Using Collinear Points to Compute Egomotion and Detect Nonrigidity

Niels da Vitoria Lobo and John K. Tsotsos*
Dept. of Computer Science,University of Toronto,Toronto, Canada M5S 1A4.
e-mail: niels@vis.toronto.edu, tsotsos@vis.toronto.edu

Abstract

We present a novel approach to computing egomotion, and detecting points not moving rigidly with the scene when an observer moves with unrestricted motion. The approach, using collinear image points, is based on an exact method for cancelling effects of the observer's rotation from optic flow. For each point only the component of velocity normal to the direction of the line joining the collinear points is needed. The algorithm is surprisingly simple, appears robust and is ideal for parallel implementation.

1 Introduction

For some time now, researchers have examined how the 3-D motion and the shape of the environment might be perceivable from the projected motion arising out of relative motion between a monocular observer and the scene, and two computational approaches to solving this problem have arisen. One uses the corresponding positions of image features in two or more images to recover the shape and motion parameters (Ullman 1979, Tsai and Huang 1984), and the other uses an instantaneous description of motion at one or more instants to accomplish the recovery (Longuet-Higgins and Prazdny 1980). This paper is concerned with the instantaneous formulation in which instantaneous 3-D velocity (i.e., time-derivatives of 3-D position) of scene primitives (points, texture elements, lines, etc.) relative to a moving observer, is related to instantaneous image velocity (time-derivatives of projected position) of the corresponding image primitives. This approach is premised on the belief that the prior step of measurement of an approximation to instantaneous image velocity (also termed *image flow* or *optic flow*) can be carried out. This assumption is not an altogether unrealistic one if allowances are made for noise in this measurement process, and some progress has been made towards achieving such measurements (see Schunck 1986, Watson and Ahumada 1985, Heeger 1988, Fleet 1990, Woodham 1990).

The subsequent step, that of computing shape information and the 3-D motion parameters from the instantaneous image velocity, has received ample attention from vision researchers. However, in addition to the fact that every 3-D algorithm proposed so far is not robust to noise in the input instantaneous velocity, the algorithms typically suffer from other important problems. Algorithms that assume general rigid motion and unrestricted shapes have to search in spaces that have at least three dimensions (Bruss and Horn 1983, Aloimonos *et al.* 1987, Mitiche 1986), and the non-linear numerical methods used are very sensitive to the initial guess. Others assume some restricted form of motion (eg., no rotation in a certain dimension; see Barron 1988), or restrict the allowable shapes (eg., planarity), in order to get closed-form solutions for the unknowns (Waxman and Wohn 1987), or assume that some parameters are known and solve for the others (Ballard and Kimball 1983, Matthies *et al.* 1989) — but these assumptions are too restrictive.

We present a novel method for computing the motion parameters for the case of unrestricted motion and unrestricted shape. The method is simple, is robust to noise and is implementable in parallel. We also show how the technique can straightforwardly detect points that do not move consistently with the rigidity assumption of the scene.

1.1 Previous Related Work

One approach to obtaining egomotion and shape information from image velocity has been to attempt to cancel the rotation component within image velocity in order to directly relate the measured image velocity and the translation component and depth. Longuet-Higgins and Prazdny (1980) suggested it, and Reiger and Lawton (1985) investigated it, but only succeeded in achieving it in an approximate manner. They assumed that the rotational effects at neighbouring positions in the image are similar, so that a subtraction of two image velocities at neighbouring positions (i.e.,differential flow) would cancel rotation, and that the resulting vector would therefore point towards the intersection point of the translation vector and the image plane. However, this assumption is typically invalid and the simplistic subtraction introduces noise from residual rotation.

Only recently has it been shown that the rotation component of instantaneous velocity can be cancelled from the image velocity measurements in a principled manner to assist in the 3-D parameter estimation problem. Prazdny (1983) shows that the rotation component can be cancelled for two image points if the component of flow that is along the line joining the two image points is used. This was used by Nelson and Aloimonos (1988) to construct an obstacle avoidance algorithm, by da Vitoria Lobo and Tsotsos (1989, 1990) to show that for three non-collinear image points, the pair-wise relative depths of the three scene points are dependent only on the unknown 3-D direction of translation and the known image velocities and image positions (i.e., that, in principle, knowledge of rotation is irrelevant to the calculation of shape from motion), and by Heeger and Jepson (1989, 1990) to compute egomotion. Another distinct way to cancel rotation in an exact manner is to employ three collinear image points, utilizing the component of velocity that is normal to the line joining the three image points (da Vitoria Lobo and Tsotsos 1989, 1990; Weinshall 1990a,b has a similar result, but in that work it was not made clear that rotation is being cancelled). This normal component of image velocity is typically easier to compute, so the algorithm we present here based on the collinear point result is of practical importance.

Here, we first review the collinear point result, *viz.* that taking a generalized approximation of the second derivative of the image velocity component that is normal to the direction of the derivative yields a relationship between measured image velocities, the translation parameters and depth information. Then we show how to use this result straightforwardly to compute the direction of translation, via what is our new *FOE algorithm*. We show the usefulness of this by recovering relative-depth using the non-collinear result of

*John K. Tsotsos is the Canadian Pacific Fellow of the Canadian Institute for Advanced Research. This work was supported by the Natural Sciences and Engineering Research Council and the Information Technology Research Center, a Province of Ontario Center of Excellence. The range data came from the Range Image Database of NRC Canada. The authors are grateful to Allan Jepson, Gregory Dudek and John Barron for helpful discussions.

CH2983-5/91/0000/0344/$01.00 © 1991 IEEE

da Vitoria Lobo and Tsotsos (1989, 1990). However, since this non-collinear result is unable to obtain robust shape information in the presence of noise, we postpone a presentation of steps for obtaining shape data in a robust manner to future work. Instead, we extend this approach to detect points not moving rigidly with the rest of the scene, so that these points can be prevented from corrupting the shape recovery process, or can be analyzed by other processes, should this be warranted. Experimental results and robustness tests accompany the theoretical ideas in the paper, followed by a discussion and conclusion.

2 Computing the Direction of Translation

In this section, we compute the direction of translation in a simple manner using collinear points. We first introduce the imaging geometry and the equations governing velocity. Next we cover the basic relationship between the direction of 3-D translation and what is commonly called the Focus of Expansion. Then, after we show how collinear points yield the direction of translation, we describe the new *FOE Algorithm* and experimentally investigate its robustness to noise.

Our preliminaries are those of Longuet-Higgins and Prazdny (1980). Using a left-handed viewer-centered 3-D coordinate system, let a point in the image plane at unity have image coordinates (x, y), and the 3-D coordinates of the corresponding scene point be (xZ, yZ, Z) for some positive Z. Let the relative motion between the viewer and the point be described by an instantaneous translation and an instantaneous rotation, (U, V, W) and (A, B, C), with respect to the viewer-centred 3-D coordinate system. Then the projected velocity in the image plane at position (x, y) is (u, v), (Longuet-Higgins and Prazdny 1980), where

$$\begin{aligned} u &= \frac{-U+xW}{Z} - Axy + B(x^2+1) - Cy, \\ v &= \frac{-V+yW}{Z} - A(y^2+1) + Bxy + Cx. \end{aligned} \tag{1}$$

2.1 The Focus of Expansion

We define the Focus of Expansion (FOE) to be the position of intersection of the imaging surface[1] and the axis of instantaneous observer translation. This[2] defines the FOE to be independent of the viewer's instantaneous rotation.

For an image plane at unity, this position is given by some (x_o, y_o) in the image plane. The direction of translation is then given by either $(x_o, y_o, 1)$ for expansion, or $(-x_o, -y_o, -1)$ for contraction. Thus the 3-D translation vector (U, V, W) can be expressed as (Wx_o, Wy_o, W) for some W. When there is no observer translation, the FOE does not exist. When there is translation with no component in depth, the FOE lies at infinity (in the direction of (U, V)) for the case of a planar imaging surface; however, for a hemi-spherical imaging surface, the FOE lies at the rim of the hemi-sphere. The important point here is that according to our definition of FOE there is a 1-to-1 mapping between the direction of translation (upto a sign factor) and the FOE; in presenting our ideas we will go back and forth between them depending on which is more intuitive at the time. Figure 1 illustrates how rotation is completely irrelevant to the position of the FOE.

[1] This surface could be planar, hemi-spherical, etc.

[2] This definition differs from that of Regan and Beverley (1982) and Cutting (1986) who define the FOE as the intersection of the image and the direction of motion, thus making it dependent on rotation. Perhaps, to avoid confusion, one should think of our FOE as the Translational FOE.

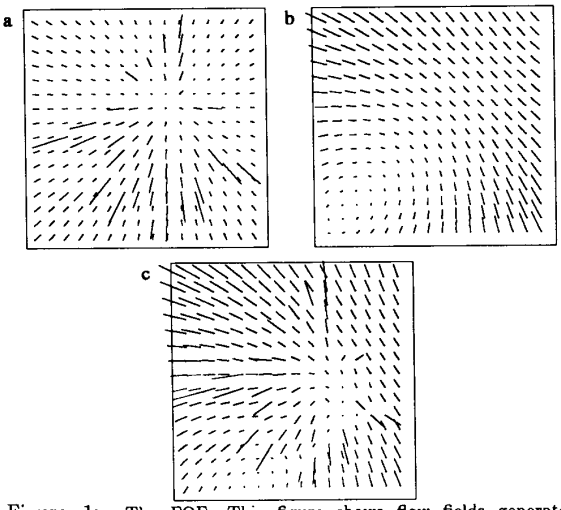

Figure 1: The FOE: This figure shows flow fields generated when observer a) only translates (translation parameters [U,V,W]= [2.2,2.2,10.1]), b) only rotates (rotation parameters [A,B,C] = [-0.04, -0.04,0.04]), and c) moves with both rotation and translation, the most typical situation. In the second case there is no FOE because there is no observer translation, while in the first and third cases the FOE is in the same position for both cases, just above and to the right of the image center. i.e., our definition of FOE renders its position in the image completely independent of observer rotation.

Several authors have researched the computation of the FOE either by restricting the allowable motion to translation or by approximately cancelling rotation(Lawton 1981, Jain 1982, Reiger and Lawton 1985). In work related to ours, Weinshall (1990a,b) finds the FOE but needs to find elliptical surface patches beforehand. Our contribution is that we find the FOE in an exact manner, even when the motion includes general rotation, with no surface shape restriction.

2.2 FOE From Collinear Measurements

We will often refer to three image points as a *triplet*. We first show how a computation involving three *collinear* image velocity measurements relates 3-D translation to depth, by cancelling rotation. Then we show how it can be used to find the FOE. The computation is a generalization of an approximation to the second derivative of the velocity component that is normal to the line joining the image points, the "derivative" being taken in the direction of the line. The measurements from the three collinear image points are assumed to be associated with three scene points on the *same* rigid structure, although not necessarily on the same surface nor on physically connected surfaces. For points subscripted by i, $(i = 1, 2, 3)$, let the image coordinates of the points be (x_i, y_i), their image velocities be (u_i, v_i), and the depth values of their counterparts in the scene be Z_i.

The computation that is needed is the weighted[3] sum[4],

$$\begin{aligned} Sum \stackrel{\text{def}}{=} & (-\sin\theta)(nu_1 - (m+n)u_2 + mu_3) + \\ & (\cos\theta)(nv_1 - (m+n)v_2 + mv_3) \end{aligned} \tag{2}$$

[3] When the collinear points are equi-spaced, the weights are (1,-2,1) which is an approximation to the second derivative (Rektorys 1969).

[4] We use the symbol $\stackrel{\text{def}}{=}$ to define a calculation; this is to distinguish it from an expression of equality.

where θ is the angle that the line through the three image points makes with the image x-axis, going from the x-axis to the line in a counterclockwise manner, and where m is the distance between the first and second image points and n the distance between the second and third. By substitution of (u_i, v_i), this generalized *Sum* can be verified[5] to be

$$Sum = \begin{pmatrix} U\sin\theta - V\cos\theta & + \\ Wy_3\cos\theta - Wx_3\sin\theta & \end{pmatrix}\left(\frac{n}{Z_1} - \frac{(n+m)}{Z_2} + \frac{m}{Z_3}\right) \quad (3)$$

i.e., *Sum* = product of translation factor and depth factor.

In da Vitoria Lobo and Tsotsos' (1990) work this was used to show that if the direction of translation is known, as with static stereo, then "curvature" information, i.e. collinearity, convexity, concavity are determined. Weinshall (1990a,b) used this to classify regions as planar, elliptic, cylindrical and hyperbolic. Zisserman and Cipolla (1990) use similar ideas in tracking image curves to get surface curvature.

The zero conditions of *Sum* are the following: either the depth factor is zero (in which case the scene points are collinear[6]), or the translation factor is zero. This translation factor can be expressed as an inner product of (U,V,W) and $(\sin\theta, -\cos\theta, y_3\cos\theta - x_3\sin\theta)$. Besides when the translation (U,V,W) itself is zero, this translation factor is zero only when the line passing through the three image points also passes through the FOE. To see this, substitute into the translation factor of equation (3) the following: $U = Wx_o$, $V = Wy_o$ (by our definition of FOE), and observe that $y_3 = (\frac{\sin\theta}{\cos\theta})(x_3 - x_o) + y_o$ must follow, which implies that (x_o, y_o) lies on the line through the triplet. Thus, in general, for a scene containing sufficient depth variation, if we compute many such collinear triplet sums across the complete image, the FOE will be in the position of intersection of many triplet lines for which, in each case, the triplet sum is zero. That is, we can look for the most evidence for the location of the FOE (see Hough techniques, in Ballard and Brown 1982.) Next, we describe an operator and an algorithm to achieve this.

2.3 The FOE Operator

We choose to implement the FOE algorithm in the following manner. Consider an operator (called the *FOE operator*) with many lines passing through its center (centered at some (x,y) position in the image) in many directions in the image (see figure 2), such that each line passes through several image points. Along the line, overlapping triplets of image velocity are used and summed (see figure 3). In this paper, we overlook the issues of choosing the optimal *intra-triplet spacing*, i.e., how far apart the points within each triplet should be, and of choosing the optimal *inter-triplet spacing*, i.e., how far apart the centers of adjacent triplets should be.

The absolute values of all triplet sums (as defined by equation 2) along a line are summed to give a *LineSum*, and then the LineSums are summed to get a response at the center. For a rigid scene, there are three reasons why the response could be zero. First, each of the triplets summed by this operator could be a collinear triplet in the scene; however, along each line the triplets overlap, thus implying that for this situation to occur the whole scene would have to be a single plane, which is quite rare. Second, there could be no translation. Third, the true FOE could be at the position of the center of the operator. Fortunately, the former two are distinguishable from the third by the fact that if either of the two were to hold, the operator would have to respond with zero everywhere; whereas, in the third case, assuming the first two do not hold, there will be a unique zero. So, the first two cases can be detected easily and eliminated. Thus, we can sweep this operator across the whole image to obtain a response map, detecting where it gives a zero response surrounded by sufficiently non-zero responses.

The response map can obviously be computed by operators in parallel, after choosing an appropriate *inter-operator spacing*. Then the operator with the zero response must be found. If the inter-operator spacing is large (i.e., if the response is coarsely sampled), one may have to interpolate to find a zero. In practice, due to noise in the input flow field we will not get a zero in the response map but will have to be satisfied with the global minimum. In the case where the FOE lies on the imaging surface, and where the noise is uncorrelated, the global minimum typically corresponds to the FOE, the direction of translation.

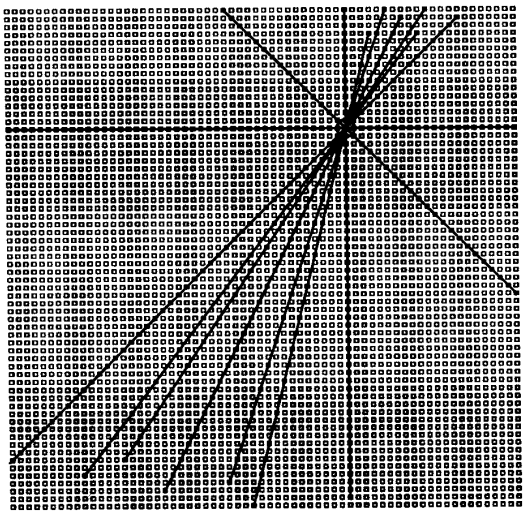

Figure 2: The FOE Operator: This shows how an operator is made up of intersecting lines of points in the image. On a regular 256x256 square grid of points, at a single point we can have about 60-70 intersecting lines each of which passes through at least several image points. Here, only 9 lines are shown. The image is a dense grid of points, shown as hollow squares, each with a flow estimate associated with it. Some of the points used by each line have been blackened to identify them. Along each line a *Line Sum* is calculated (see figure 3). The Line Sums are added to give the response of the operator at position (x,y) which is where the lines intersect.

2.4 Robustness of the FOE computation

In this section, we test the FOE algorithm and examine its robustness to noise in the input image velocity field. To judge the accuracy of the results we need synthetic data for which the motion parameters are known in advance. After computing the direction of translation we can also compute

[5] Verification: The p_i lie on a line with slope $\frac{\sin\theta}{\cos\theta}$ (in this footnote, we ignore the case where $\cos\theta = 0$, as in this case the verification is easy). So, we can use the substitutions $y_1 = (\frac{\sin\theta}{\cos\theta})(x_1 - x_3) + y_3$, and $y_2 = (\frac{\sin\theta}{\cos\theta})(x_2 - x_3) + y_3$. Also, assume without loss of generality that the ordering of the points is such that $x_1 < x_2 < x_3$. Then, we can use $x_3 = x_2 - n\cos\theta$, and $x_2 = x_1 - m\cos\theta$ to obtain the desired result.

[6] See da Vitoria Lobo and Tsotsos (1989) for a proof.

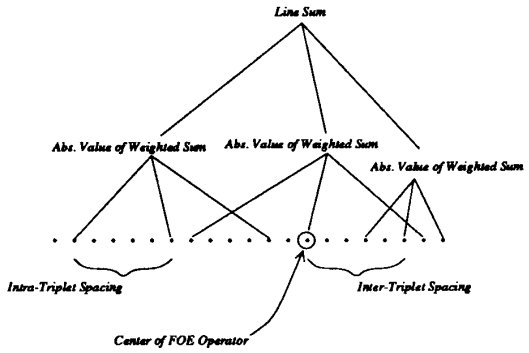

Figure 3: This shows how points on a line were processed. Flow estimates at points are grouped into triplets and summed according to the weighted sum of equation (2). Then the absolute values of these weighted sums are summed to give the *Line Sum* for the line. For now we overlook issues of the optimal *intra-triplet spacing*, i.e., how far apart the points within each triplet should be, and the optimal *inter-triplet spacing*, i.e., how far apart the centers of adjacent triplets should be.

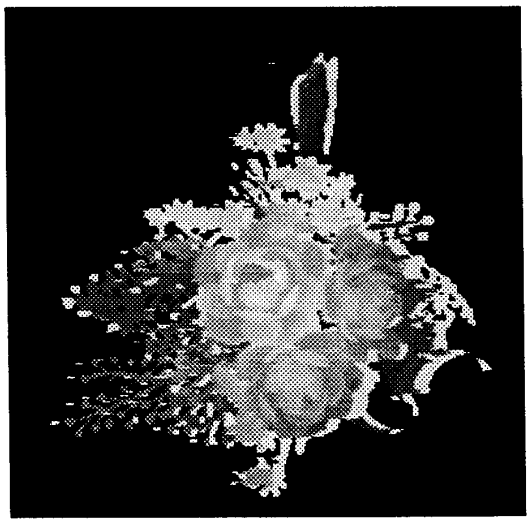

Figure 4: Range data that was used to generate flow field used in these experiments. Observer moves with some instantaneous 3-D rotation and translation parameters with respect to these depth points and the flow equations are used to give an image velocity vector at each image point.

relative depth by the algorithm (for non-collinear triplets) in da Vitoria Lobo and Tsotsos (1989, 1990). However, the algorithm for relative depth, without further enhancements, is best suited to be run on noise-free flow data, and hence will only be used in the tests involving ideal data.

We used the range image shown in Fig 4 to generate the synthetic flow field shown in Fig 5. This image contains the kind of depth variation that a perceiver may encounter in typical situations. The translation and rotation parameters used to generate this flow field were (U,V,W)=(4.5,8.5,10.0) and (A,B,C)=(0.004,0.003,0.004). Using equations (1), the field acquired flow values at every pixel position. It was used

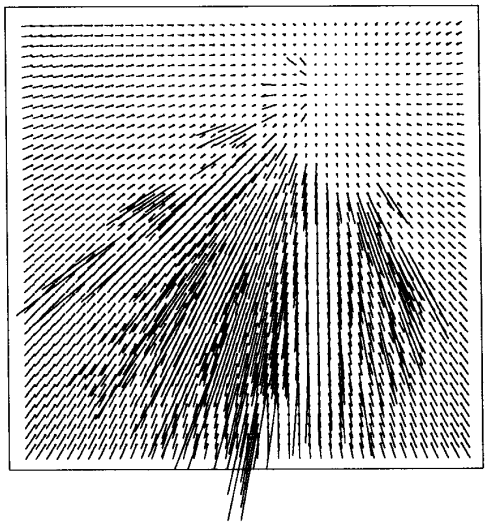

Figure 5: Flow field generated using 3-D motion and the range data shown earlier. The rotation used was (0.004,0.003,0.004) and the translation used was (4.5, 8.5, 10.0). Field shown sub-sampled.

as input to our FOE algorithm and the direction of translation was computed. This direction was then used for the relative depth computation of da Vitoria Lobo and Tsotsos (1990). For the implementation of the FOE algorithm, an FOE operator was centered at each pixel. Each FOE operator consisted of 24 lines passing through it (see Fig 2). Along each line we centered a triplet at each point on that line. (Note that the spacing of points on a line depends on the slope of the line. See Fig 2 and inspect the spacing between the blackened pixels for the different lines.) Within each triplet, the intra-triplet spacing was such that three alternate points on the line were used.

Fig 6 shows the response map for the operator as a function of (x,y). All response maps are shown with brightness proportional to the *Log* (response). The darkest point in the map is the global minimum; this corresponds to the computed FOE and it is not surprising that it is exactly correct. Fig 7 shows the depth obtained from the computed relative depth and one known depth value.[7] In this reconstruction, errors are virtually non-existent.

The above ideal flow field was then made more realistic by adding random noise that was on average 8% of each of the image velocity estimates. The results are still good: the correct direction of translation was found easily. The filter response map is shown in Fig 8; note that the position of the minimum has not changed. As mentioned earlier, in this case the reconstruction of relative depth is not shown, because the relative depth computation requires near error-free flow estimates. It will be necessary to eliminate the noisy flow estimates so that reliable estimates of relative depth can then be achieved for those points where the flow estimate is reliable. Alternatively, given that one now has an accurate estimate of the direction of translation, one can solve for the unknown rotation parameters in a least squares manner and then use these motion parameters to recover depth as Matthies *et al.* (1989) or Heeger and Jepson (1990) do.

[7] Without having this single known depth value, it is impossible to do any better than to compute relative depth.

Figure 6: Response map for noise-free flow. In these maps, we show brightness proportional to *Log*(response). For this input, the global minimum is exactly at the true FOE.

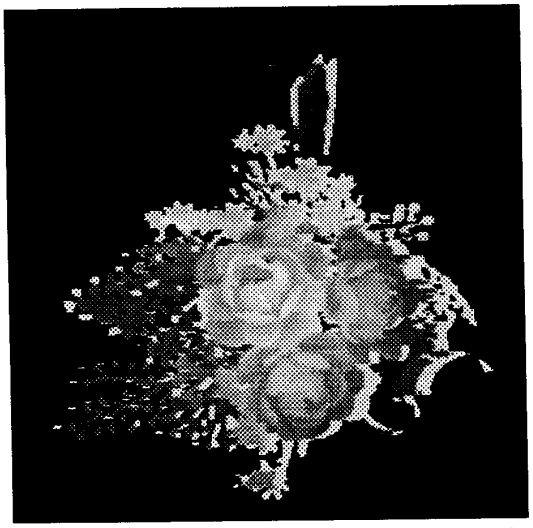

Figure 7: Reconstructed depth produced from the computed relative depth.

2.5 Factors Influencing Noise Tolerance

We found that the noise tolerance of the FOE computation depends on the amount of depth variation in the scene. For instance, when the range data was moved away from the simulated observer before generating the synthetic flow data, so that the scene is flatter now than it was before, the noise tolerance dropped. When the range data was close to the simulated observer the algorithm was quite stable.

Intra-triplet spacing also influenced the sensitivity to noise. Using wider spacing seemed to increase robustness, but this may have been an artifact of the particular choices of range data used. A deeper study of both these factors in an empirical and an analytical manner is in progress.

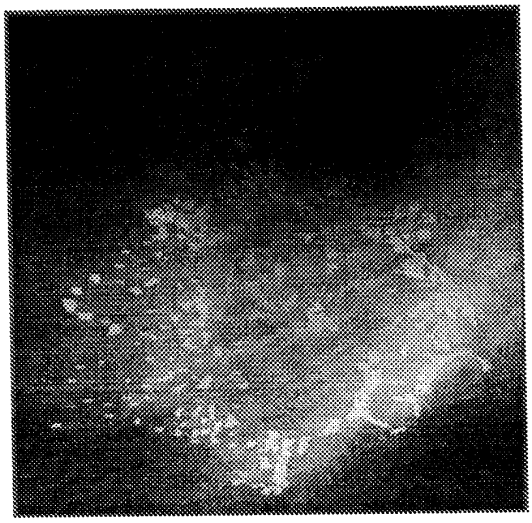

Figure 8: Response map for the noisy flow field.

3 Functioning in a non-rigid scene

Objects in a moving observer's view will often move independently — a typical scene is rarely perfectly rigid. It is important that any 3-D reconstruction algorithm remain competent in these situations. Fig 9 depicts the flow field generated by combining the original rigid flow field used in the previous section with the flow for an independently moving rectangular patch in the upper middle of the image. The frontoparallel patch translates upward, to the right, with an image flow magnitude comparable to those of the flower petals. The FOE algorithm was tested on this combined flow field. The output direction of translation is still correct (filter response map in Fig 10.) Tests indicate that even with larger patches the FOE computation is accurate, demonstrating robustness to patches of non-rigidity in the scene.

3.1 Finding points that thwart rigidity

For a 3-D motion algorithm that maintains competence despite independently moving scene parts and noise, the next step in its enhancement is to enable it to detect parts of the scene where the overall-rigidity assumption does not hold.

A point may be measured as moving differently from the rest of the scene either because it may be a noisy measurement, in which case when the shape reconstruction step is being carried out this erroneous measurement should be treated with caution, or because it could legitimately belong to an independently moving part of the scene. If this is the case, this moving part will probably need additional attention and possibly some special purpose processing. Such processing might lead to segmenting the moving objects from their background, or segmentation into parts (in articulated motion). Analysis of the non-rigidity may also provide data for classifying objects in scene recognition.

As a first step, it is desirable to have our algorithm signal that certain points are moving inconsistently with respect to the rest of the scene. We can straightforwardly extend our algorithm to indicate whether or not all three points in a triplet lying on a line through the FOE are moving with

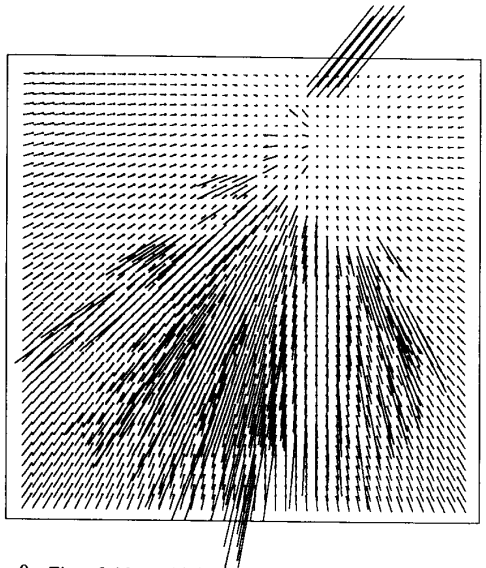

Figure 9: Flow field combining the original rigid flow field and an independently moving patch in upper right part of image. The frontoparallel rectangular patch translates upward, to the right, with an image flow magnitude that is comparable to those of the flower petals.

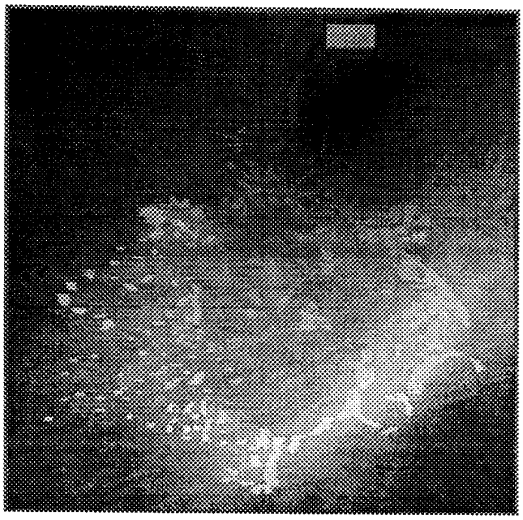

Figure 10: Response map for the flow associated with the nonrigid scene in which a patch moves independently. The global minimum is in the same position, indicating that the FOE computation is robust to non-rigidity in the scene.

the 3-D motion of the rest of the scene: simply calculate the triplet *Sum* for the three points and test if it is near zero or not[8]. When sufficiently above zero, at least one of the three points is inconsistent with the rest of the scene. A practical way to implement this is to use the operator that gave the minimum response indicating that the FOE is at its center. We simply traverse each of its various lines (we only used 24 lines in this implementation) searching for triplets that don't give near-zero sums.Figure 11 (with an inset) shows the region around the patch in the flow field of Figure 9 being marked by the program as a set of inconsistent triplets. Note that because the patch itself is moving in a planar fashion, the triplet sum, when all three points are inside the patch, is zero. So, at an independently moving planar region, only the boundary areas of the region will be detected (this depends on the intra-triplet spacing used). For an independently moving non-planar region, all the triplets, in any way overlapping the region's points, will be detected.

4 Discussion

We have investigated the use of a straightforward method for obtaining the direction of translation, and for detecting points not moving rigidly with the scene. This can be implemented in parallel, and the operator is a simple summation of sums. Earlier work had showed that the direction of translation yielded relative-depth, i.e., that, in principle, knowledge of rotation is irrelevant to the task of recovering shape. Our computation of the direction of translation has proved to be reasonably robust to noise, in tests to date.

The basis of the technique is a method of combining collinear points which allows rotation to be cancelled in an exact manner. Along any straight line in the image, the rotational contribution to the image velocity component that is orthogonal to the line varies linearly with length, so that taking approximations to the second derivative of this component of velocity cancels out rotation. Thus our operator simultaneously cancels out rotation exactly and samples the translation contribution to find the FOE. We sample the flow field densely with this operator. If rotation is the only 3-D motion occuring, i.e., the translation is zero, the operator will have a zero response everywhere. The same is true if the complete scene is a single plane. However, if neither of the above situations hold, and if the scene is rigid, there will be a unique minimum in the response map. Experiments show that even with scene non-rigidity caused by independently-moving objects, the minimum is found easily. Future work will study how much non-rigidity can be tolerated. Once this minimum has been located, it is used to detect subsets of points moving inconsistently with the rest of the rigid scene, serving as a first step towards motion segmentation.

In principle, a hemi-spherical retina suffices to ensure that all possible directions of translation intersect the imaging surface. In practice, however, one may know in advance that there will always be a substantial component of translation along the line of sight, in which case one may retain the planar imaging surface used in the analysis in this paper.

The fact that this computation essentially only needs normal velocities (normal to the line of collinearity) is an interesting aspect of the algorithm. In practice, for now, we require a dense set of normals, which is equivalent to requiring an almost equally dense set of two-dimensional flow estimates. This density requirement is the only remaining major potential drawback. However, it is possible that the algorithm will perform competently even with a sparser set of normal flow vectors. On the other hand, research in the measurement of image velocity is not that far off from delivering the requisite level of dense measurements (Fleet 1990, Woodham 1990).

We also need to investigate the performance of the algorithm when it is run on image displacement estimates (point-to-point correspondences) rather than instantaneous velocity

[8] We have already shown in Section 2 that if such a collinear triplet were moving with the same rigid parameters as the scene, the triplet *Sum* must be zero.

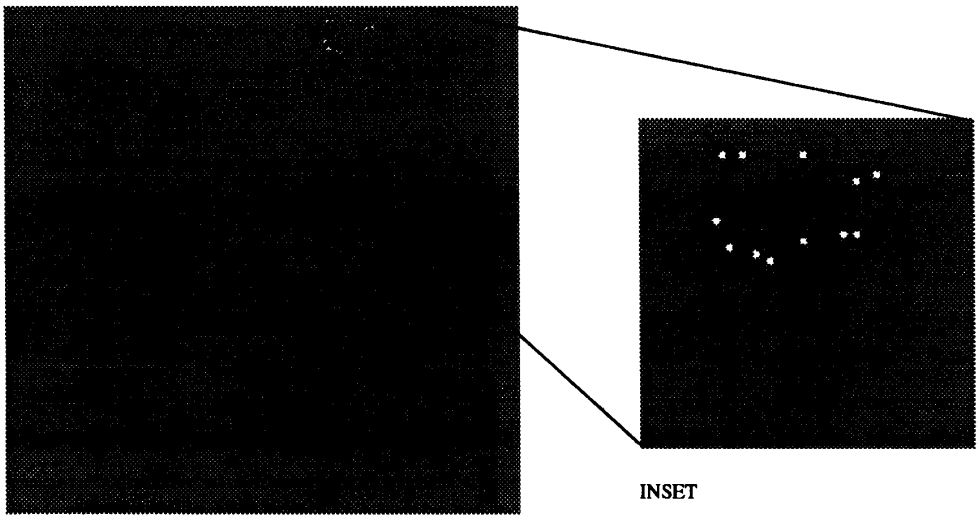

Figure 11: The program marks points in triplets that overlap the patch in the flow of Figure 9 as inconsistent triplets. Because the patch itself is moving in a planar fashion, the triplet sum, when all three points are inside the patch, is zero. So, these internal points are not marked.

estimates. Adiv (1985) has shown that the instantaneous approximation to image displacement is acceptable if the temporal sampling rate is high, the rotation is small, and the translational component in depth is small relative to the scene depth. Thus while we expect error in the computed direction of translation, we do not expect it to be large.

5 Conclusion

We have presented a novel approach to computing the translation component of egomotion, and detecting points not moving rigidly with the scene, in the view of an observer moving with unrestricted motion. The approach uses collinear subsets of points. The subsets enable the exact cancellation of rotation from the flow field. No assumption of smooth surface structure is made and in fact the computation is more robust to noise when there is significant depth variation in the scene. For these collinear points, only the component of the flow that is normal to the line of collinearity is needed. The algorithm is surprisingly simple, and much of it is parallelizable. The tests of robustness are promising, and further analytical work is being carried out to study the aforementioned issues arising from this work.

References

Adiv G. (1985), "Determining 3-D motion and structure from optical flow generated by several moving objects", *IEEE PAMI*, **7**, 384-401.

Aloimonos J.Y., Weiss I., Bandhopadhay A., (1987), "Active Vision," *Proc. ICCV*, 35-54.

Ballard D.H. and Brown C. (1982), **Computer Vision,** Prentice-Hall.

Ballard D.H. and Kimball O.A. (1983), "Rigid body motion from depth and optical flow," *CVGIP*, **22**, 95-115.

Barron J. (1988), "Determination of egomotion and environmental layout from noisy time-varying image velocity information in monocular image sequences," *Ph.D. Thesis*, Dept. of Computer Science, Univ. of Toronto. Available as RBCV-TR-88-24.

Bruss A.R. and Horn B.K.P. (1983), "Passive navigation," *CVGIP*, **21**, 3-20.

Cutting J. (1986), **Perception with an Eye for Motion**, MIT Press.

da Vitoria Lobo N. and Tsotsos J.K. (1989), "Shape Information from Image Motion: What Can We Compute from Three Points?" *RBCV-TR-89-32*, Dept of Comp Sci, Univ. of Toronto, December.

da Vitoria Lobo N. and Tsotsos J.K. (1990), "Extracting Qualitative Shape from Image Motion: Applications to Stereo-Pairs", *Proc. AAAI Workshop on Qualitative Vision*, pp. 36-40, Boston, July.

Fleet D. (1990), "The Measurement of Image Velocity", *Ph.D. Thesis*, Dept. of Computer Science, Univ. of Toronto.

Heeger D. (1988), "Optical flow using spatiotemporal filters", *IJCV*, **1**, 279-302.

Heeger D. and Jepson A. (1989), "Visual perception of 3-D motion," MIT MediaLab TR-124, December.

Heeger D. and Jepson A. (1990), "Simple method for computing 3-D motion and depth," *Proc. ICCV*, Osaka, Japan, 96-100.

Jain (1982), "An approach for the direct computation of the focus of expansion," *Proc. PRIP-82*, 262-268.

Lawton D.T. (1981), "Optical flow field structure and processing image motion," *Proc. IJCAI-81*, 700-703.

Longuet-Higgins H.C. and Prazdny K. (1980), "The interpretation of a moving retinal image," *Proc. Roy. Soc. Lond.*, **B 208**, 385-397.

Matthies L., Szeliski R., Kanade T. (1989), "Kalman filter-based algorithms for estimating depth from image sequences," *IJCV*, **3**, 181-208.

Mitiche A. (1986), "On kineopsis and computation of structure and motion," *IEEE PAMI* , **8**, No. 1, 109-112.

Nelson R. and Aloimonos J. (1988), "Using flow field divergence for obstacle avoidance: towards qualitative vision," *Proc. ICCV* Florida, 188-196.

Prazdny K. (1983), "On the information in optical flows", *CVGIP*, **22**(2), 239-259.

Regan D.M. and Beverley (1982), "How do we avoid confounding the direction we are looking and the direction we are moving?" *Science*, **215**, 194-196.

ReigerJ.H. and Lawton D.T. (1985), "Processing differential image motion", *J. Optical Society of America*, **A2**, 354-359.

Rektorys K. (1969), **Survey of Applicable Math,** MIT Press.

Schunck B. (1986), "The image flow constraint equation", *CVGIP*, **35**, 20-46.

Tsai R.Y. and Huang T.S. (1984), "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE PAMI* , **6**, No. 1, 13-27.

Ullman S. (1979) **The Interpretation of Visual Motion,** MIT Press.

Watson A. and Ahumada J. (1985) "Model of human visual motion sensing," *J. Optical Society of America*, **A2**, 322-342.

Waxman A.M. and Wohn K. (1987), "Image flow theory: a framework for 3-D inference from time-varying imagery," Chapter Three in **Advances in Computer Vision**, ed. C. Brown, (Erlbaum Publishers).

Weinshall D. (1990a), "Shortcuts in the computation of qualitative 3-D shape and motion invariants," *Proc. AAAI-Qualitative Vision Workshop*, 31-35. Boston.

Weinshall D. (1990b), "Direct computation of qualitative 3-D shape and motion invariants," *ICCV*, Osaka, Japan, 230-237.

Woodham R. J. (1990), "Multiple light source optical flow," *Proc. ICCV*, Osaka, Japan, 42-46.

Zisserman A. and Cipolla R. (1990), "Qualitative surface shape from deformation of image curves," *Proc. AAAI-Qualitative Vision Workshop*, Boston, 41-45.