401, 1985.)
- Aloimonos, J., and Brown, C.M. 1984. Direct processing of curvilinear sensor motion from a sequence of perspective images. In *Proc. 2nd Workshop on Comput. Vision: Representation and Control*, Annapolis, pp. 72-77.
- Aloimonos, J.Y., and Rigoutsos, I. 1986. Determining the 3-D motion of a rigid planar patch without correspondence, under perspective projection. In *Proc. Workshop on Motion: Representation and Analysis*, Kiawah Island, S.C., May 7-9.
- Aloimonos, Y., Weiss, I., and Bandyopadhyay, A. 1987. Active vision. In *Proc 1st Intern. Conf. Comput. Vision*, London, pp. 35-54.
- Anandan, P. 1989. A computational framework and an algorithm for the measurement of visual motion. *Intern. J. Comput. Vision* 2 (3): 283-310.
- Bandyopadhyay, 1986. A computational study of rigid motion perception. Ph.D. Thesis, TR-211, Dept. of Computer Science, University of Rochester, NY, December.
- Bandyopadhyay, A., and Aloimonos, J. 1985. Perception of rigid motion from spatio-temporal derivatives of optical flow, TR-157, Dept. of Computer Science, University of Rochester, NY, March.
- Barron, J.L. 1984. A survey of approaches for determining optic flow, environmental layout and egomotion. RBCV-TR-84-5, Dept. of Computer Science, University of Toronto, November.
- Barron, J.L. 1988. Computing motion and structure from noisy time-varying image velocity information. Ph.D. thesis, Dept. of Computer Science, University of Toronto, June. (Also TBCV-TR-88-24, Dept. of Computer Science, University of Toronto, August.)
- Barron, J.L., Jepson, A.D., and Tsotsos, J.K. 1987a. The sensitivity of motion and structure computations. In *Proc. 6th Amer. Assoc. Artif. Intell. Conf.*, Seattle, July, pp. 700-705.
- Barron, J.L., Jepson, A.D., and Tsotsos, J.K. 1987b. Determining egomotion and environmental layout from noisy time-varying image velocity in binocular image sequences. In *Proc. 10th Intern. Joint Conf. Artif. Intell.*, Milan, pp. 822-825.
- Barron, J.L., Jepson, A.D., and Tsotsos, J.K. 1987c. Determination of egomotion and environmental layout from noisy time-varying image velocity in monocular image sequences. In *Proc. ICIAP*, Sicily, Italy, September, pp. 425-432.
- Barron, J.L., Jepson, A.D., and Tsotsos, J.K. 1988. The feasibility of motion and structure computations. In *Proc. 2nd Intern. Conf. Comput. Vision*, Tampa, FL, December, pp. 651-657.
- Barron, J.L., Jepson, A.D. and Tsotsos, J.K. 1989. Computing binocular motion and structure from monocular motion and structure. In *Proc. ICIAP*, Positano, Italy, September, pp. 399-406.
- Broida, T.J., and Chellappa, R. 1986. Kinematics of a rigid object from a sequence of noisy images: A batch approach. In *Proc. Conf. Comput. Vision Pat. Recog.*, Miami Beach, FL, pp. 176-182.
- Bruss, A.R., and Horn B.K.P. 1983. Passive Navigation. *Comput. Vision, Graphics, Image Process.* 21: 3-20.
- Buxton, B.F., Buxton, H., Murray, D.W., and Williams, N.S. 1984. 3-D solutions to the aperture problem. In T. O'Shea (ed.), *Advances in Artificial Intelligence*. Elsevier Science Publishers B.V.: North Holland, pp. 631-640.
- Dennis, J.E., and Schnabel, R.B. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall: Englewood Cliffs, NJ.
- Dreschler, L.S., and Hagel, H.-H. 1982. Volumetric model and 3-D trajectory of a moving car derived from monocular TV-frame sequences of a street scene. *Comput. Graphics Image Process.* 20: 199-228.
- Dutta, R., Manmatha, R., Williams, L.R., and Riseman, E.M. 1989. A data set for quantitative motion analysis. In *Proc. Conf. Comput. Vision Pat. Recog.*, San Diego, June, pp. 159-164.
- Eagleson, R. 1989. Measurement of motion-induced image deformations: Spatio-temporal filters for translation, divergence, curl and shear. In *Proc. of Vision Interface '89*, London, Ontario, June, pp. 61-69.
- Fang, J.-Q., and Huang, T.S. 1984. Solving three-dimensional small-rotation motion equations: Uniqueness, algorithms and numerical results. *Comput. Vision, Graphics, Image Process.* 26: 183-206.
- Fang, J.-Q., and Huang, T.S. 1984. Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames. *IEEE Trans. Patt. Anal. Mach. Intell. (PAMI)* 6 (5): 545-554.
- Fleet, D.J., and Jepson, A.D. 1990. Computation of component image velocity from local phase information. *Intern. J. Comput. Vision* 5 (1): 77-104. (See also RBCV-TR-89-27, Dept. of Computer Science, University of Toronto, March, 1989.)
- Gibson, J.J. 1957. Optical motions and transformations as stimuli for visual perception. *Psychological Review* 64 (5): 288-295.
- Hay, J.C. 1966. Optical motions and space perception: An extension of Gibson's analysis. *Psychological Review* 73 (6): 550-565.
- Heeger, D.J. 1988. Optical flow using spatiotemporal filters. *Intern. J. Comput. Vision* 1 (4): 279-302.
- Horn, B.K.P. 1986. *Robot Vision*, MIT Press: Cambridge, MA.
- Horn, B.K.P., and Schunck, B.G. 1981. Determining optical flow. *Artificial Intelligence* 17: 185-203.
- Horn, B.K.P., and Weldon, E.J. 1988. Direct methods for recovering motion. *Intern. J. Comput. Vision* 2 (1): 51-76.
- Kanatani, K. 1985. Structure from motion without correspondence: General principle. In *Proc 9th Intern. Joint Conf. Artif. Intell.*, Los Angeles, pp. 886-888.
- Lawton, D.T. 1983. Processing translational motion sequences. *Comput. Graphics Image Process.* 22: 116-144.
- Longuet-Higgins, H.C. 1981. A computer algorithm for reconstructing a scene from two projections. *Nature* 293, September, pp. 133-135.
- Longuet-Higgins, H.C., and Prazdny, K. 1980. The interpretation of a moving image. In *Proc. Roy. Soc. (London)* B208: 385-397.
- Marr, D., and Ullman, S. 1981. Directional selectivity and its use in early visual processing. *Proc. Roy. Soc. (London)* B211: 151-180.
- Mathies, L., Szeliski, R., and Kanade, T. 1989. Kalman filter-based algorithms for estimating depth from image sequences. *Intern. J. Comput. Vision* 3 (3): 209-238.

- Negahdaripour, S., and Horn, B.K.P. 1987. Direct passive navigation. *IEEE Trans. PAMI* 9 (1): 168-176.
- Prazdny, K. 1979. Motion and structure from optical flow. In *Proc. 6th Intern. Joint Conf. Artif. Intell.*, Tokyo, pp. 702-704.
- Regan, D. and Beverley, K.I. 1982. How do we avoid confounding the direction we are looking and the direction we are moving. *Science* 215: 194-196.
- Roach, J.W., and Aggarwal, J.K. 1980. Determining the movement of objects from a sequence of images. *IEEE Trans. PAMI* 2 (6): 554-562.
- Shariat, H., and Price, K.E. 1986a. How to use more than two frames to estimate motion. In *Proc. IEEE Workshop on Motion: Representation and Analysis*, Charleston, SC, May, pp. 119-124.
- Shariat, H., and Price, K.E., 1986b. Results of motion estimation with more than 2 frames. In *Proc. DARPA Image Understanding Workshop*, Miami, FL, pp. 694-703.
- Spetsakis, M.E., and Aloimonos, J. 1988. Optimal computing of structure from motion using point correspondences in two frames. CAR-TR-389 (CS-TR-210), Computer Vision Lab, Center for Automation Research, University of Maryland.
- Subbarao, M. 1986. Interpretation of visual motion: A computational study. Ph.D. thesis, CAR-TR-221, Center for Automation Research, University of Maryland, September.
- Subbarao, M., and Waxman, A.M. 1985. On the uniqueness of image flow solutions for planar surfaces in motion. CAR-TR-114 (CS-TR-1485), Center for Automation Research, University of Maryland. (Also, *3rd Workshop on Computer Vision: Representation and Control*, 1985, pp. 129-140.)
- Synder, M.A. 1986. The accuracy of 3-D parameters in correspondence-based techniques. Dept. of Computer and Information Science, University of Massachusetts, June.
- Thompson, W.B., and Kearney, J.K. 1986. Inexact vision. In *Proc. Workshop on Motion: Representation and Analysis*, May 7-9, Charleston, SC, pp. 15-21.
- Tsai, R.Y., and Huang, T.S. 1984. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *Trans. IEEE PAMI* 6 (1): 13-27.
- Tsai, R.Y., Huang, T.S., and Zhu, W.-L. 1982. Estimating three-dimensional motion parameters of a rigid planar patch II: Singular value decomposition. *IEEE Trans. Acoustics, Speech and Signal Process.* 30 (4): 525-534.
- Ullman, S. 1979. *The Interpretation of Visual Motion*. MIT Press: Cambridge, MA.
- Verri, A., and Poggio, T. 1987. Against quantitative optical flow. In *Proc. 1st Intern. Conf. Comput. Vision*, London, pp. 171-180.
- Waxman, A.M., and Ullman, S. 1983. Surface structure and 3-D motion from image flow: A kinematic analysis, CAR-TR-24, Center for Automation Research, University of Maryland, October.
- Waxman, A.M., and Ullman, S., 1985. Surface structure and three-dimensional motion from image flow kinematics. *Intern. J. Robotics Res.* 4 (3): 72-94.
- Waxman, A.M., and Wahn, K. 1984. Contour evolution, neighbourhood deformation and global image flow: Planar surfaces in motion. CAR-TR-58, Center for Automation Research, University of Maryland, April.
- Waxman, A.M., and Wahn, K. 1985. Contour evolution, neighbourhood deformation and global image flow: Planar surfaces in motion. *Intern. J. Robotics Res.* 4 (3): 95-108.
- Waxman, A.M., Kamgar-Parsi, B., and Subbarao, M. 1987. Closed-form solutions to image flow equations. In *Proc. 1st Intern. Conf. Comput. Vision*, London, pp. 12-24.
- Waxman, A.M., Wu, J., and Bergholm, F. 1988. Convected activation profiles and the measurement of visual motion. In *Proc. Conf. Comput. Vision Patt. Recog.*, Ann Arbor.
- Webb, J.A., and Aggarwal, J.K. 1981. Visually interpreting the motion of objects in space, *IEEE Computer* 14: 40-46.
- Wu, J.J., Rink, R.E., Caelli, T.M., and Gourishankar, V.G. 1989. Recovery of the 3-D location and motion of a rigid object through camera image (An Extended Kalman Filter Approach). *Intern. J. Comput. Vision* 2 (4): 373-394.

## Notes

<sup>1</sup>Of course, we must still be able to solve surface-to-surface correspondence, i.e., we must be able to group together all image velocities distributed locally in space and time that belong to the same planar surface. See Adiv [1984] for one approach to this problem.

<sup>2</sup>Given  $f = 1$  and that the minimum/maximum horizontal/vertical coordinates of the image are  $\pm 1/2f$ , if we assume a  $256 \times 256$  pixel image coordinate system,  $\vec{V}_s = (20, 20)$  in pixels corresponds to  $\vec{V}_s = (0, 0.78125, 0.78125, 1)$  in  $f$  units. We use pixel image locations here because these can be expressed as simple integers and are more intuitive.

<sup>3</sup>Note that all the flow figures are scaled by some value. For example, the image velocities in Figures 7-10 are unscaled (a scale factor of 1) while the image velocity derivatives of Figure 11 are actually 4 times longer than shown (the scale factor is 0.25). Different scale factors mean comparison of the vector magnitudes must be adjusted accordingly.

<sup>4</sup>If pure translation is assumed we only need  $2\frac{1}{2}$  image velocities. In this case we let the LU decomposition of  $J$  choose the  $2\frac{1}{2}$  image velocities to be used in the motion and structure computation.