ISSES IN EXTRACTING MOTION PARAMETERS AND DEPTH FROM APPROXIMATE TRANSLATIONAL MOTION

R. Dutta, R. Manmatha
E. Riseman, M.A. Snyder

COINS Technical Report 88-52

May 1988

Issues in Extracting Motion Parameters and Depth from Approximate Translational Motion

R. Dutta R. Manmatha Edward M. Riseman M. A. Snyder
Computer and Information Sciences Department
University of Massachusetts at Amherst *

*This research was supported in part by the DARPA and US Army ETL under contract number DACA76-85-C-0008, and DARPA and RADC under contract number F30602-87-C-0140. Part of the research is also supported by the National Science Foundation under grant number DCR-8500332.

Abstract

In dynamic imaging situations where the sensor is undergoing primarily translational motion with a relatively small rotational component, it might seem likely that *approximate* translational motion algorithms can be effective in determining depth. By restricting the processing to the two dimensions of translational motion there is a great reduction in complexity from the five dimensions of general motion (excluding the scaling component of sensor velocity or displacement). In an attempt to recover depth of points over a sequence of frames in such situations a set of three algorithms was applied sequentially: elimination of the effect of small rotations, then extraction of the FOE, and then determination of depth of points. The goal was the extraction of obstacles at a distance beyond the accuracy of the range sensors in the ALV research program.

In this paper we show, however, that even small rotations can significantly affect the performance of algorithms designed for pure translational motion. In addition, a theoretical analysis of the problems in computing the FOE is presented. The attempt to correct for small rotations in the motion system described above was ineffective. Accurate extraction of the FOE and recovery of depth without dealing with general motion was not achieved.

More recently, a pair of previously developed algorithms have been combined to yield a *general motion algorithm* and have been applied to sequences of approximate translational motion. By determining the small rotational components of motion, more promising depth results have been extracted under the same circumstances described above. The conclusion is that in order to determine depth of points in many real situations, general motion analysis must be applied even when sensor motion is primarily translational. Various alternatives for extracting depth from motion are briefly considered.

1 Introduction

1.1 Background - Motion Algorithms Developed at the University of Massachusetts

At the University of Massachusetts several algorithms have been developed and applied to real scenes in an attempt to develop practical and robust techniques for mobile vehicle applications. The goals have been the recovery of sensor motion parameters when they are unknown, the determination of vehicle location via landmarks in known environments, and the recovery of the depth of environmental points, both for the derivation of mechanisms for obstacle avoidance, and for object recognition.

In cases of pure translational motion of the sensor, Lawton [1,2] developed reasonably robust techniques for recovery of the sensor motion parameters (or equivalently the focus of expansion – FOE); these techniques for FOE search were extended by Pavlin [3,4]. On several synthetic image sequences, the FOE was recovered within 1-2 degrees (about 5–10 pixels) for axes inside a 45 degree cone around the line of sight and within 5-10 degrees outside that range [3]. It has also been successfully applied to several natural scenes.

In another investigation the goal was to recover depth from *known* motion. Since the exact motion parameters could never be known, some degree of imprecision should be included. Snyder [5] has provided an analysis of the effects of uncertainty in the location of the FOE and the tracked feature points on the determination of depth, and on the location and size of the search window in future frames, as a function of the uncertainties in the FOE, the feature points, and the corresponding environmental depths. Bharwani [6] applied these ideas using the time-adjacency relationship in an iterative spatio-temporal strategy

of prediction-refinement of feature point displacements and their associated depths. The algorithm was applied to a sequence of hand-registered real-world images, with either an FOE that was hand-supplied or extracted from the Lawton-Pavlin algorithm. In several cases this algorithm achieved useful depth values over a sequence of images [6,7]. Thus, the methods of FOE determination and depth extraction, used together, showed strong promise for practical vehicle navigation in outdoor environments.

In addition to the above a two–stage algorithm for analyzing general motion has also been implemented. The first stage is based on Anandan's hierarchical extraction of displacement vectors [8,9,10]. The second stage is based on Adiv's algorithm [11,12] for analyzing general motion in the presence of multiple independently moving objects.

1.2 The Problem - Obstacle Avoidance via Motion of an Approximately Translating Mobile Vehicle

The algorithms evaluated in this paper are part of the research effort directed towards the development of a system for obstacle avoidance via motion analysis as part of the Autonomous Land Vehicle (ALV) project. The primary goal of the ALV project is to design a vehicle which can autonomously navigate through a wide range of environmental situations. For obstacle avoidance the depth of objects out to about 80 feet needs to be extracted to determine whether they may be obstacles. The phase–difference based ERIM scanning laser range finder [13] seems to be limited to distances of less than 40 feet Thus our specific research goal was to extract depth at distances greater than the effective range of the ERIM sensor, or approximately in the 40-80 foot range.

In an effort to provide more powerful constraints that would enhance the effectiveness of such a practical application, we decided to consider the case of depth from *known*

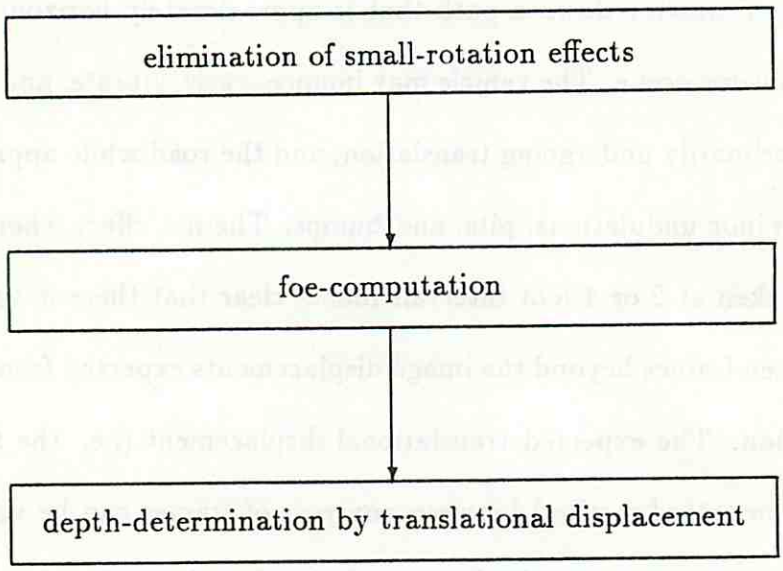
translational motion. Since precisely known sensor motion is not possible without sophisticated instrumentation, the problem was modified to one of *approximate* translational motion.

However, in actual application of the methods to the Carnegie–Mellon University NAVLAB [13] moving down a path that is approximately horizontal and planar, certain general problems occur. The vehicle may bounce, sway, vibrate, and of course turn slightly while still primarily undergoing translation, and the road while approximately planar may vary with minor undulations, pits, and bumps. The net effect when examining sequences of images taken at 2 or 4 foot intervals makes clear that there is significant "misregistration" between frames beyond the image displacements expected from smooth translational sensor motion. The expected translational displacement (i.e. the translation due to the *average* or *smoothed* motion) between any pair of frames can be significantly affected by perturbations which involve both a translational and rotational component. This could result in unpredictable movement of the "FOE"[1] over the sequence of frames (i.e. there might be a varying direction of translational motion which if "averaged" across frames would be the "approximate" translational motion), as well as a varying rotational component that causes additional rotational displacements of image points independently of the depths of the associated environmental points. Thus, even though the vehicle is translating in some fixed direction that remains approximately constant across many frames, there can be significant local variations between nearby pairs of frames in the sequence. In order to obtain accurate depths from translational motion, it may be necessary to update the FOE location and remove the effect of rotational displacements.

[1] If there is rotation, there is of course no FOE; nevertheless, the translation vector will intersect the image plane somewhere and sometimes we will use the term FOE in this manner.

1.3 An Attempt at Building a Translational Motion Subsystem

Figure 1: Processing of Approximate Translational Motion for Depth Extraction



As a consequence of the effects described in the previous section, the two translational motion algorithms for FOE recovery [1,3] and extraction of depth [6] mentioned earlier cannot be applied directly. Either some form of preprocessing is necessary to allow translational processing to be maintained despite the problems described, or else the complexities of general motion must be considered. An attempt was made to carry out "registration" of the images [14] —i.e., removal of small rotations by translation of the image — as a simple preprocessing step to allow effective translational motion analysis. The three algorithms shown in Figure 1 were to be applied sequentially to determine the depth of environmental points. The first algorithm was intended to eliminate a significant portion of small rotations by image registration [14]. The second algorithm was to compute the position of the

FOE for the registered images [2,4]. The third algorithm was to use the time-adjacency relationship and an iterative refinement procedure to compute depth [6].

1.4 Analysis of the Problems

In view of all the difficulties with the estimation of motion parameters, the difficulty of autonomous navigation through unknown terrain is not surprising. Our restriction of the problem to finding the depth of points from approximately known translational motion was intended to introduce sufficient constraints to overcome these problems. It is the purpose of this paper to draw attention to the practical difficulty of developing such a system. While each of the algorithms in isolation had some experimental success in achieving its intended goal, they failed as an integrated set of algorithms. A global system is much more difficult to construct because of cumulative errors generated by the subsystems. In this paper we show that even small amounts of rotation can cause large errors in determining the FOE and depth using translational motion algorithms. Simple rotation-elimination mechanisms, such as the one used here, often place too many restrictions on the imaging situations for practical use.

The effect of errors on the estimation of depth and motion parameters has been studied by many authors [5,15,16,17,18]. Tsai and Huang [18] demonstrated experimentally that "unless the error in finding point correspondences is less than 3%" the solutions to the motion parameters are "overwhelmed by noise." They also assert on the basis of experimental evidence that "error of the estimated motion and geometrical parameters can be reduced by using more point correspondences only if the error for the latter is less than 3%." Fang and Huang [17] showed that as the distance between the object and the image plane increases it becomes more difficult to solve the motion equations proposed by them.

It must be noted that these studies are by no means general and essentially apply to the algorithms proposed or studied by the researcher. However, they provide us with insight regarding the difficulty of motion parameter and depth estimation.

As a consequence of the difficulty of recovering the depth points in practical situations of approximate translational motion, we return later in this paper to an experimental investigation of the case of general motion. We consider a two stage algorithm for analyzing general motion: Anandan's hierarchical extraction of displacement vectors [8,9,10] followed by Adiv's algorithm [11,12] for analysis of general motion in the presence of independently moving objects. Initial results show the combination of these two algorithms to provide reasonably good estimates for the depth of environmental points.

2 Depth From Approximate Translational Motion

Recovery of depth from approximately known translational motion has been studied by Bharwani, et al [6]. This scheme is based on the time-adjacency relation and involves a prediction-based iterative refinement scheme for computing increasingly more accurate depth maps. Since it is based on the time–adjacency relation (see equation (1)), the position of the FOE must be known before the algorithm can be applied. In the next section, we give a brief discussion of the various components of the approach as developed at the University of Massachusetts.

2.1 FOE determination

The basic algorithm for determining the FOE was developed by Lawton [2]. The position of the FOE is determined in essentially two phases. In the first phase distinctive features

are extracted under the assumption that they can be more easily tracked across frames. In the second phase the direction of translation is found in the following manner. Initially a position of the FOE is hypothesized in the image plane. Selection of an FOE constrains each feature point to appear in the second frame on the radial line emanating from the hypothesized FOE and passing through the corresponding feature point in the first frame. The point on the path that correlates best (up to some resolution of correlation matching with the feature point in the first frame) will be assumed to be the feature correspondence, and the deviation from perfect correlation is regarded as an error. The sum of these errors for the set of feature points gives the total error for the hypothesized FOE. The objective is to find the FOE position for which the total error is minimized. A search algorithm involving a coarse sampling of the FOE followed by local hill climbing was demonstrated to be effective on several natural scenes.

In Pavlin, et al.'s modification of Lawton's algorithm, the orientation of the axis of translation and associated error values are represented in polar coordinates on the unit sphere centered at the camera focus. This gives a more uniform sampling of the hypothesized FOE positions than working with the image plane. The error surface is constructed by fitting a smooth surface over the values produced by a coarse sampling over polar angles. The sampling is then repeated locally around the minimum of the coarsely sampled error surface. This hierarchical search along with smoothing was intended to make it robust and more efficient. The smoothing was intended to eliminate fluctuations in the error surface and reduce the need for finer sampling resolution, thereby reducing the required computation.

Experimental results have shown that this algorithm is effective on real image data if

1. the motion is purely translational, and

2. approximately 8 to 16 distinctive feature points can be found which can be reasonably tracked across frames.

However, for reliable results when no a priori information is available to place constraints on the processing (e.g. approximate location of the FOE) the algorithm is quite time consuming. In addition various design parameters must be adjusted to account for the worst eventualities (e.g., the initial FOE sampling cannot be made too coarse because it might result in the selection of an incorrect global minimum for the FOE). Nevertheless, this algorithm appears to be quite useful for images in many practical situations.

2.2 Depth Determination

The discussion here describes the algorithm proposed and implemented by Bharwani, et al [6]. Consider the *time-adjacency* [19] relation:

$$\frac{D(t)}{d(t)} = \frac{Z(t)}{W(t)}, \tag{1}$$

where

d(t) : the radial velocity(displacement) in pixels of a point P in the image

D(t) : the distance of the point P from the FOE in pixels

W(t) : the Z-component of the actual velocity(displacement) of the corresponding scene point.

Z(t) : the depth of the corresponding scene point from the camera. Hence, depth Z can be found via pure translational motion if we know W, the FOE position, and the

matches across frames.

Essentially the technique can be described as iteratively improving the precision of depth estimates over a *sequence* of frames by using the depth estimates (along with their associated uncertainties) from the previous time steps. In future frames, the match region can be predicted better, if the following are known:

(a) the current estimate of the depth of a point

(b) the uncertainty associated with the depth.

Therefore, the method matches points between frames at a particular match resolution, finds depth and uses this to predict a smaller match window to be searched with a finer match resolution. Increasing the match resolution was intended to give a more accurate depth. In addition, the method can be guaranteed to have a constant upper bound on computation for processing between frames by controlling the correlation resolution (via interpolation) of the feature point matchings as an inverse function of the size of the predicted search window. This makes it a simple scheme for depth determination with a sequence of images for translational egomotion.

Even when the motion of the camera is known, ambiguities in the matching process between frames leads to erroneous depth determination. The matching process might fail because of occlusion, highlights, shadows, distortion of the surface over time, and similarity of image features etc. In addition to these problems, an insufficient search area for the match along the displacement path can result in no matches or incorrect matches being found. The matching problem is a very widely studied area in motion analysis [9,20,21] and will not be discussed here.

The major problems with the depth determination algorithm are:

1. The correspondence problem (i.e. effectiveness of the match process);

2. The accuracy of FOE determination;

3. The effect of small rotations;

A discussion of the problems of FOE determination and effect of small rotations is given in the next two sections and is a key focus of this paper.

3 Problems of FOE determination

3.1 The Image Model

We present here the equations describing displacement fields which will be used throughout the paper. Let

- (X, Y, Z) be a Cartesian coordinate system with its origin at the nodal point of the camera,
- (x, y) represent the corresponding coordinate system of a planar image,
- f be the focal length of the camera,
- F_i be the planar image of the environment at time t_i,
- F_{i+1} be the planar image of the environment at time t_{i+1}.

An environmental point $P(X, Y, Z)$ then appears at the point $p(x, y)$ in the image F_i. If x and y are given in pixels, then

$$x = f\frac{X}{Z},$$

$$y = f\frac{Y}{Z},$$

where f is the focal length of the camera in pixels, given by

$$f = \frac{N}{2}\cot\frac{FOV}{2} \tag{2}$$

Here FOV is the field of view of the camera, and the image resolution is $N \times N$.

Let the motion relative to the camera of a point in the rigid environment have translational component $\vec{T} = (U, V, W)$ and rotational component $\vec{\Omega} = (A, B, C)$. It can be shown that if

1. W/Z is much smaller than 1 (i.e., the translation of the point in the Z direction is much smaller than the depth of the point);
2. the rotation parameters are small; and
3. the field of view is of the camera is not very large;

then the displacement vector (u, v) of the image point between frames F_i and F_{i+1} is given by

$$u = \frac{fU - xW}{Z} + (\frac{-Axy + Bx^2}{f}) + Bf - Cy, \tag{3}$$

$$v = \frac{fV - yW}{Z} + (\frac{-Ay^2 + Bxy}{f}) - Af + Cx. \tag{4}$$

We note that u and v both have translational and rotational components:

$$(u, v) = (u_t, v_t) + (u_r, v_r) \tag{5}$$

where

$$u_t = \frac{fU - xW}{Z} \tag{6}$$

$$v_t = \frac{fV - yW}{Z} \tag{7}$$

$$u_r = (\frac{-Axy + Bx^2}{f}) + Bf - Cy \tag{8}$$

$$v_r = (\frac{-Ay^2 + Bxy}{f}) - Af + Cx \tag{9}$$

3.2 Overview

The *motion field* is the projection of the 3-D velocity/displacement field onto the image plane. Given the motion field, the motion parameters (U, V, W, A, B, C) and the scene structure (Z) can be uniquely determined up to a scale factor - except for a two-way ambiguity in certain special cases [22].

The *optical flow* is the image velocity or displacement recovered from the image. If the optical flow is precisely equal to the motion field, the motion parameters and scene structure recovered are the correct ones. In general, the optical flow is not equal to the motion field. This may be due to noise, measurement errors (e.g. usually our displacements are only accurate to within 0.5 pixel), correspondence error, etc. Even small errors in the optical flow can produce motion parameters which are far from the correct ones [11].

In the following subsections we first establish the error in the FOE position as a function of the error in the image displacements. By interpreting small rotations as deviations/errors from the displacements derived from pure translation, we show that even small rotations may cause large errors in the FOE position returned by pure translational methods. As a corollary we show how the different feature points should be weighted to improve the FOE computation.

3.3 The Effect of Displacement Errors on FOE Position

Let us assume that the motion is purely translational. Then from equation (5) it follows that

$$u = u_t = \frac{fU - xW}{Z} = f\frac{U}{Z} - x\frac{W}{Z}, \tag{10}$$

$$v = v_t = \frac{fV - yW}{Z} = f\frac{V}{Z} - y\frac{W}{Z}. \tag{11}$$

It is well known that the flow field given by (10) and (11) is purely radial, with center at the FOE. The position of the FOE (when it exists) is the point where $u = v = 0$. From (10,11) we easily see that this is the point

$$(x_0, y_0) = (f\frac{U}{W}, f\frac{V}{W}). \tag{12}$$

Suppose now that an error is made in determining the displacement (u, v), so that we find (u', v'), where

$$(u', v') = (u, v) + (a, b). \tag{13}$$

Thus, substituting (10) and (11) into (13), we find:

$$u' = f\frac{U}{Z} - x\frac{W}{Z} + a, \tag{14}$$

$$v' = f\frac{V}{Z} - y\frac{W}{Z} + b. \tag{15}$$

For purposes of illustration, let us assume that the scene being viewed is coplanar at depth Z with the image plane, and that (a, b) is the same for each image point. We can combine the coordinate-independent terms in (14) and (15) to find:

$$u' = f\frac{(U + \delta U)}{Z} - x\frac{W}{Z},$$

$$v' = f\frac{(V+\delta V)}{Z} - y\frac{W}{Z},$$

where the *constants* δU and δV are given by

$$\delta U = aZ/f, \quad \delta V = bZ/f. \tag{16}$$

The displacement field is therefore of the same form as (10,11), and hence there is still an FOE in this case, located (according to (12)) at

$$(x_0', y_0') = (f\frac{U+\delta U}{W}, f\frac{V+\delta V}{W}) = (x_0, y_0) + (\delta x_0, \delta y_0), \tag{17}$$

where

$$(\delta x_0, \delta y_0) = \frac{Z}{W}(a, b). \tag{18}$$

We will call this the "error in the FOE" due to the error (a, b) in the image displacement of the point located at depth Z.

Suppose now that we allow the displacement errors (a, b) in (13) to vary with position in the image plane, and that the environment is no longer coplanar with the image plane. It follows that the "errorful" displacement will no longer be radial, and hence there will not, in general, be an FOE. However, we could still apply our purely translational algorithm to this flow field. How will the "errorful" flow field affect our algorithm?

This is clearly not a question that can be given a general answer. In fact, the only result we can quote is equation (18), which we have proved only in the case of constant depth and constant error (a, b). We can, however, conclude one general rule from (18): for fixed, constant displacement error, distant points (those for which $Z \gg W$) will contribute a larger error to the position of the "FOE" than will nearby points (those for which Z and W are comparable). This is to be expected on intuitive grounds as well, since (generally

speaking) distant points will have smaller displacements than nearby points, and so a given displacement error will have a proportionally larger effect on the displacement of distant points than on that of nearby points.

This situation is illustrated in Figure 2. In Figure 2(a) the shift in the FOE is smaller than in Figure 2(b) because the displacement vectors d_1 and d_2 in Figure 2(a) are larger than the corresponding displacement vectors in Figure 2(b). The displacement vectors are larger when the points are closer. Hence, the larger percentage error in displacement for distant points returns a FOE position with greater error.

We note that no assumption about the source of the displacement error has been made–it could be noise in the input image, correspondence error, or anything else. The usual assumption is that the displacement errors are due to random, uncorrelated noise. This would lead to displacement errors which are random in both magnitude and direction, and so on average would not lead to a large error in the position of the FOE. However, if there is a systematic error, for instance when the displacement error is a constant, then the FOE error would be expected to be given by an expression something like equation (18). In the next section, we show that under certain not terribly restrictive conditions rotations can give rise to just such a systematic error.

3.4 Rotations as a Source of Systematic Error

When the scene undergoes a general rigid motion, the displacement field will be given by (5), with translational and rotational components by equations (6)–(9). We can, a la equation (13), consider the rotational components (u_r, v_r) as an error term on the underlying pure–translational displacement field (u_t, v_t). From (8) and (9), we see that

Figure 2: Shift in the FOE relative to the size of the displacement vectors

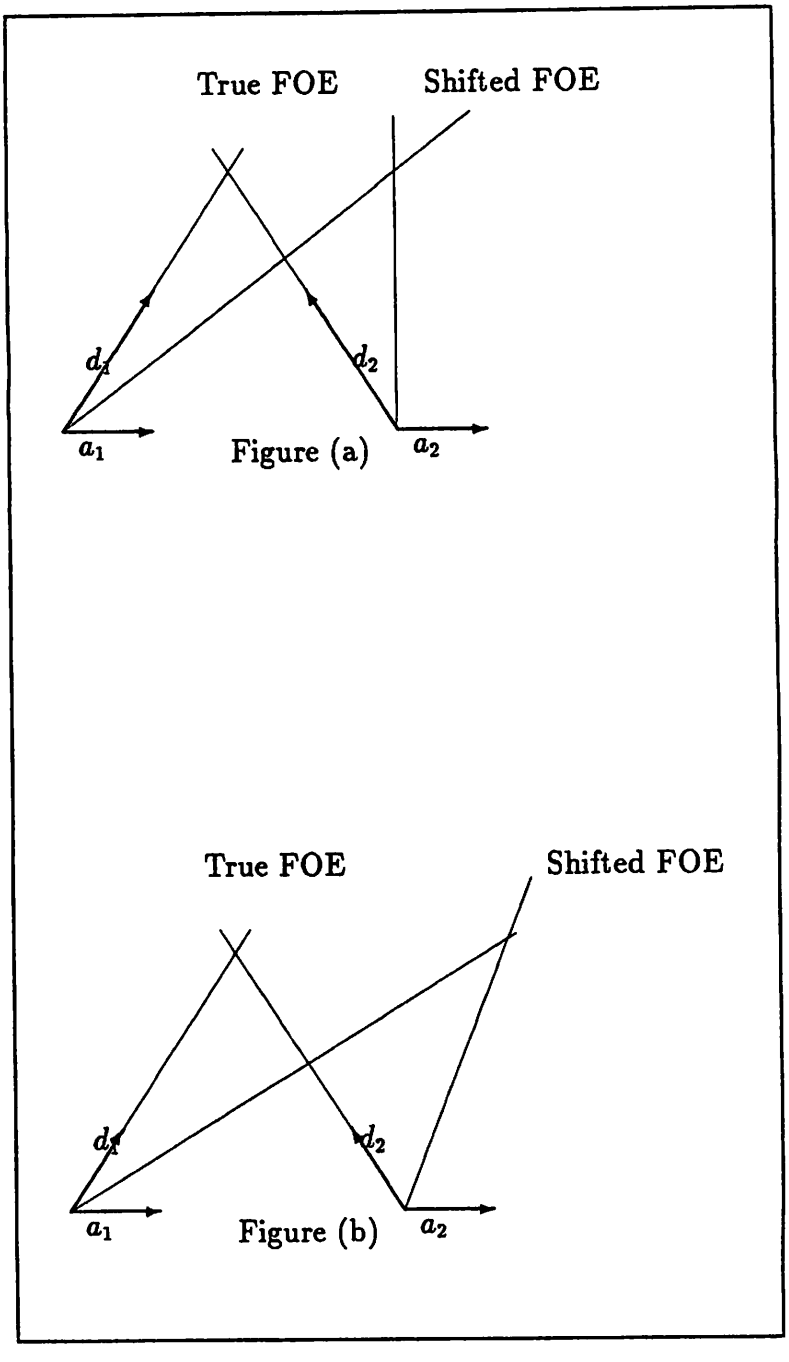
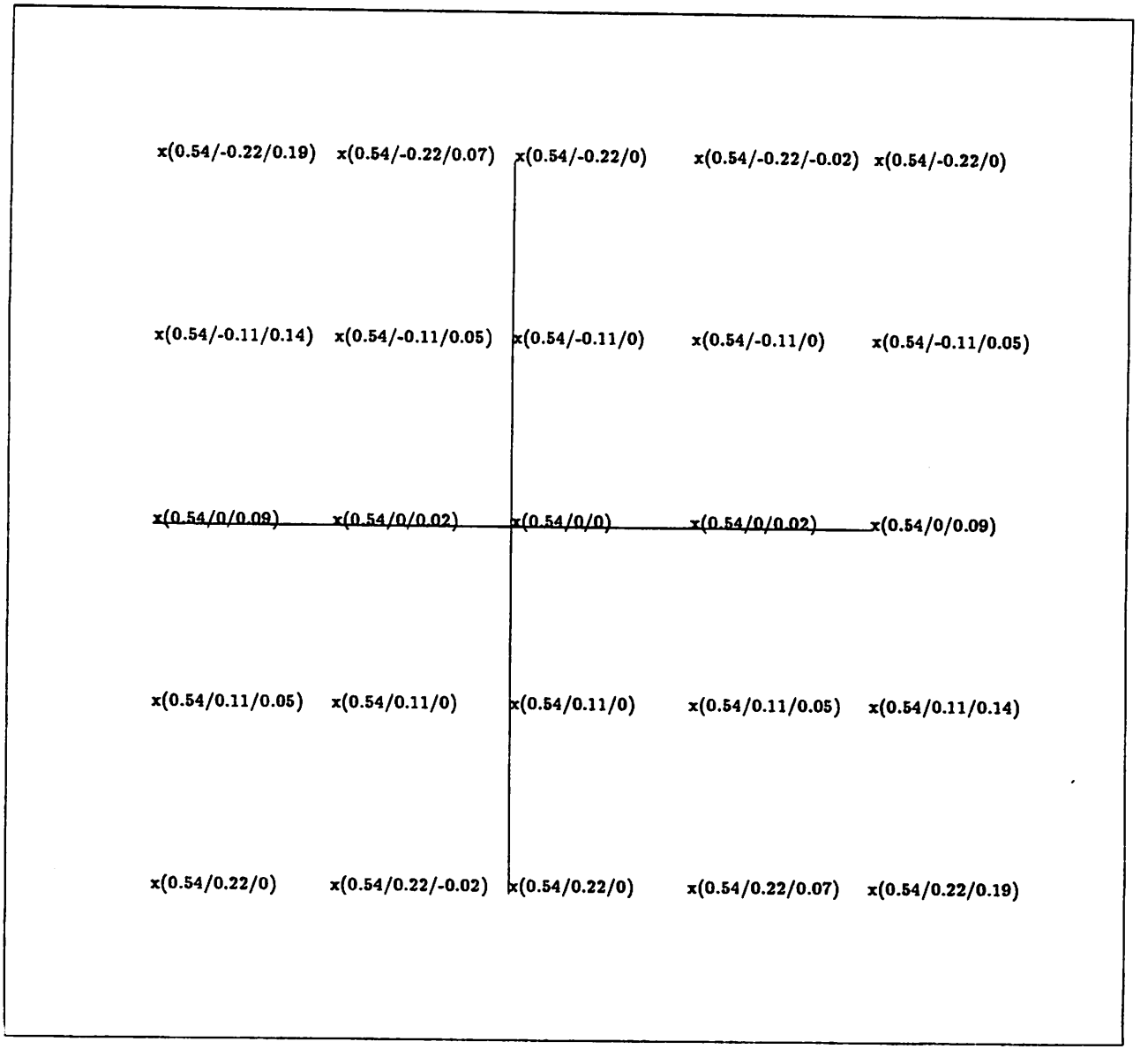


Table 1: **Rotational contributions to optical flow as a function of position.** Shows the (constant,linear,quadratic) contributions (in pixels) to the x–component of the displacement field at various spatial locations



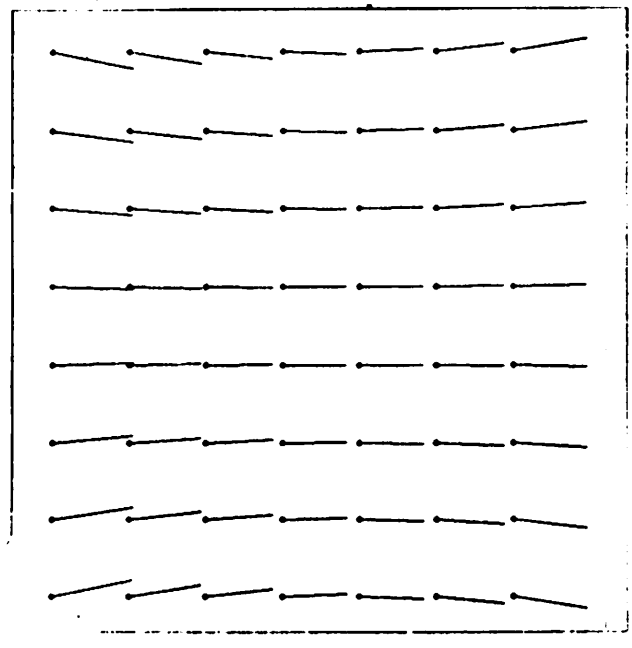
this "error" term has the following structure:

$$u_r = \{+Bf\} + \{-Cy\} + \{\frac{-Ay + Bx}{f}x\}, \tag{19}$$

$$v_r = \{-Af\} + \{+Cx\} + \{\frac{-Ay + Bx}{f}y\}. \tag{20}$$

That is, each component has a constant term, a term which varies linearly with image

Figure 3: Purely rotational flow fields.



coordinates, and a term which is quadratic in image coordinates. This is illustrated in Figure 3, where we see how the flow deviates more and more from a constant as we approach the periphery of the image, where the linear and quadratic terms become important.

Let us see how the constant, linear, and quadratic contributions compare to one another. We recall from equation (2) that f is the camera focal length in pixels. For instance, for a 256×256 image with 45° field of view, f is about 309 pixels. The coordinates x and y, on the other hand, are limited to being less than or equal to $N/2$ in magnitude,

which in this case is 128. Thus,

$$|\frac{x}{f}|, |\frac{y}{f}| \leq \tan \mathrm{FOV}/2 \cong \frac{128}{309} \cong 0.4. \tag{21}$$

For rotational angles which are comparable ($A \cong B \cong C$), we see from (19,20) that the order of magnitude of the linear and quadratic terms, with respect to the constant term, is given roughly as

$$\frac{\text{linear}}{\text{constant}} \leq 0.4,$$

$$\frac{\text{quadratic}}{\text{constant}} \leq 2 \times (0.4)^2 \cong 0.32.$$

Indeed, these upper limits are reached only at particular regions near the image boundaries. For the greater part of the image surrounding the center, the linear and quadratic terms are *much* smaller than the constant term. This is illustrated in Table 1, where we show the values taken by the constant, linear, and quadratic terms contributions to u at various points in a 256×256 image, with a field of view of 45°, for a rotation of $A = B = C = 0.1°$.

This leads us to make the assumption that the dominant effect of rotations is to add a constant term to the displacement field. We see that this will be a good assumption when:

1. The field of view of the camera is small (say $\leq 45°$).

2. The Z–component of the rotation is comparable to, or smaller than, the X– and Y–components of the rotation.

In particular, the assumption of constant displacement error is *not* valid when the primary rotation is around the line of sight (roll) (i.e., when C is much larger than A or

B), since the linear term will then be larger than the constant term over a large portion of the image.

3.5 The Effect of Rotation on FOE–error

In this section we combine the results of the previous two sections to find how rotations affect the position of the FOE.

In Section 3.3 we showed that for an environment coplanar with the image plane located at depth Z a constant displacement error leads to a displacement field which is still purely translational, so that an FOE exists. The error in this FOE due to the displacement error is given by equation (18). In Section 3.4 we argued that under certain not–terribly–restrictive conditions, the dominant effect of rotations was to add a constant error to the displacements, given by the constant term in equations (19,20):

$$(a, b) \equiv (Bf, -Af). \tag{22}$$

By combining (18) and (22), we find that the X– and Y–rotations give an FOE error (in pixels) of

$$(\delta x_0, \delta y_0) = \frac{Zf}{W}(B, -A). \tag{23}$$

To see how large an effect this is, let the FOV for a 256×256 image be 45°, and let $B = A$. In Table 2, we show $|\delta x_0|$ for various values of Z/W and of B. The first column gives the rotational angle, the second the displacement error (in pixels) for each angle, and the remaining columns give the FOE error (in pixels) for each angle.

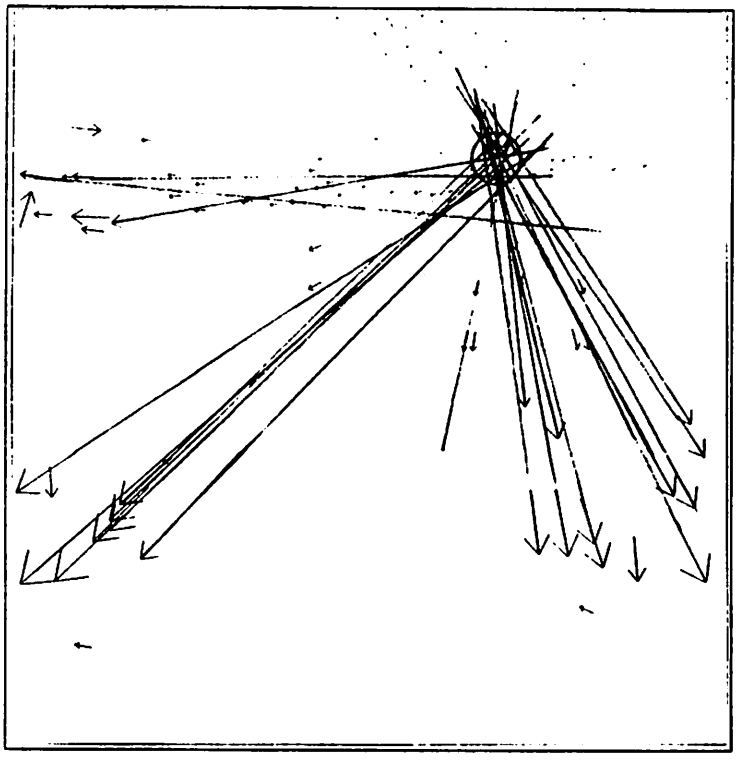
Of course, it is exceedingly rare that the environment and the image are coplanar, so these results cannot be strictly applied. However, we can draw a few general conclusions

Table 2: **Error in FOE (in pixels) with rotational angle and Z/W**

rotational error, B	corresponding disp., Bf	Z/W					
		5	10	20	30	50	100
0.1deg.	0.5 pixels	2.5	5	10	15	25	50
0.2deg.	1.0 pixels	5	10	20	30	50	100
0.3deg.	1.5 pixels	7.5	15	30	45	75	150
0.4deg.	2.0 pixels	10	20	40	60	100	200

from this analysis. The first conclusion is that when an image sequence contains even a

Figure 4: Estimated FOE obtained by extending the larger flow vectors.



small amount of rotation, the error in the FOE can be large. This error will be larger when the environmental points used to find the FOE are more distant. It would therefore be advantageous to weight nearby environmental points more than distant ones. We note that the fundamental reason for this is that nearby points will in general have larger interframe displacements than distant points. Thus, to minimize the FOE error we should give points with large displacements greater weight than those with small displacements (eg., Figure

4 shows the larger flow vectors obtained from frame pair 9-11 of Figure 5 extended to give an approximate FOE).

The second conclusion is that since most schemes round off displacements to the nearest half-pixel, the first row in Table 2 gives the worst-case error in finding the FOE for such schemes. In general, such rounding-off will give randomly-distributed displacement errors, which should not have much effect on the FOE. The worst-case error assumes that the displacement errors are not random, and hence do not cancel out.

4 Removing Rotation without Analyzing General Motion

We have shown in the previous sections that if an algorithm which assumes purely translational motion is applied to an image sequence which has even a small amount of interframe camera rotation, the "FOE" found by the algorithm can be far from the FOE that would be found if no rotation were present. Thus even small rotations can seriously degrade the performance of a purely translational algorithm.

One possible solution to this difficulty would be to somehow remove the (small) rotational component of the camera motion, producing a so-called *registered* image sequence in which the inter-frame camera motion is purely translational. This possibility has recently been investigated by Pavlin and Snyder [14]. In the next sections we discuss this registration scheme, its limitations, and some experimental results from using it.

4.1 The Registration Algorithm

Consider a dynamic image sequence $\{F_1, \ldots, F_n\}$ in which the camera is allowed to have general inter–frame motion. If we wish to run a purely translational motion algorithm on this sequence we will, as just discussed, obtain unreliable results in general because of the existence of small rotational components to the inter–frame motion. Suppose, however, that we could somehow find the rotational components of the inter–frame motion. We could then subtract out this motion from the second of each image pair, obtaining thereby a *new* image sequence in which the inter–frame camera motion is purely translational. This new image sequence $\{F_1^{\text{reg}}, \ldots, F_n^{\text{reg}}\}$ will be called a *registered* image sequence. The translational motion algorithm could then be safely applied to this registered sequence.

Unfortunately, for reasons given in [14], the construction of the registered image sequence is in principle impossible, but an approximation to a registered image sequence can be constructed as follows (see [14] for more details, justifications, etc.). The key idea behind the registration algorithm is that (infinitely) distant points move, from frame to frame, only because of rotations (this is a trivial consequence of the general optical flow equations). Furthermore, if these points are not too far from the FOE, the terms in (u_r, v_r) which are quadratic in image coordinates will be negligible compared to the constant and linear terms, and hence may be ignored. We will call the points satisfying these conditions the set of *registration* points. As a consequence, the displacement field for such registration points is extremely simple:

$$\begin{aligned} u = u_r &= \{Bf\} + \{-Cy\} \\ v = v_r &= \{-Af\} + \{+Cx\}. \end{aligned}$$

We note that this displacement field is just a rotation by an angle C around the origin,

followed by a translation $(Bf, -Af)$ in the image plane, the combination of which is just a general rigid transformation of the image plane. If we now consider the set of all such registration points, it is clear from the displacement field above that these registration points move from one image to the next as a rigid body.

The idea in [14] is to use an optical flow algorithm, such as that of Anandan [8], to find point correspondences between successive frames of the unregistered image sequence. Picking a certain set of such points and considering it as a (two–dimensional) rigid body, one can define the center of mass of this set of points, and the principal axes of its inertia tensor. The rotation angles A and B can then be found by finding the displacement of the center of mass of this rigid body between the frames in question, and the rotation angle C can be found by finding the rotation of the principal axes between the two frames. Thus, the inter–frame rotational component of the camera's motion can be found.

Once these rotations have been found, then the registered image can be obtained by subtracting off the rotational component of the displacement at each pixel. Unfortunately, this is a nonlinear transformation, and hence will be computationally expensive. As a consequence, the registered image is found by linearizing the rotational component, that is by performing the inverse of the rigid motion, i.e., translate the second image by $(-Bf, +Af)$, then rotate it by $-C$ around the origin.

In practice, the registration algorithm of [14] translates the second image by $(-Bf, +Af)$, rounded off to the nearest whole pixel, and does not perform the rotation by $-C$ around the origin. The former approximation is made in the interests of computational efficiency, since sub–pixel interpolation would be expensive. The latter approximation is made since the quantization grid of a rotatated image does not coincide with the initial quantization grid, so a complicated sort of interpolation would be necessary.

4.2 Difficulties with the Registration Algorithm

There are a number of difficulties with the registration algorithm described above, but the primary one is this: How do you find the set of registration points, those environmental points which are infinitely distant from the image plane, yet whose image projections are close to the FOE ?

The answer, of course, is that it is difficult (if not impossible) to know these things a priori. To be sure, in some domains one might be able to specify a region of the image in which distant points might be expected to be found, but in general, one may not know this. The second point we want to make is that genuinely "infinitely distant" points are rare, so the question then becomes one of whether distant points are distant "enough."

The basic idea is that an environmental point is distant "enough" if its image displacement due to translation is much smaller than its image displacement due to rotation. Only then can one say that the bulk of the displacement is due to rotation. We can easily derive a condition that expresses this. From the time-adjacency relation (1), we see that the displacement d due to translation is just

$$d = D\frac{W'}{Z}. \tag{24}$$

For image points near the center of the image, their displacement is, as we have shown previously, of order Af (we assume that $B = C = 0$ for simplicity). Combining these two results, we see that the translational displacement (d) will be much smaller than the rotational displacement (Af) if

$$D\frac{W'}{Z} = d \ll Af \Rightarrow \frac{Z}{W'} \gg \frac{Z_0}{W'} = \frac{D}{Af}. \tag{25}$$

This therefore implies that a point is "distant" enough only if Z/W is much larger than

Z_0/W, where

$$\frac{Z_0}{W} \equiv \frac{D}{Af}. \tag{26}$$

In Table 3 we list the values that the quantity Z_0/W takes for various values of rotation Af and various distances D from the "FOE" (in pixels). For example, with a rotation of 0.2 degrees and a point at a distance of 25 pixels from the "FOE", $\frac{Z_0}{W}$ is 23. Therefore in our experiment with a translation of four feet between frames (i.e. W = 4 feet) a point must be greater than 92 feet for the approximation to hold. For other rotations and distances from the FOE, the values required can be considerably larger. We see that except for quite large angles (of the order of several degrees), and image points quite near the FOE, "distant" points are unlikely to be found in typical images.

Table 3: **Minimal distances Z_0/W for which points can be considered "distant".** Only points for which $Z/W \gg Z_0/W$ can be considered "distant" (see text for an explanation)

Rotation A	Af	Distance D in pixels			
(degrees)	(pixels)	25	50	75	100
0.1	0.5	50	100	150	200
0.2	1.1	23	45	68	91
0.5	2.7	9.3	18	28	37
1.0	5.4	4.6	9.3	14	19
2.0	10.8	2.3	4.6	6.9	9.3
5.0	27.0	0.9	1.9	2.8	3.7

Additional difficulties with the registration algorithm are of an implementational nature. The approximation of the (nonlinear) rotational displacement by its linear terms will introduce errors, especially in the displacements of peripheral points (which are precisely those with the greatest displacement in general, and which, as we saw in the previous section are therefore those which should receive the greatest weight in the finding of the FOE). The rounding off of the rigid displacement $(-Bf, +Af)$ to the nearest pixel could

introduce errors of up to 100 % in the removal of the rotational parameters (for instance, if $Bf = 0.51$ pix, it would be rounded off to 1 pix). The neglect of the "roll" term by not performing the rotation about the origin by $-C$ could introduce severe difficulties into an FOE–finding algorithm.

There are, as mentioned in [14], several ways of eliminating some of the problems with the registration algorithm. The most severe problem with the algorithm, the difficulty of ensuring that the registration points are distant points, could be minimized if, for instance, motion were integrated with stereo, or with some sort of active sensing. The latter modules could then be used to select regions in the image which contain distant points.

Other difficulties, which are essentially related to the finite resolution of the image plane, would be minimized by working with high–resolution images. In this regard, it should be noted that aside from the correspondence component, the complexity of the registration algorithm is independent of the resolution.

Table 4: **Recovered FOE's with and without Registration.** The image size is 256 by 256 with (0,0) at the centre. The estimated FOE is around (27,80).

Frames	FOE (unregistered frames)	Registration Correction in pixels (x,y)	FOE (registered frames)
1–3	(177,149)	(-0.1, 1.9)	(44,135)
3–5	(162,47)	(-0.8, 1.1)	(261,163)
5–7	(226,142)	(0.0, 1.6)	(36,111)
7–9	(244,109)	(0.0, 1.6)	(282,126)
9–11	(243,109)	(0.1, 1.1)	(51,76)

In order to see how the the combination of the registration algorithm and the FOE–finding algorithm performs, we ran the combined algorithm on several frames from the image sequence described in the next Section, and illustrated in Figure 5. The position

found for the FOE from each pair of frames is shown in Table 4. The "correct" FOE position (found from known camera parameters and measured vehicle motion) should be somewhere around $(x_0, y_0) = (27, 80)$, where the origin is in the center of the image, and the distances are given in pixels. We see that some of the values found for the FOEs are not too bad, while others are completely erroneous (although even the best in this experiment would not be sufficient for depth extraction). We conclude that this combination of algorithms does not generally give reliable FOEs.

The solution to our problem would seem to be an algorithm that finds both the rotational and translational motion parameters (i.e the FOE) at the same time. That is, we should consider a general motion algorithm. In the next section we consider such an algorithm constructed from the work of Anandan and Adiv at UMass, and show that promising results are obtained on the CMU NAVLAB image sequence.

5 Depth from a General Motion Algorithm

In this section we review a general motion algorithm to estimate the motion parameters and depth as implemented at the University of Massachusetts. We then show the results of applying this algorithm to a real image sequence. Finally, we compare this algorithm to the approximate translational motion scheme discussed in the previous sections.

5.1 The General Motion Algorithm

The general motion algorithm implemented at the University of Massachusetts to compute 3-D motion parameters and depth is based on the work of Anandan [8,9,10] and Adiv [12]. We discuss the two phases of the algorithm below.

5.1.1 Displacement Field Determination

In the first phase Anandan's [8] algorithm is used for determining displacement fields. It operates on a pair of images and uses a hierarchical correlation matching approach. A multi-resolution, multiple spatial frequency channel representation of the images is used for efficient computation and accurate determination of displacement fields. Confidence measures indicating the reliability of each displacement vector are provided. A smoothness constraint is then used to correct unreliable vectors based on the reliable ones.

5.1.2 Depth and Motion Parameter Estimation

In the second phase the flow field, along with associated confidences, found via Anandan's algorithm is taken as input to Adiv's [11,12] algorithm. This algorithm consists of two steps, which we now describe.

In the first step the flow field is segmented into connected sets of flow vectors where each set is consistent with the rigid motion of an approximately planar patch. The segmentation is based on a modified version of the generalized Hough transform, with displacement vectors voting for the motion parameters. It is expected that each segment would correspond to the motion of a portion of only one rigid object. The segmentation approach makes it possible to deal with independently moving objects. When there are no independently moving objects, this stage of segmenting the flow field is unnecessary and the entire static environment can be viewed as a single rigid object in the second phase of the algorithm.

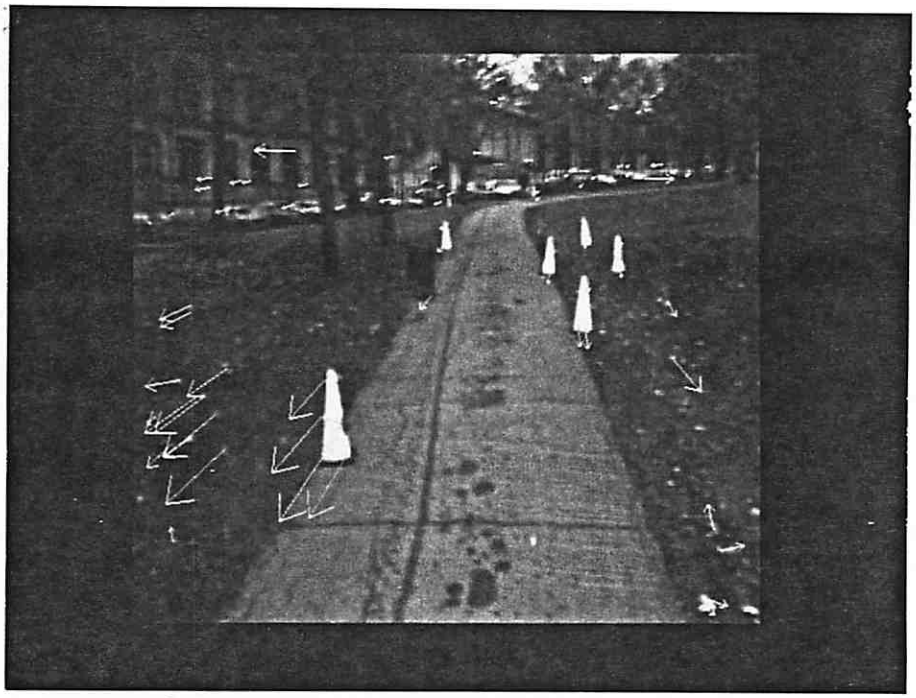
In the second step, the segments found in the first step are *grouped* together under the hypothesis that they have been induced by a single rigidly moving object (i.e. the

Table 5: **Depth Values of some points over a sequence of frames using the General Motion Algorithm.** The two tables used 100 and 200 points respectively. Depths are in feet. * and @ indicate respectively that the point was not among the top 100 or top 200 Moravec points. ** indicates that the point is absent in the image-pair.

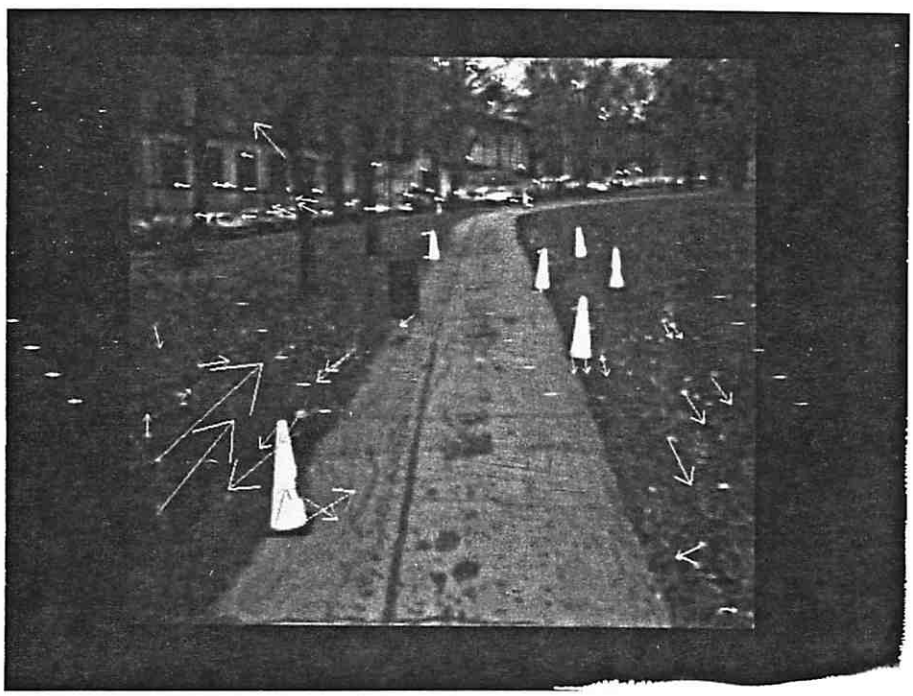
		1-3	1-3	3-5	3-5	5-7	5-7	7-9	7-9	9-11	9-11
Object	pts.	Exptl	True	Exptl	True	Exptl	True	Exptl	True	Exptl	True
cone1	1	65.7	76	58.3	72	61.7	68	50.3	64	61.2	60
	2	66.9	76	67.7	72	60.5	68	63.6	64	59.6	60
cone2	3	61.4	76	67.2	72	65.1	68	63.0	64	63.9	60
	4	60.8	76	82.3	72	56.2	68	61.7	64	61.7	60
cone3	5	50.2	56	59.2	52	46.3	48	40.8	44	38.4	40
	6	51.1	56	49.6	52	46.1	48	41.0	44	38.5	40
cone4	7	59.3	56	53.8	52	44.4	48	35.9	44	37.9	40
	8	46.3	56	53.3	52	47.6	48	41.8	44	39.8	40
can	9	44.1	46	44.4	42	47.6	38	41.8	34	39.8	30
	10	*	46	*	42	*	38	*	34	*	30
	11	*	46	*	42	*	38	*	34	*	30
cone5	12	31.0	36	32.2	32	26.0	28	22.0	24	20.0	20
	13	31.1	36	31.1	32	26.3	28	22.5	24	20.8	20
	14	31.9	36	30.9	32	28.5	28	21.9	24	20.5	20
cone6	15	18.1	21	*	17	**	**	**	**	**	**
	16	18.4	21	*	17	**	**	**	**	**	**
	17	18.9	21	-29	17	**	**	**	**	**	**
	18	18.6	21	-42	17	**	**	**	**	**	**

		1-3	1-3	3-5	3-5	5-7	5-7	7-9	7-9	9-11	9-11
Object	pts.	Exptl	True	Exptl	True	Exptl	True	Exptl	True	Exptl	True
can	9	38.1	46	43.3	42	36.5	38	32.2	34	29.6	30
	10	40.4	46	@	42	@	38	30.3	34	32.4	30
	11	44.2	46	42.5	42	@	38	32.7	34	@	30

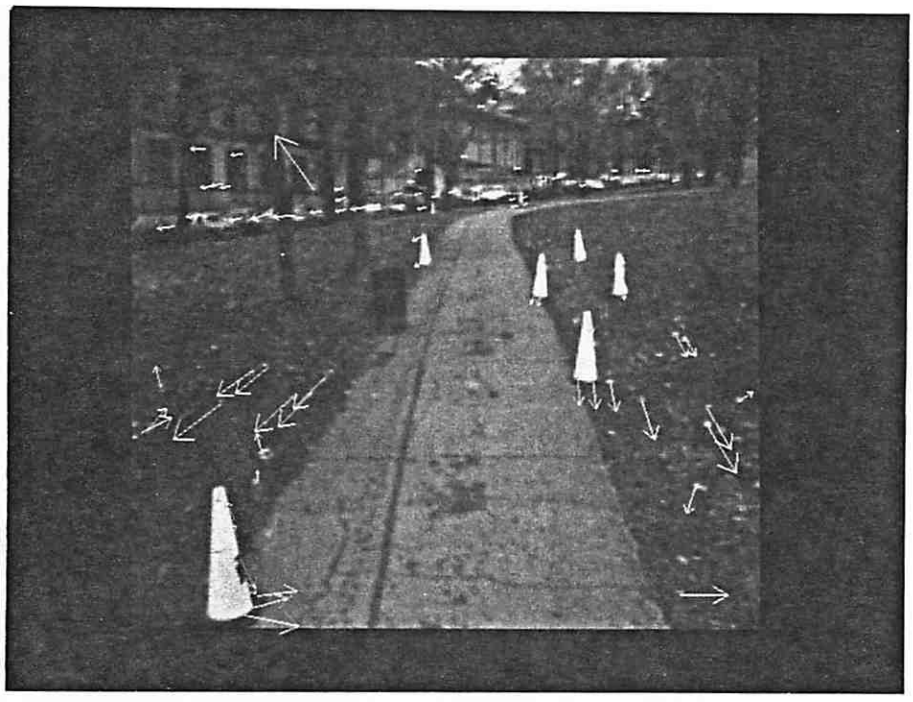
Figure 5: The sequence of image frames



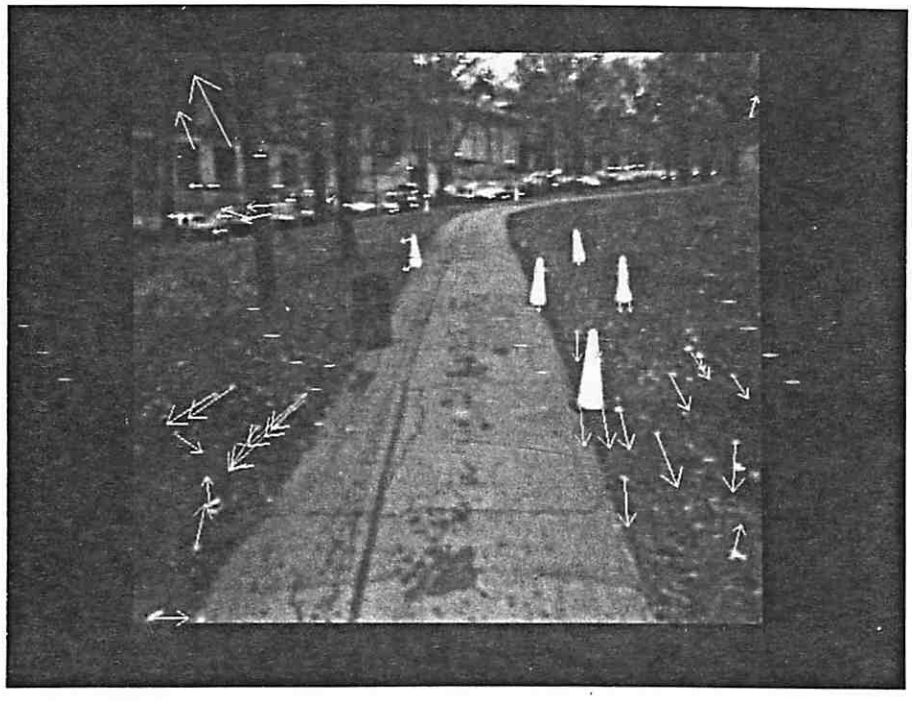
Frame 1 with displacement vectors for 1 - 3



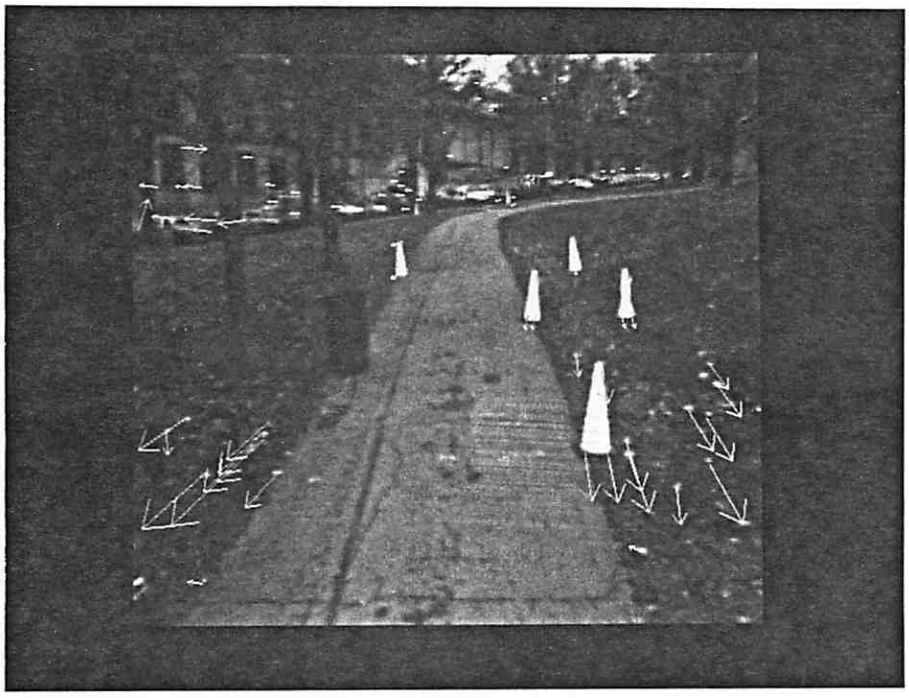
Frame 3 with displacement vectors for 3-5



Frame 5 with displacement vectors for 3-5



Frame 7 with displacement vector for 7-9



Frame 9 with displacement vectors for 9-11

Frame 11

planar surface assumption is dropped). This is done by computing the optimal motion parameters and related error measure for each segment by employing a least-squares approach that minimizes the deviation between the measured flow fields and that predicted from the estimated motion and structure. For each hypothesized FOE the optimal rotation parameters and the related error value are computed. A multiresolution discrete sampling technique is used to compute the minimum value of the resulting error function. *This step essentially involves grouping segments of the flow field which are consistent with the same motion parameters.*

Following the computation of 3-D motion parameters the depth can easily be computed if the total translation between frames is known.

5.2 Experimental Domain

In an effort to determine ground truth for the experiments a sequence of 20 images were taken using the NAVLAB at Carnegie-Mellon University. The field of view of the camera was measured, as was the inclination of the camera axis from the road plane. Traffic cones were placed at measured locations in the environment, and the actual interframe translation of the vehicle was measured and marked on the roadway, so that ground–truth depth data could be reasonably precisely obtained. We obtained a sequence of about 20 usable images some of which are shown in Figure 5.

5.3 Experimental Results

Experiments to determine depth and motion parameters were conducted using the general motion algorithm described above. To speed up computation of Anandan's algorithm,

Table 6: **Motion Parameters obtained through the General Motion Algorithm.** The frame pairs are at 4 ft. intervals. The results have been tabulated for 100 Moravec points and 200 Moravec points. (U,V,W) is the unit translational vector and (A,B,C) are the rotational components in degrees

100pts	1–3	3–5	5–7	7–9	9–11
U	-0.09	-0.09	-0.09	-0.09	-0.09
V	-0.25	-0.25	-0.25	-0.25	-0.25
W	-0.96	-0.96	-0.96	-0.96	-0.96
A	-0.19	0.17	-0.10	-0.04	-0.03
B	0.39	0.56	0.53	0.49	0.43
C	-0.30	0.01	0.07	0.06	0.28
200pts	1–3	3–5	5–7	7–9	9–11
U	-0.09	-0.16	-0.09	-0.09	-0.09
V	-0.25	-0.21	-0.25	-0.25	-0.25
W	-0.96	-0.96	-0.96	-0.96	-0.96
A	-.19	0.11	-0.10	-0.03	0.03
B	0.41	0.17	0.53	0.49	0.43
C	-0.22	-0.52	0.10	0.07	0.31

"interesting" or distinctive points were found via the Moravec operator in the images and flow fields determined only at these locations. Two sets of experiments were performed - one using 100 points and the other using 200 points). The flow fields are shown in Figure 5. False matches are sometimes produced, particularly near the boundary of the image due to feature points moving out of the image. This is particularly noticeable in the frame-pairs (3-5) and (5-7).

Adiv's algorithm was run on the flow fields produced by Anandan's algorithm. As mentioned earlier the segmentation part of Adiv's algorithm was unnecessary because there is in effect only one moving object. The motion parameters produced by Adiv's algorithm are shown in Table 6.

The translation results are all the same except for the frame pair 3-5 run with 200

points. The result suggests that Adiv's algorithm finds the translation vector fairly well. The less accurate results for frame pair 3-5 run with 200 points are because of false matches at the boundary of the image aggravated by choosing points which are not distinctive.

The results also indicate an approximate constant rotation of about 0.4 to 0.5 degrees about the Y-axis, a small and varying component about the X-axis, and a random rotation about the Z-axis. This is consistent with

1. (x-axis): a road surface which deviates very slightly from being planar either because of bumps or the surface itself being non-planar.

2. (y-axis): a small drift of the vehicle to the right.

3. (z-axis): some random motion probably due to the vehicle roll.

However , we do not have any measurements of the rotational components to verify how correct the derived rotational parameters are.

The depth results (for the set with 100 points) are shown in Table 5 for the obstacles on the road (the points are labelled in Figure 6). Since several points are missing in the case of the can,for the can we also show depth values when 200 interest points were used. In general, these are fairly good when compared to other attempts at the recovery of the depth of points (average error roughly 15%). The results improve for some of the later frame-pairs. Occasionally absurd depth values are returned because of false matches; this is particularly noticeable for cone 6 when it is very close to the edge of the image. For points which are at a distance, the depths returned are usually reasonable with some absurd values returned because occlusion causes false matches. The depth values obtained by using 200 interest points show small variations from the ones in table 5. Since in most cases the translation parameters found and the displacements of the points are the same,

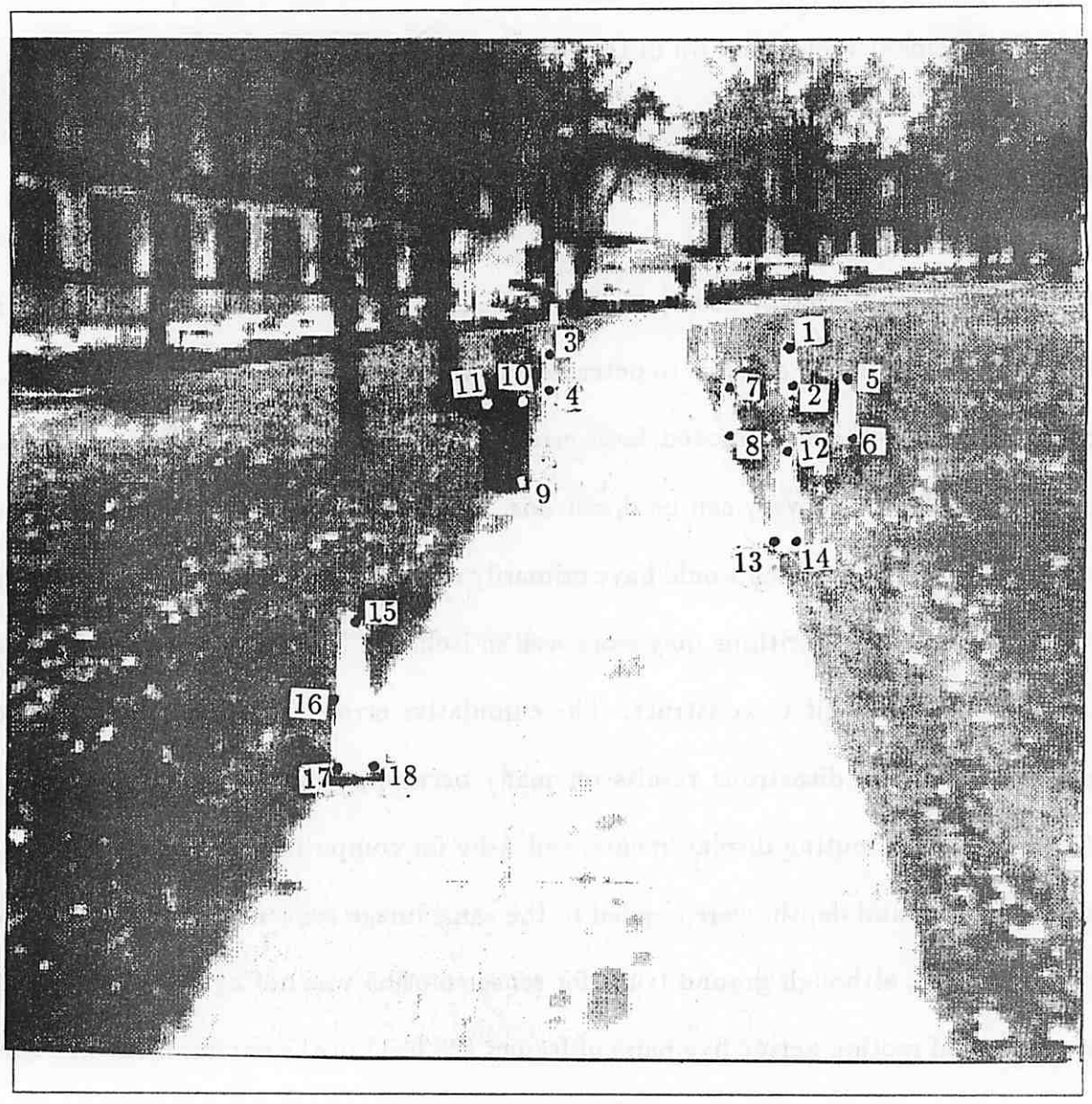
this suggests that depth computation is very sensitive to the accuracy with which the rotational parameters are computed.

It should be noted that in view of the results reported by earlier researchers regarding structure computations this appears to be a very promising result. It also appears as if this technique of solving the motion problem is superior to the methodologies which rely on approximate translational algorithms. It should also be noted that the displacement vectors found by Anandan's algorithm are indeed remarkable as can be seen from the displacement vectors shown in the images at the end of this paper.

6 Conclusion

It is our contention that the computation of the depth of points from approximately known translational motion is difficult to determine in the presence of even small rotations. If the rotational effects are not removed, large errors in the location of the FOE can occur, and the effect upon depth recovery can be disastrous. One scheme to eliminate small rotations by using distant points, which would have primarily rotational displacements was not effective. While each of the algorithms may work well in isolation from each other, a global system is much more difficult to construct. The cumulative errors from the subsystems tend to propagate to give disastrous results on many occasions. The algorithms developed by Anandan for computing displacements, and Adiv for computing the parameters of general sensor motion and depth, were applied to the same image sequence. The results appear to be quite good, although ground truth for sensor motion was not available. The extracted translational motion across five pairs of frames (20 feet) in the environment was constant, while the rotation was computed to be small; (we note that no ground truth was available for the rotational components). The patterns of these rotations seems quite reasonable for

Figure 6: The interest points marked on frame 1 of the image sequence



the context of the vehicle moving on a planar path. The extraction of depth from many of the objects was reasonably good, and sometimes surprisingly accurate at distances of 40 feet or greater.

The use of a gyroscopically stabilized platform would allow the simpler translational algorithms to be applied in extracting the depth of points. Translational motion processing should be computationally more efficient and effective. As an alternative to the algorithms described in this paper, a variety of other methods have been proposed to determine depth and motion parameter from image sequences. Accurate depth determination through methods based only on stereo are probably difficult to apply from a moving vehicle because of the rather small baseline. However, a combination of stereo and motion has been proposed [23,24,25], including recent research from our group and presented in this volume [26], in a form that might provide additional constraints to be effective. Alternatively the detection of features like lines might be more effective in depth computation. Current research (also presented in this volume) describe token-based techniques for computing correspondence of linear structures over time in order to extract depth from looming [27,28]. Initial experimental results have shown cases where depth extraction has been quite accurate.

In summary, there are currently no practical motion systems providing robust and accurate determination of depth in real scenes. However, there are promising directions actively being explored. Nevertheless, it will require considerable effort to develop a working system which will perform robustly in view of many of the problems that occur in real world situations.

7 Acknowledgements

We would like to thank Harpreet Sawhney, Rakesh Kumar and Yigal Gur for help with different aspects of the experiments.

References

[1] D. T. Lawton. *Processing Dynamic Image Sequences from a Moving Sensor.* PhD thesis, University of Massachusetts at Amherst, 1984. COINS TR 84-05.

[2] D. T. Lawton. Processing translational motion sequences. *Computer Graphics and Image Processing*, 22:116–144, 1983.

[3] I. Pavlin, E. Riseman, and A. Hanson. *Translational Motion Algorithm with Global Feature Constraints.* Technical Report COINS TR 86-58, University of Massachusetts at Amherst, 1986.

[4] I. Pavlin, E. Riseman, and A. Hanson. Analysis of an algorithm for detection of translational motion. *Proceedings of DARPA Image Understanding Workshop*, 388–398, December 1985.

[5] M. A. Snyder. Uncertainty analysis in image measurements. *Proceedings of the DARPA Image Understanding Workshop, Los Angeles, California*, February 1987.

[6] S. Bharwani, E. Riseman, and A. Hanson. Refinement of environmental depth maps over multiple frames. *Proceedings of the IEEE Workshop on Motion: Representation and Analysis*, 73–80, 1986. Also in *Proceedings of the DARPA Image Understanding Workshop*, December 1986.

[7] Ronald C. Arkin. *Towards Cosmopolitan Robots: Intelligent Navigation in Extended Man-Made Environments.* PhD thesis, University of Massachusetts at Amherst, 1987. COINS TR 87-80 , page 304.

[8] P. Anandan. A unified perspective on computational techniques for the measurement of visual motion. *IEEE First International Conference on Computer Vision*, 219–230, 1987.

[9] P. Anandan. *Measuring Visual Motion from Image Sequences.* PhD thesis, University of Massachusetts at Amherst, 1987. COINS TR 87-21.

[10] P. Anandan and R. Weiss. Introducing a smoothness constraint in a matching approach for the computation of optical flow fields. *Proceedings of the IEEE Third Workshop on Computer Vision: Representation and Control, Bellaire, Michigan*, October 1985.

[11] Gilad Adiv. *Interpreting Optical Flow.* PhD thesis, University of Massachusetts at Amherst, 1985. COINS TR 85-35.

[12] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.

[13] C. Thorpe, S. Shafer, T. Kanade, and the members of the Strategic Computing Vision Lab. Vision and navigation for the carnegie mellon navlab. *Proceedings DARPA Image Understanding Workshop*, 143–152, February 1987.

[14] I. Pavlin and M. Snyder. The problem of registration of dynamic image sequences. 1987. Internal Memo, Computer Science Department, University of Massachusetts at

Amherst.

[15] R. Y. Tsai and T. S. Huang. *Three Dimensional Motion and Structure from Image Sequences.* New York: Springer-Verlag, 1983.

[16] J. L. Barron, A. D. Jepson, and J. K. Tsotsos. The sensistivity of motion and structure computations. *Proceedings sixth national conference on Artificial Intelligence*, 700–705, 1987.

[17] J. Q. Fang and T.S. Huang. Solving three dimensional small-rotation motion equations. *Proceedings Computer Vision and Pattern Recognition*, 253–258, June 1983.

[18] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of 3-d motion parameters and surface structures of rigid objects. *Image Understanding 1984*, 135–171, 1984.

[19] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall Inc., 1982.

[20] A. Bandopadhay and R.Dutta. Measuring image motion in dynamic images. *Proceedings of the IEEE Conference on Motion: Representation and Analysis*, 67–72, May 1986.

[21] A. Bandopadhay and R. Dutta. Measuring motion in dynamic images: a clustering approach. *Proceedings Sixth Canadian Conference on Artificial Intelligence*, May 1986.

[22] B. K. P. Horn. Motion fields are hardly ever ambiguous. *International Journal of Computer Vision*, 1(3), October 1987.

[23] A. M. Waxman and J. H. Duncan. Binocular image flows - steps towards stereo-motion fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (vol 8 no 6), 1986.

[24] A. M. Waxman and S. Sinha. Dynamic stereo: passive ranging to moving objects from relative image flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (vol 8 no 4), 1986.

[25] M. Jenkin and J. K. Tsotsos. Applying temporal constraints to the dynamic stereo problem. *Computer Vision Graphics and Image Processing*, 1986.

[26] Poornima Balasubramanyam and M. A. Snyder. Computation of motion in depth parameters: a first step in stereoscopic motion interpretation. *Proceedings DARPA Image Understanding Workshop*, 1988. Cambridge, Massachusetts.

[27] Lance R. Williams and Allen R. Hanson. Translating optical flow into token matches. *Proceedings DARPA Image Understanding Workshop*, 1988. Cambridge, Massachusetts.

[28] Lance R. Williams and Allen R. Hanson. Depth from looming structure. *Proceedings DARPA Image Understanding Workshop*, 1988. Cambridge, Massachusetts.