

**DETERMINING MOTION PARAMETERS FOR SCENES
WITH TRANSLATION AND ROTATION**

By

Charles Jerian and Ramesh Jain

Department of Electrical and Computer Engineering

June, 1983

CENTER FOR ROBOTICS AND INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109

Abstract

A study of methods that determine the rotation parameters of a camera moving through synthetic and real scenes is conducted. Algorithms that combine ideas of Jain and Prazdny are developed to find translational and rotational parameters. An argument is made for using hypothesized motion parameters rather than relaxation labelling to find correspondence.

Introduction

There have been many different approaches to extracting the motion information from sequences of dynamic scenes [Nag81a,Nag81b,ULL79a,ULL79b]. A number of these methods concentrated on determining the FOE or focus of expansion form of the translational parameters. Most of these methods require optical flow vectors or corresponding points in different discrete frames as input. Other methods seek to use the focus of expansion to yield the correspondence[LAW82]. The FOE as the intersection of all optical flow vectors does not exist when the system undergoes rotation, however we feel that the concept of an FOE can still be used in analyzing three-dimensional motion parameters and in solving the correspondence problem for such cases [PRA79b,PRA80,PRA81a,PRA81b].

One simple method for determining the focus of expansion in scenes where corner points or other matchable entities have been extracted has been proposed by Jain [JAI82a]. This method does not use a correspondence between the frames, rather it uses the triangle inequality to show that in approaching motion the sum of the distances of points in the second frame from the FOE minus the sum of the distances of points in the first frame from the FOE is maximized. This method is difficult to apply directly to real scenes because most corner detectors can lose points from frame to frame and can also find multiple points in both frames. The idea of hypothesizing a likely focus of expansion and verifying it in some way against the image is an important method. In this paper we will suggest other variations of this basic paradigm. In Prazdny's work also optimization of a quality of fit function was proposed to find the rotation parameters

from given optic flow vectors and this was tested on synthetic data [PRA81a].

As an alternative to this method a linear solution to rotation and translation parameters have been found by Tsai and Huang [TSA81,TSA82a,TSA82b]. Tsai and Huang have a linear algorithm for finding the direction of motion and the rotation parameters given optical flow. Their method is in contrast to the optimization methods in that it uses a system of linear equations. All other systems studied have non-linear equations in their methods.

Their algorithm was implemented in a straight forward way by least squares using subroutines from the *NAAS IMSL* package. The results indicated that there are problems with using low accuracy vectors, vectors for objects which are close in depth or vectors that are spatially close. These problems become worse for objects that are far away. This study showed that their method requires some modification and that the least squares approach might not be the best approach to use. It also shows that getting the rotation from the hyperbolic vs. linear error that occurs locally is next to impossible. If rotation rather than the more general distortion is desired a problem arises in that their method does not specifically obtain a rotation represented by 3 real numbers but 8 parameters of a 3 by 3 distortion matrix. Only a subset (the orthonormal matrices) actually are rotations. Therefore it is possible that the optimal solution is not constrained to be a rotation because of noise or other errors. In any case it seems that their method will not recover the motion of a synthetic version of Nagel's [DRE81a] car with the optical flow we can obtain using a least squares implementation [CPJ82].

Determining rotation in synthetic scenes using feature points

If a collection of points from a translating rigid body or, equivalently, if the observer translates, the disparity (or optical flow) vectors lie on lines that intersect at a point called the FOE. The FOE gives the direction of motion of the object or observer. Figure 1 shows the disparity vectors for an observer moving toward two boxes. These boxes are at position $(0,0,10)$ and $(0,10,25)$ where $(0,0,1)$ represent the center of the image. The boxes are inclined by a rotation of $-.035$ radians about the x-axis followed by -0.50 radians about the y-axis. The observer is displaced 2 units in the z direction. For this example the focus of expansion is at $(0,0)$ and the direction of motion is $(0,0,1)$.

Imagine a spherical retina whose center is at $(0,0,0)$. Then the lines in this figure would correspond to a collection of great circles of the sphere which pass through a pole at $(0,0,1)$. Now, if the imaging system is rotated before it is translated, a motion component is added to each point on the sphere along a latitude line of the axis of rotation. The projection of such latitude lines on a plane is discussed by Prazdny [PRA81a]. For rotations about axes on the x-y plane, these circles map onto hyperbolic arcs and, for rotations about the z-axis they map into circular arcs. The synthetic images in this section all are created using a $\frac{\pi}{2}$ radians field of view. This gives an aperture ratio of 1. Locally the hyperbolic arcs are almost linear, especially near the center of the camera's field of view. For a camera with a narrower field of view than our synthetic imaging system, they are fairly linear throughout the image. Rotation of the camera causes each point in the plane to be displaced in a way depending on its position on the plane. The actual mapping of image points (X,Y) to rotated points

$(NEWX, NEWY)$ under rotation about the y axis by ϑ radians is given by the equations:

$$NEWX = \tan(\arctan(X) + \Delta\vartheta)$$

$$NEWY = Y\sqrt{(1+NEWX^2)/(1+X^2)}.$$

Then, translation moves the points along lines passing through the FOE. The effect for most imaging systems, with a reasonable field of view, is almost constant over the entire plane. Note that such rotations can be closely simulated by a translation of an object, at distance r , by $r\vartheta$ in the direction perpendicular to the axis of rotation.

In figure 3, we have the same two objects as figure 1, however the camera is rotated .5 radians about the y-axis before translating 2.5 units along the z-axis. The motion vectors in figure 3 fall on lines (shown in figure 4) that tend to cluster at two intersection points along the x-axis. Any method that finds good possible FOEs will find likely values at these points. The object points that have these values would then be found. This facilitates a segmentation of the scene based on the variation in depth. It is observed that the intersection points all are roughly along a line (in this case the x-axis since the FOE is at $(0,0)$) and the axis of rotation is always perpendicular to that line. The axis of rotation therefore, can be quickly and easily determined by examining the local FOEs.

Different objects in the image can be at different depths. The points at each depth are observed to intersect near a single point. If the image is pre-segmented into regions that are each part of one object of approximately constant depth, a clear single intersection point for the optical flow vectors of that

region would result. This is the converse of using the distinct local peaks to help segment the image. Standard techniques such as finding the least squares intersection point may suffice to obtain reliable results for such a region. Similarly, Jain's method would give a local FOE for such a segmented region. The different local peaks would be fitted with an appropriate curve to find the direction and amount of rotation. The position of the local FOEs would give an ordering of the regions by depth.

Rotational parameters seem to operate orthogonally to the radial direction on which Jain's function depends, so that rotations have little consistent effect on its value. Consider the case where the FOE is at $(0,0)$ and rotation is about the z-axis. The value of the function at the correct FOE $(0,0)$ is not affected by any amount of rotation. The maxima may not be at $(0,0)$ anymore.

From a correct FOE if we draw radial lines to a distinctive feature point, its matching point is found on the same line within a certain distance that depends on the maximum velocity allowed. A smooth error measure can be devised that accounts for points that do not precisely fall on a line by finding the nearest matching point within a range of angles and using the sum of the absolute or square of the error the best matching points. If there is no match, a large amount such as π radians can be added to the sum of the errors. This allows us to effectively find an optimizable function for the FOE that can be tailored to find good peaks along the different clusters or that can average the values to find an average FOE. To find strong peaking the maximum angle for allowing a match is decreased and changes are made to the error function to cause poorer matches to weigh more heavily against a given choice for the FOE. Since rotation

of the camera is merely a mapping of the image to itself, the effects of such rotation can be reversed by performing an inverse mapping of the image using equations previously described here, or those of Prazdny [PRA81a]. Such transformations are called de-rotation mappings here to distinguish them from three-dimensional rotations. Since such a measure is very sensitive to error in the FOE caused by rotation it can be used to effect controlled de-rotation of the image by intersecting the vectors more tightly in the de-rotated image and reducing the error measure. This procedure does not require any a-priori match, however if any a-priori matching information is available the routine can selectively examine such matches to select the match that is most consistent with its hypothesized FOE.

Figure 5 shows the same objects as figure 1. The observer translates along the z-axis 2.5 units and rotates 0.5 radians about the z-axis. Figure 5 shows the optical flow vectors that result. Figure 6 shows the lines that extend these vectors to intersection points. As these figures show, the lines intersect on a circle, where the FOE of the lines is at the center of the circle. The width of the circle is proportional to the amount of rotation and inversely proportional to the translational velocity. If the rotation angle is zero the circle degenerates to a point. If the angle is π radians, the circle becomes the point at infinity. A circle detector could be used on the local FOE histogram to find the circle and obtain the FOE and amount of z-rotation. If we have only part of the circle is represented strongly because objects are more concentrated in some parts of the image than others, an averaging method would locate the FOE at the center of mass of the points on the circle rather than at the center of the circle. The amount of error would depend on the non-uniformity of the distribution and the radius of

the circle. Jain's algorithm for example, tends to find a value near such a center of mass. This leads to increasing inaccuracies if the feature points are non-uniformly distributed. Inaccuracy increases as the radius of the circle increases.

A local approach that would find the FOE in windows would be useful here since we could plot the local FOEs and apply a circle detector to find the FOE and amount of rotation.

As a variation of the above methods, an FOE can be computed by Jain's method using data that has been de-rotated using Prazdny's equations. The computed FOE is used to obtain a correspondence of the feature points. This involves a search in the radial direction from the FOE for a matching feature point. In the synthetic scenes first explored, only position information was present, so no comparison of the feature points was possible. To discern the presence and amount of rotation the following are computed: the distance in the x direction of the matched point, the perpendicular distance, and the difference in angle of the matching points. We only need to explore a few point in the neighborhood of a given point to find the matching point. All these points can be preselected by using a maximum velocity constraint. Additionally, if feature points have feature values, these would be used to further constrain the list to be searched. We de-rotate by moving the position of points in one of the images along hyperbolic paths. For de-rotation purposes rotation about an axis in the x-y plane can be replaced by a rotation about the x-axis followed by a rotation about the y-axis. To simplify the work only rotations about the y-axis were considered, however, everything will still work with an increase in cost for rotations

about two axes. The amount of rotation about a given axis is treated as a parameter to vary in an optimization method. The best average FOE is redetermined using Jain's method. It may be necessary to do much re-computation. To avoid this assume that the position of the best FOE varies smoothly and by a small amount with small changes in the amount of rotation. Then only a small amount of additional computation is required to find a new average FOE, given a new angle. In this method we have tested various parameters to determine the amount of error from the correct FOE. Among these is the mean square angular error of all the corresponding points. Other measures tested were the distance of the hypothesized line connecting the corresponded points along the x direction, from the FOE, and from the average of all the x positions. One could also find the least squares value of the intersections and determine the orthogonal distance of each line from the least squares value. A different error measure is the amount of spread of the clusters to the intersections.

Table 1 summarizes the results of an implementation of this theory. The table shows the extracted position of the FOE for different amounts of assumed rotation about the y-axis. The correct values used in generating the data would give an angle of -0.03 radians and locate the FOE at $(0.0, -1.00)$ on the x axis at the extreme left of the image. The column labelled match angular error shows the average of the sum of the squared differences in angular position of vectors from the extracted FOE to pairs of points that are established as corresponding. The correct result yielded the minimum error. The average of the sum of the squares of the angular errors was computed as a measure of the quality of the FOE and is reported in Table 1. Lines were drawn between the matched points to a line going through the computed FOE orthogonal to the x-axis. The actual

average position of the intersections of the extended optical flow lines and this line were determined. The x position column reports that position. If there is no systematic error this value should be the same as the x coordinate of the FOE. The variance of the intersections is noted under the variance column. This is an excellent measure of the spread of the intersection cluster and gives the best indication of rotation of all the measures. The FOE variance column reports the variance using the x-coordinate of the FOE rather than the refined x-position obtained from the intersecting vectors. As one can see from examining this table, all of the error measures are quite capable of determining the correct amount of rotation to within .01 radians.

The plot represents the search space for an optimization algorithm that seeks the peak of the surface. The function is examined over a coarsely quantized subset of its range. The range is quantized using equal angles of the x and y axes of a polar coordinate sphere. This allows for the uniform treatment of cases where the z velocity is much less than the x or y velocities or the rotation is large and the FOE is not on the image. Then a finer search of the area showing the best results in the first search is performed. This gives us a coarse to fine strategy. Figure 9 shows the FOE function for the de-rotated feature points that was generated by the program from the same run whose results are in Table 1. Figure 10 shows the close-up of the function near the correct value.

If there is a non-geometric check on correspondence available, then the quality of correspondence can be evaluated. The possibility of no-match could be included. Correspondence can be used with such equations as those of Tsai and Huang to extract rotation and translation parameters. If correspondence

can be verified with a non-geometrical test, then a hypothesize and test paradigm becomes possible. A preliminary correspondence of some points can be found. This can be used to determine some motion parameters. These parameters can then be used to establish a better correspondence for more points. The procedure can be iterated until a sufficiently satisfactory result ensues. If the process fails to converge, new starting values can be hypothesized from the data. A random sampling and consensus algorithm [FIS81] [FIS82] may be useful here.

Real scenes

Since edge points seem to cause problems for the methods described in this paper [CPJ82] corner points were used for work with real scenes. The input image consisted of a pair of 128 by 128 images quantized to 256 grey levels. The images contain a frame surrounding a cube. The observer is approaching the center of the curb with uniform velocity.

A corner detector suggested by Kitchen and Rosenfeld [KIT81] was applied to the two scenes and corner points were extracted. All corner points within a certain fixed distance determined by an estimate of the maximum allowed velocity (in this example 10 pixels) were considered as possible matching corner points. The square grey-level differences of 5 by 5 windows was used as an error measure. To more accurately locate the actual matching point a 3 by 3 neighborhood of the point in the previous frame was examined to find the lowest error match. At most five of the best such matching corner points are recorded in an array with the quality of the match for each corner point in frame 2.

The space of possible FOEs is searched on a coarse 10 by 10 grid. For a given point in frame 2, only those possible matching points determined earlier which are not further away from the FOE in frame 1 are tested for approaching motion, for receding motion only points which are not closer would be checked. Of these points, the point with the least difference in angle is selected, if there is such a point. A weighted sum of this angular error and the grey value error measure would be better here. The chosen point is the best match consistent with the FOE. Running totals of angular error and grey value errors are kept.

An attempt was made to minimize the error by changing the FOE. Many choices of the FOE result in an identical correspondence of points in frame 1 and frame 2. These choices yield the same sum total error. The angular error values are not constant for these values but they are less reliable because of quantization errors. One could attempt to more accurately optimize one of these functions but the results would be meaningless unless the disparity vectors are more precisely determined. With a correspondence of points in the two frames a conventional approach to finding the FOE can be used such as a least squares pseudo-intersection of the vectors. Such an answer is also no more accurate than the result that the point is in a region near the center of the image (the image is 128 by 128 with center at 64,64) between (40,40) and (60,60) because of the quantization error. One would obtain a high variance if such a method were applied. To more accurately position the FOE the flow vectors must be determined to a higher degree of precision using interpolation techniques. Since our method gives an excellent correspondence of the points the search costs of interpolation are greatly reduced. The more precise vectors can be intersected to find the FOE with more certainty. Figure 9 shows the disparity vectors

generated by this method for the real scene Figure 10 shows the intersecting lines and gives an idea of the quantization errors in angle and the uncertainty of the position of the FOE.

If the points extracted by this method are used with any version of Jain's algorithm, no reasonable answer results. For real scenes, Jain's [JAI82a] algorithm fails to deal with missing points and addition of new points. It also ignores the invaluable information contained in the feature values returned by a corner detector or contained in the actual grey-value window around the feature point.

Quantization problems leads to uncertainty with some of the methods proposed in this paper when they are applied to real scenes. If a point moves one pixel, its direction of motion is only discernable to $\frac{\pi}{4}$ radians precision. To correctly determine the differences between rotation and translation and to precisely fix the FOE information of a much higher resolution is required. To some extent these problems may be ameliorated by the use of image sequences rather than image pairs. Such sequences can be fitted with a curve and the curve used instead of the actual points in the calculations.

Conclusions

In this paper we have shown that methods that seek to use preliminary or hypothesized motion parameters to find correspondence and better motion parameters are viable alternatives to the usual paradigm of attempting to obtain correspondence or optical flow first and then to solve for motion parameters. This will be especially important in a system that has many sources of information that can generate hypothesized motion trajectories and integrate

these sources into the dynamic scene problem.

Acknowledgement

We gratefully acknowledge Richard Hill for his endless ideas, discussions and editorial advice.

References

- [CPJ82] Jerian, Charles. *Determining Motion Parameters for scenes with Translation and Rotation*. Wayne State University, Department of Computer Science, Masters Thesis, Detroit Michigan, 1982.
- [DRE82a] Dreschler, L. and H.H. Nagel. "Volumetric Model and 3D- Trajectory of a Moving Car Derived from Monocular TV-Frame Sequences of a Street Scene." *Proceedings IJCAI* (1981), Vancouver, Canada
- [DRE82b] Dreschler, L. and H.H. Nagel. "On the Frame-to-Frame Correspondence between Greyvalue Characteristics in the Images of Moving Objects." *Proceedings GI Workshop on Artificial Intelligence*, Bad Honneff, Germany, 1981
- [FIS81] Fischler M.A. and Bolles R.C. "Random Sample Consensus: A paradigm for model fitting with applications to image anlysis and automated cartography," *CACM* , Vol. 24(6). pp. 381-395 (June 1981)
- [FIS82] Fischler M.A., Barnard S.T., Bolles R.C., Lowry M., Quam L. Smith G. and Witkin A. "Modelling and Using Physical Constraints in Scene Analysis", in *AAAI-82* pp. 30-35
- [GLA81] Glazer, (?????first name-initial???) "A simple method for determining optical flow", in *IJCAI-81*
- [HIL80] Hill, Richard, "Determining optical flow", Wayne State University Computer Science Department Masters Thesis (1980)
- [HOR81] Horn, B.K.P. and B.G. Schunck. "Determining Optical Flow". *Artificial Intelligence* 17 (1981) 185-203
- [JAI82a] Jain, R. "An Approach for the Direct Computation of the Focus of Expansion", in *PRIP-82* pp. 262-268
- [KIT80] Kitchen, L. and Rosenfeld A., "Gray-level Corner Detection", Computer Vision Laboratory, Computer Science Center. University of Maryland College Park, MD 20742 TR-887 April, 1980
- [LAW82] Lawton D.T. "Motion Analysis via Local Translational Processing" in *Workshop on Computer Vision: Representaion and Control* pp. 59-72
- [MOR77] Moravec, H.P. "Towards Automatic Visual Obstacle Avoidance." *Proceedings 5th IJCAI* (August 1977), p. 584.

- [MOR79] Moravec, H.P. "Visual Mapping by a Robot Rover." *Proceedings 6th IJCAI* (1979), pp. 598-600.
- [MOR80] Moravec, H.P. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Stanford: Stanford University, Department of Computer Science, Ph.D. Thesis, 1980; also Stanford AI Lab Memo AIM-340; also Computer Science Department Report Number STAN-CS-80-813; also Pittsburgh: Robotics Institute, CMU-RI-TR-3, September 1980.
- [NAG81a] Nagel, H.H. "On the Derivation of 3-D Rigid Point Configuration from Image Sequences." *Proceedings IEEE Conference on Pattern Recognition and Image Processing* (1981).
- [Nag81b] Nagel, H.H. and B. Neumann. "On 3-D Reconstruction from Two Perspective Views." *Proceedings IJCAI* (1981).
- [PRA79b] Prazdny, K. "Motion and Structure from Optical Flow." *Proceedings 6th IJCAI* (1979), pp. 702-704.
- [PRA80] Prazdny, K. "Egomotion and Relative Depth map from Optical Flow." *Biological Cybernetics*, 36 (1980), pp. 87-102.
- [PRA81a] Prazdny K. "Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinearly Moving Observer" *CGIP* 17 (1981) 238-248
- [PRA81b] Prazdny K. "A Simple Method for Recovering Relative Depth Map in the case of a Translating Sensor" *IJCAI-81*, pp. 698-699
- [TSA82a] Tsai R.Y. and Huang T.S. "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", in *PRIP-82* pp. 112-118
- [TSA81] Tsai R.Y. and Huang T.S. "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch", in *IEEE Transactions on ASSP* Volume ASSP-29 no. 6 (December 1981) pp. 1147-1152
- [TSA82b] Tsai R.Y. and Huang T.S. "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch", in *IEEE Transactions on ASSP* Volume ASSP-30 no. 4 (August 1982) pp. 525-534
- [ULL79a] Ullman, S. *The Interpretation of Visual Motion*. Cambridge, Massachusetts: MIT Press, 1979.
- [ULL79b] Ullman, S., "The Interpretation of Structure from Motion," *Proc. Royal Soc. London*, Vol. B, No. 203, pp. 405-426

TABLE 1 ROTATION ABOUT THE Y-AXIS DE-ROTATION RESULTS						
Angle	Foe		Match angular error	X-axis intercept		
	X axis	Y axis		X position	variance	Foe variance
-.04	-.04	-1.00	9.2E-6	.05	1.3E-2	1.4E-3
-.03	.04	-1.00	2.8E-6	.03	5.0E-5	1.4E-4
-.02	.09	-1.00	6.6E-6	.11	2.0E-2	2.0E-2
-.01	.18	-1.00	2.5E-5	.18	1.6E-2	1.6E-2
-.00	.22	-1.00	4.1E-5	.25	1.1E-1	1.1E-1

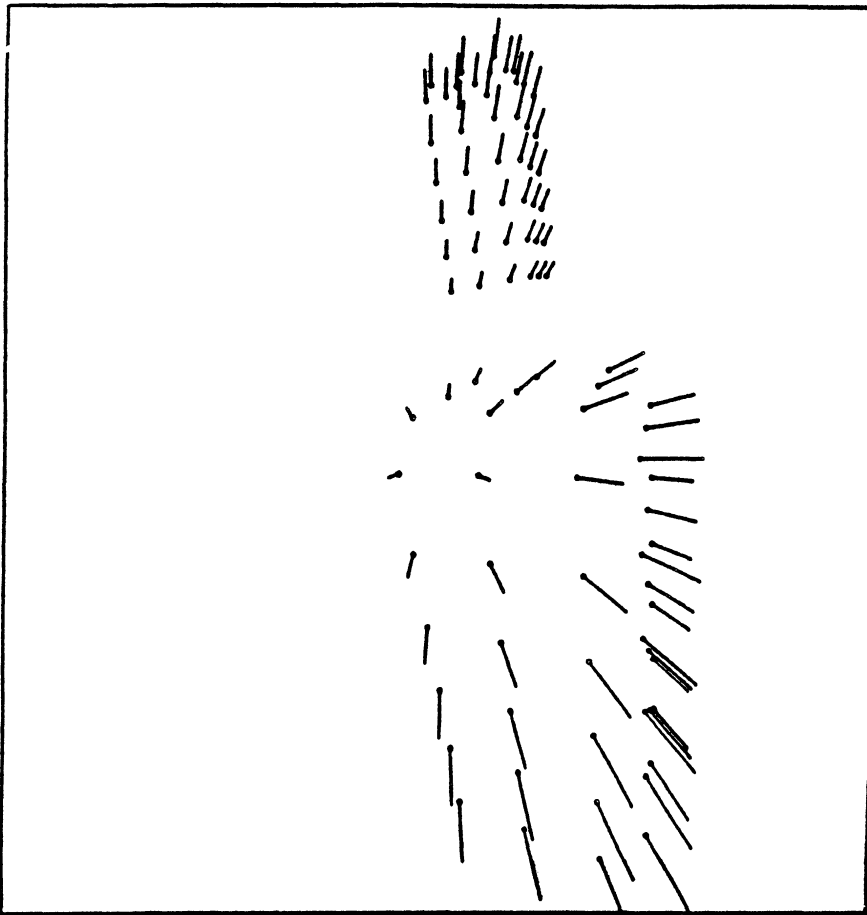


Figure 1 Flow vectors for z translation

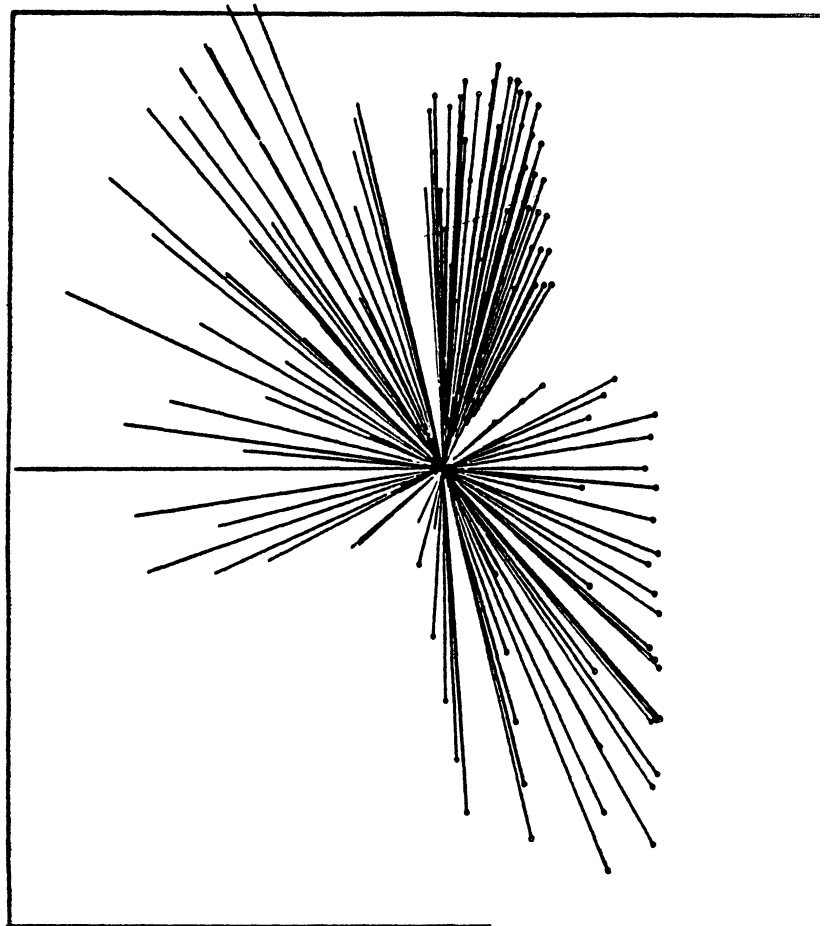


Figure 2 Line extending flow vectors to meet at fo

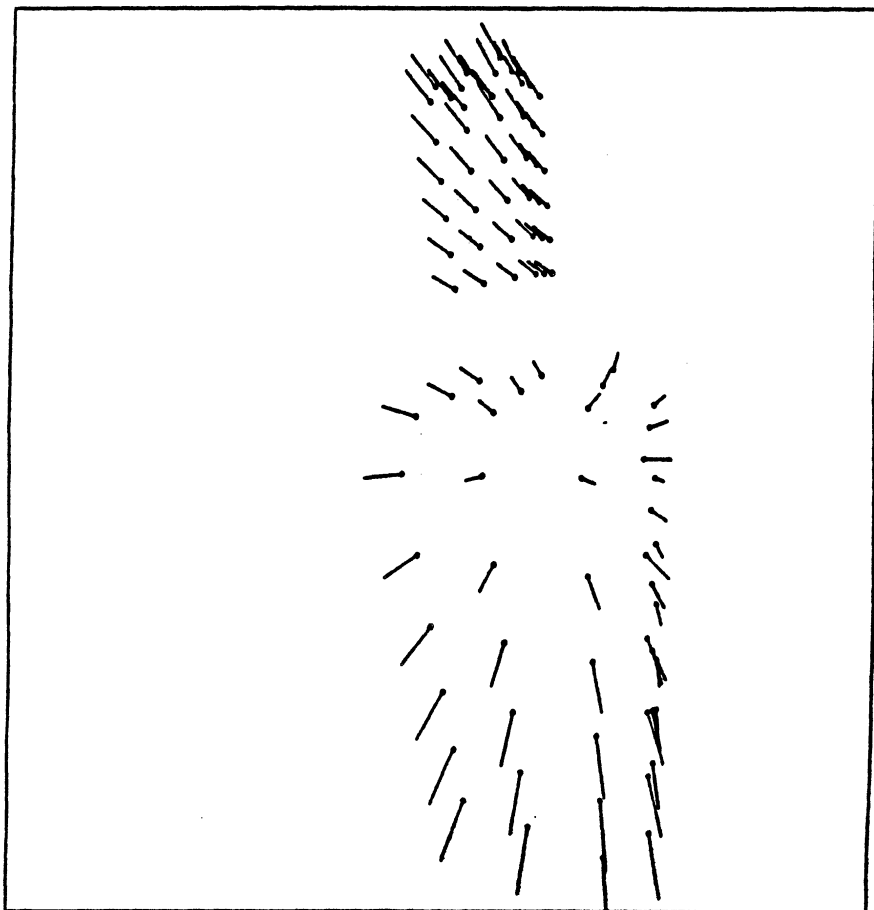


Figure 3 Flow vectors showing y -rotation and translation

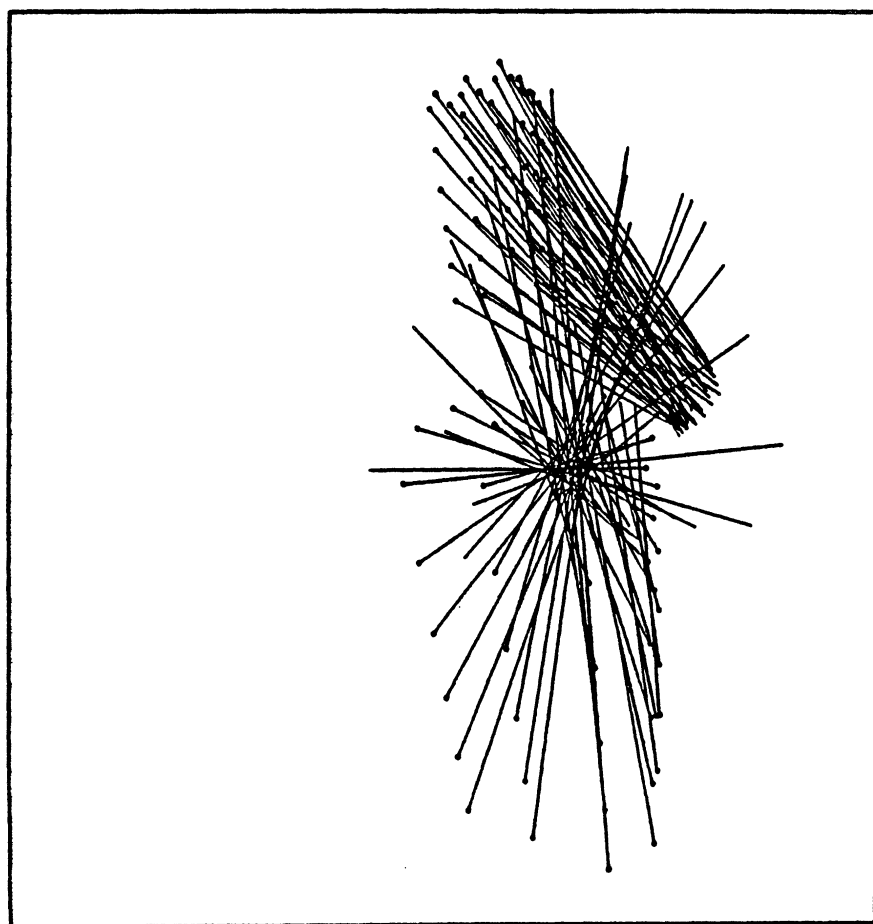


Figure 4 Lines extending flow vectors for y -rotation and translation

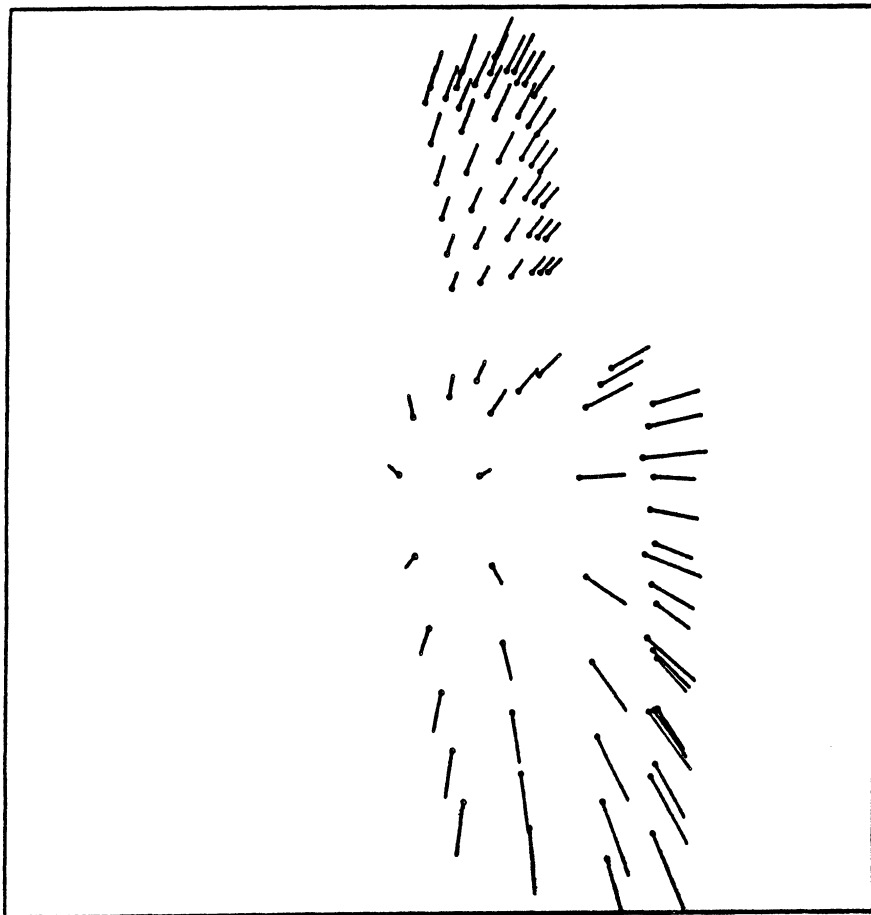


Figure 5 Flow vectors for z -rotation and x and z translation

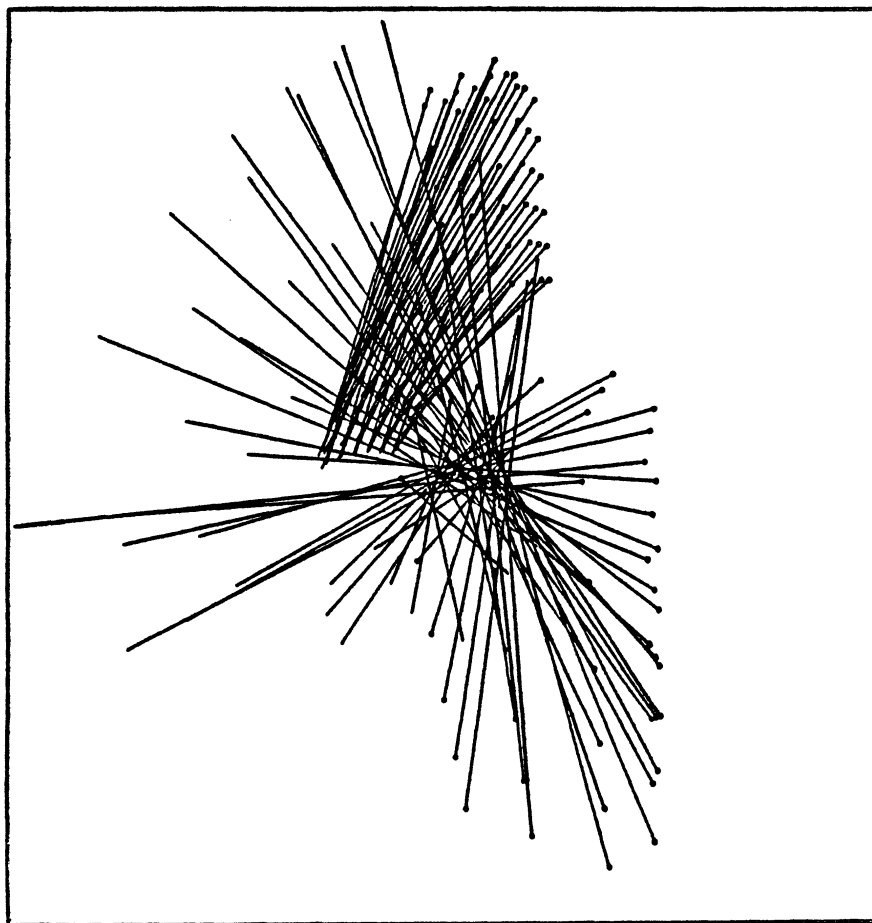


Figure 6 Lines extending z -rotated flow vectors showing circle pattern

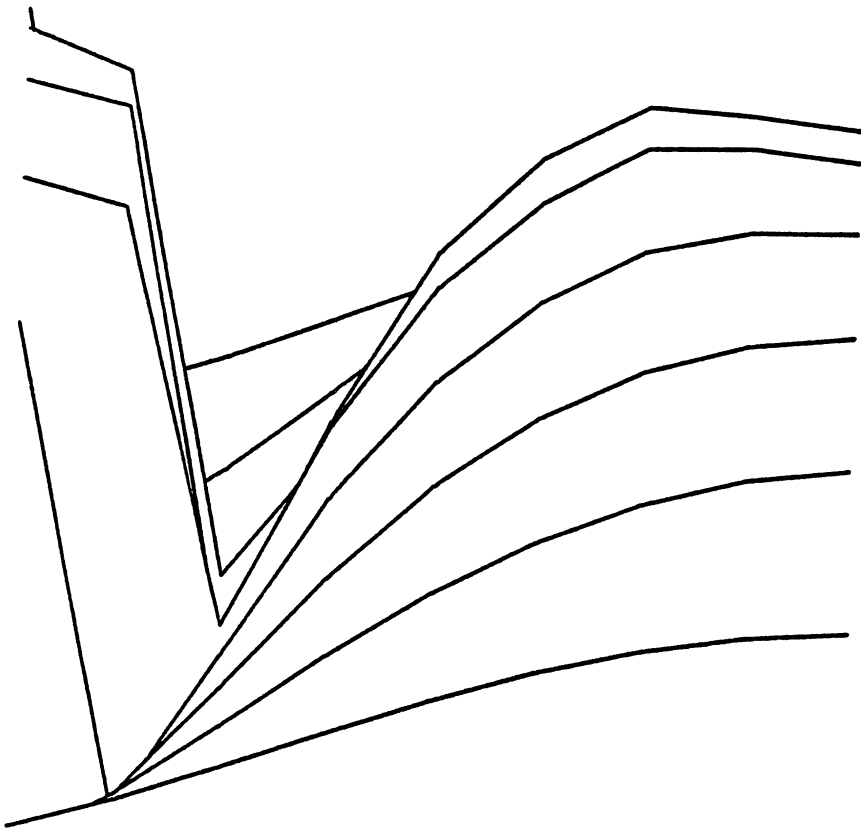


Figure 7 Foe function for foe at 0, -1

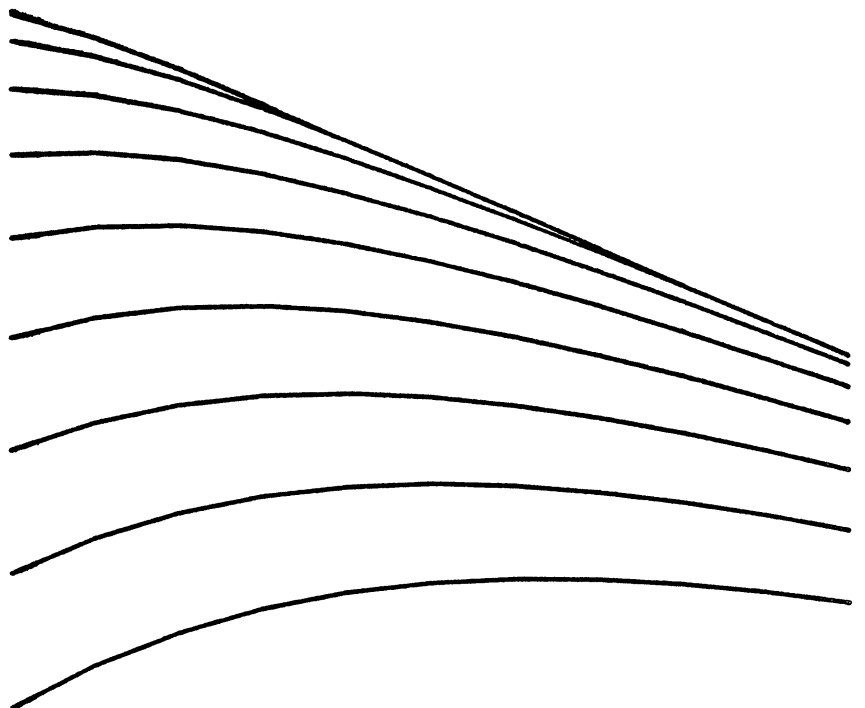


Figure 8 Close-up of foe function at 0, -1

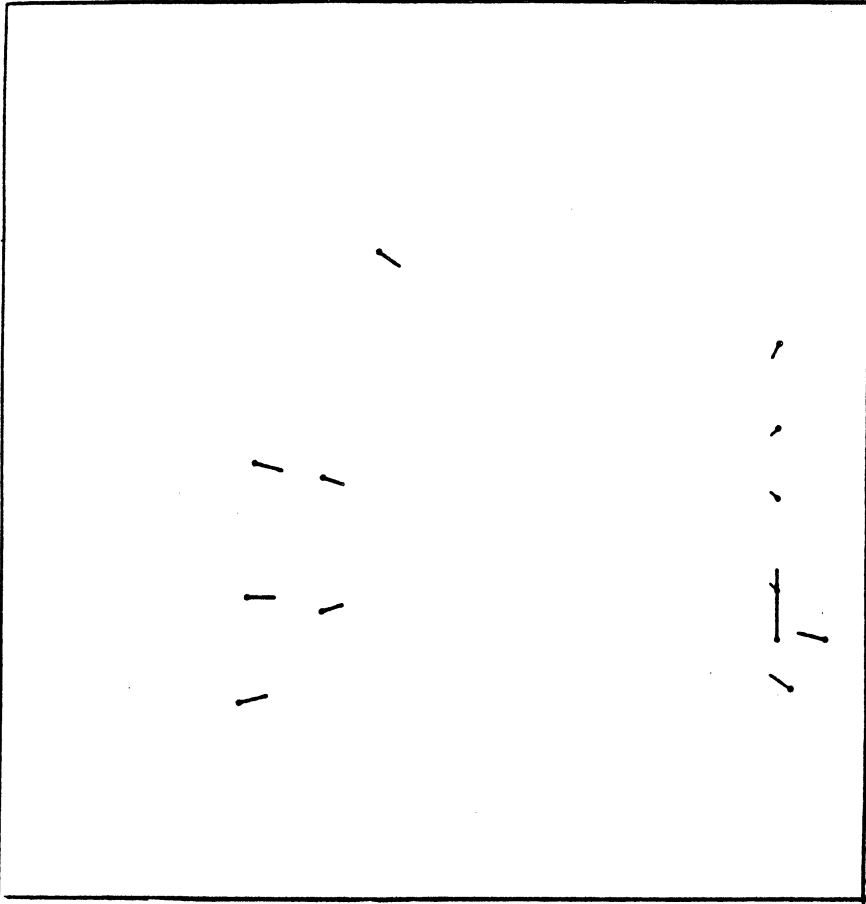


Figure 9 Flow vectors of cube scene

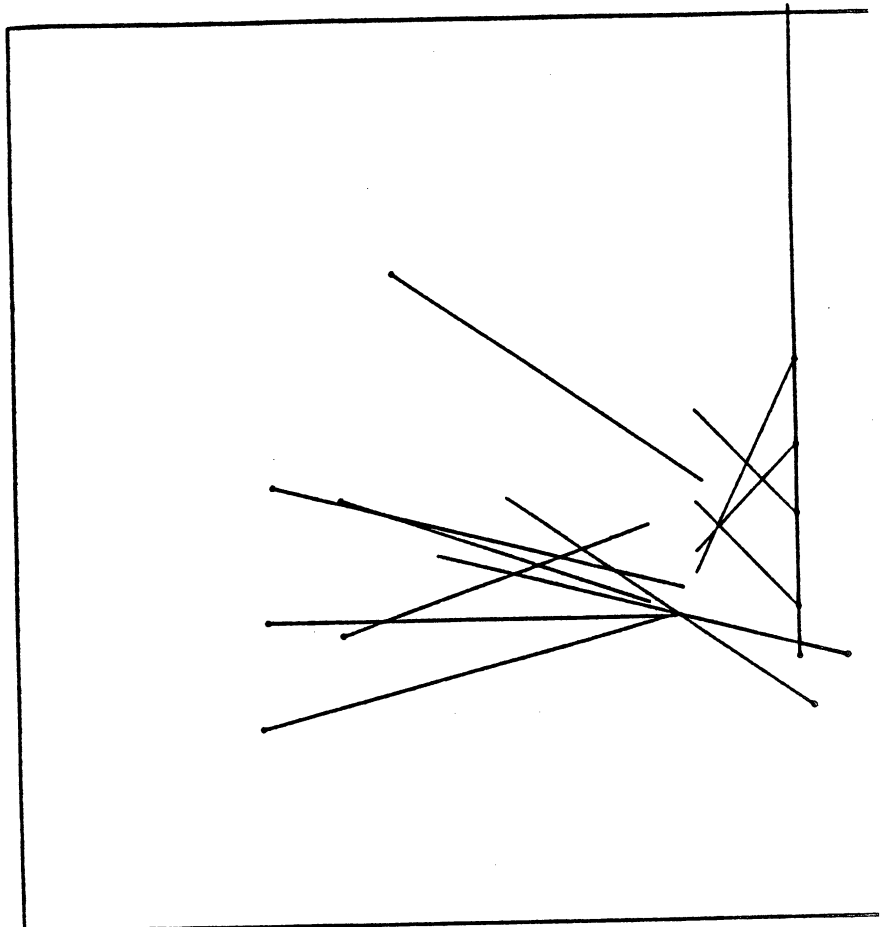


Figure 10 Lines extending flow vectors of cube scene