

Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects

GILAD ADIV

Abstract—A new approach for the interpretation of optical flow fields is presented. The flow field, which can be produced by a sensor moving through an environment with several independently moving, rigid objects, is allowed to be sparse, noisy, and partially incorrect. The approach is based on two main stages. In the first stage, the flow field is partitioned into connected segments of flow vectors, where each segment is consistent with a rigid motion of a roughly planar surface. In the second stage, segments are grouped under the hypothesis that they are induced by a single, rigidly moving object. Each hypothesis is tested by searching for three-dimensional (3-D) motion parameters which are compatible with all the segments in the corresponding group. Once the motion parameters are recovered, the relative environmental depth can be estimated as well. Experiments based on real and simulated data are presented.

Index Terms—Computer vision, dynamic scene analysis, generalized Hough transform, multiple moving objects, multiresolution search, optical flow, recovering three-dimensional motion parameters, segmentation.

I. INTRODUCTION

DYNAMIC visual information can be produced by a sensor moving through the environment and/or by independently moving objects in the visual field. The interpretation of such information consists of forming object hypotheses, recovering the motion parameters of the sensor and each moving object, and structure determination. The results of this interpretation can be used to control behavior, as in robotics or navigation. They can also be integrated, as an additional knowledge source, into an image understanding system such as the VISIONS system [12].

The most common approach for the analysis of visual motion is based on two phases: computation of an optical flow field and interpretation of this field. In the present discussion, the term “optical flow field” refers to both a “velocity field,” composed of vectors describing the instantaneous velocity of image elements, and a “displacement field,” composed of vectors representing the displacement of image elements from one frame to the next. In the latter case, we will assume small values of motion parameters.

The second phase, i.e., the interpretation of the optical flow field, is the main concern of this paper. The information in only one flow field, as opposed to a time sequence of such

Manuscript received June 18, 1984; revised February 22, 1985. Recommended for acceptance by W. E. L. Grimson. This work was supported by the Defense Advanced Research Projects Agency under Grant N00014-82-K-0464.

The author is with the Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.

fields, is utilized. Hence, this work is relevant also to stereoscopic vision, when the relative position and orientation of each camera are not accurately known.

A major problem which has emerged in research on optical flow interpretation is sensitivity to noise. Flow fields generated by existing techniques are noisy and partially incorrect, especially near occlusion or motion boundaries (see the discussion in [29]). Many of the algorithms for interpreting these fields fail under such conditions. Global approaches, which utilize all the available information, can be expected to be relatively robust. Still, an inadequate choice of an optimization criterion often limits the performance of these techniques. Furthermore, the presence of independently moving objects usually makes such global techniques impractical.

These two issues, the presence of noise and the presence of independently moving objects, are addressed in this paper. A new scheme, based on two main stages, is proposed. In the first stage, the flow field, which is allowed to be either dense or sparse, is partitioned into connected segments of flow vectors where each segment is consistent with a rigid motion of a roughly planar surface and, therefore, is likely to be associated with only one rigid object. In the second stage, segments are grouped under the hypothesis that they are induced by a single rigidly moving object. Each hypothesis is tested by searching for 3-D motion parameters which are compatible with all the segments in the corresponding group. Once the motion parameters are recovered, the relative environmental depth can be estimated as well.

This technique of segmenting the flow field and then combining segments to form object hypotheses, makes it possible to deal with independently moving objects while employing all the available information associated with each object. In addition, the search for 3-D motion parameters is based on a least-squares technique which minimizes the deviation between the given flow field and that predicted from the computed parameters. Thus, the proposed scheme is relatively insensitive to noise. There are, however, inherent ambiguities in the interpretation of noisy flow fields. These ambiguities, which are only discussed briefly in this paper, are analyzed and demonstrated in [4].

In the next section, techniques existing in the literature for visual motion interpretation are examined. The mathematical formulation of the model and the task is presented in Section III. In subsequent sections, algorithms for flow field segmentation, estimation of motion parameters, and structure deter-

mination are developed. Experiments based on real and simulated data are described in Section VI.

II. LITERATURE REVIEW

In this section, we examine existing methods for interpreting optical flow fields. Most of these methods rely on the information contained in one flow field and are restricted to rigid motion. In addition, it is usually assumed that the scene contains only one object or, equivalently, that the sensor is moving but the environment is stationary (e.g., [8], [14], [27]). Only a few researchers, e.g., [20], [28], explicitly consider scenes with several rigidly moving objects, and investigate techniques for decomposing the flow field into sets corresponding to these objects, while simultaneously recovering the 3-D motion and structure associated with each of them.

Many of the algorithms described in the literature for interpreting flow fields cannot successfully deal with a realistic level of noise. A few algorithms are less sensitive and may work reasonably well in restricted real-world situations. This issue, as well as the adequacy of the various techniques in the presence of independently moving objects, will be emphasized in the review. Let us now start with a discussion of techniques which assume only one rigid motion.

Several researchers [9], [10], [18], [19], [22] present sets of nonlinear equations with motion parameters as unknowns. Methods for solving such equations are usually iterative and require good initial guesses of the unknowns. Sensitivity to noise is indicated by experiments reported in [9], [10], and [22]. It is shown there that the results can be improved by using a large number of flow vectors and by increasing the size of the region containing these vectors. Therefore, assuming no independently moving objects, the best results can be achieved when these algorithms are applied to the whole image.

Longuet-Higgins [17] and Tsai and Huang [27] develop computationally simple techniques based on solving a set of linear equations. Furthermore, conditions for the uniqueness of the solutions are formulated. However, difficulties in the presence of noise are reported again [27].

The sensitivity to noise of the algorithms mentioned above may be partly due to inadequate choices of an optimization criterion. A more appropriate approach is seemingly taken by Roach and Aggarwal [25] and Bruss and Horn [8]. A least-squares criterion is employed in order to minimize the deviation between the measured data and the corresponding values predicted from the computed 3-D motion and structure. This approach leads to a system of nonlinear equations from which the motion parameters and the depth values can be numerically computed. Difficulties with noisy data, which are still reported in [25], may point out that in certain situations recovering 3-D information from flow fields is inherently unstable.

Assuming a purely translational motion, all the flow vectors are oriented toward or from a single point in the image plane. Determining this point, called the focus of expansion (FOE), yields the direction of the translation. A few techniques, reviewed below, are based on this observation.

Early results based on real images are reported in [33]. However, only sensor motion restricted to translation is allowed, and the environment is assumed to contain only planar surfaces

at one of two given orientations. Thus, the algorithm can be based on a search for the FOE and the distances to the surfaces in the scene. Lawton [14], [15] describes a robust algorithm which has been applied to real-world images from several different task domains. This algorithm requires no restrictions on the shape of the environment, but is still restricted to translation. It is based on a global sampling of an error measure corresponding to the potential positions of the FOE, followed by a local search to determine the exact location of the minimum value.

Prazdny [23] describes a method which relies on decomposition of the velocity field into rotational and translational components. For a hypothesized rotational component, the FOE of the corresponding translational field and a related error measure are computed. Thus, an error function of the three rotation parameters is obtained and the solution can be determined by minimizing this function. Jerian and Jain [13] report on difficulties with applying a similar approach to noisy data.

Rieger and Lawton [24] develop a relatively robust and simple technique for computing the motion parameters of a camera moving in a stationary environment with significant depth discontinuities. The algorithm is based on the observation that the differences between vectors at the corresponding discontinuities in the flow field are oriented toward or from the FOE of the translational field [16]. This technique depends on the ability to compute reasonably accurate values of the flow vectors near occlusion boundaries.

A number of methods presented in the literature is based on a local analysis of the flow field. Ullman [28] allows the presence of independently moving objects and examines small sets of adjacent vectors. If there exists a unique, rigid interpretation consistent with all the vectors in a given set, then this interpretation is assumed to be correct and the vectors in the set are grouped together. Because of its local nature, this approach seems to be very sensitive to errors in the flow field.

Longuet-Higgins and Prazdny [16] and Waxman *et al.* [30]-[32] introduce equations for computing the motion parameters and the local structure at a given point in the environment from the flow field and its first and second spatial derivatives at the corresponding point in the image. If the scene consists of several objects in relative motion, then a separate computation can be carried out on each one. Local estimates of the second derivatives of the optic flow seem to be inaccurate in the presence of noise, and no algorithm has been presented for reliably computing such derivatives. Good results when determining 3-D motion and structure of planar surfaces are reported, however, in [32].

More global approaches, which still take into account independently moving objects, are proposed by a few researchers. Neumann [20] proposes an elegant hypothesize-and-test scheme: for any rotation hypothesis, the translation component may be decomposed such that motion compatibility of many flow vectors can be easily tested. This technique relies heavily on the assumption of orthographic projection.

The generalized Hough technique [6] is another global approach for recovering motion parameters from a given flow field [1], [2], [7]. This technique is relatively insensitive to

noise and partially incorrect data. It can also be applied, using a multipass approach [1], [2], [11], in scenes containing independently moving objects. However, the Hough technique may be very expensive, since high dimensionality and fine resolution in the parameter space require large amounts of memory and computation time. Therefore, in [7] the depth information is assumed to be known, thus making the task much easier, and in [1] and [2] the approach is demonstrated only for two-dimensional (2-D) motion with four parameters.

This review demonstrates typical restrictions and difficulties of algorithms reported in the literature for interpretation of optical flow fields. Most of these techniques are sensitive to noise and are difficult to apply in the presence of independently moving objects, unless severe restrictions are assumed. In some situations, there exists inherent ambiguity in recovering 3-D information from noisy flow fields. Yet, in order to improve the performance as much as possible, it is necessary to efficiently utilize all the available information and to employ an adequate optimization criterion. These obvious conclusions constitute the basis of the approach developed in this paper.

III. THE MODEL AND THE TASK—A MATHEMATICAL FORMULATION

A. Basic Model and Equations

In this section, we present a notation for describing the motion of a camera through an environment containing independently moving objects. We also review the equations describing the relation between the 3-D motion and structure and the corresponding optical flow, assuming a perspective projection. The equations are developed both for velocity fields and displacement fields.

Let (X, Y, Z) represent a Cartesian coordinate system which is fixed with respect to the camera (see Fig. 1), and let (x, y) represent a corresponding coordinate system of a planar image. The focal length, from the nodal point O to the image, is assumed to be known. It can be normalized to 1 without loss of generality. Thus, the perspective projection (x, y) on the image of a point (X, Y, Z) in the environment is

$$x = X/Z \quad (3.1a)$$

$$y = Y/Z. \quad (3.1b)$$

The motion, relative to the camera, of a rigid object in the scene can be decomposed into two components: translation $\mathbf{T} = (T_X, T_Y, T_Z)$ and rotation $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$. In the equations corresponding to velocity fields, these symbols represent instantaneous spatial velocities and, in the equations corresponding to displacement fields, they represent differences in position and orientation between two time instances.

In the velocity-based scheme, if (X, Y, Z) are the instantaneous camera coordinates of a point on the object, then the corresponding projection (x, y) on the image moves with a velocity (α, β) , where [16]

$$\alpha = -\Omega_X xy + \Omega_Y(1 + x^2) - \Omega_Z y + (T_X - T_Z x)/Z \quad (3.2a)$$

and

$$\beta = -\Omega_X(1 + y^2) + \Omega_Y xy + \Omega_Z x + (T_Y - T_Z y)/Z. \quad (3.2b)$$

Notice that (α, β) can be represented as the sum

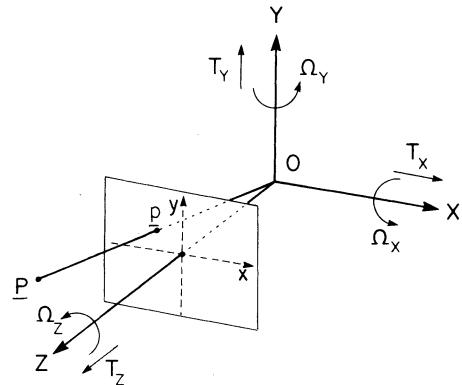


Fig. 1. (Redrawn from [16].) A coordinate system (X, Y, Z) attached to the camera, and the corresponding image coordinates (x, y) . The image position p is the perspective projection of the point P in the environment. $\mathbf{T} = (T_X, T_Y, T_Z)$ and $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ represent the relative translation and rotation of a given object in the scene.

$$(\alpha, \beta) = (\alpha_R, \beta_R) + (\alpha_T, \beta_T) \quad (3.3)$$

where (α_R, β_R) and (α_T, β_T) are, respectively, the rotational and translational components of the velocity field

$$\alpha_R = -\Omega_X xy + \Omega_Y(1 + x^2) - \Omega_Z y, \quad \alpha_T = (T_X - T_Z x)/Z \quad (3.4a,b)$$

$$\beta_R = -\Omega_X(1 + y^2) + \Omega_Y xy + \Omega_Z x, \quad \beta_T = (T_Y - T_Z y)/Z. \quad (3.4c,d)$$

In the displacement-based scheme, let (X, Y, Z) be the camera coordinates at time t of a point on the object and let (X', Y', Z') be the corresponding coordinates at time t' . If the rotation $\boldsymbol{\Omega}$ is followed by the translation \mathbf{T} , then

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{T} \quad (3.5)$$

where the rotation matrix R can be approximated, assuming small values of the rotation parameters, by (see [9])

$$R = \begin{pmatrix} 1 & -\Omega_Z & \Omega_Y \\ \Omega_Z & 1 & -\Omega_X \\ -\Omega_Y & \Omega_X & 1 \end{pmatrix}. \quad (3.6)$$

If (x, y) and (x', y') are the image coordinates corresponding to the points (X, Y, Z) and (X', Y', Z') , respectively, then

$$x' = \frac{X'}{Z'} = \frac{x - \Omega_Z y + \Omega_Y + T_X/Z}{-\Omega_Y x + \Omega_X y + 1 + T_Z/Z} \quad (3.7a)$$

and

$$y' = \frac{Y'}{Z'} = \frac{\Omega_Z x + y - \Omega_X + T_Y/Z}{-\Omega_Y x + \Omega_X y + 1 + T_Z/Z}. \quad (3.7b)$$

Now, let (α, β) be, in this case, the displacement vector $(x' - x, y' - y)$. Then from (3.7) we get

$$\alpha = \frac{-\Omega_X xy + \Omega_Y(1 + x^2) - \Omega_Z y + (T_X - T_Z x)/Z}{1 + \Omega_X y - \Omega_Y x + T_Z/Z} \quad (3.8a)$$

and

$$\beta = \frac{-\Omega_X(1+y^2) + \Omega_Yxy + \Omega_Zx + (T_Y - T_Zy)/Z}{1 + \Omega_Xy - \Omega_Yx + T_Z/Z}. \quad (3.8b)$$

If $|T_Z/Z| \ll 1$ and the field of view of the camera, i.e., the visual angle corresponding to the whole image, is not very large, then (employing also the assumption that the rotation parameters are small) we can approximate the displacement vector (α, β) by (3.2).

To conclude, (3.2a) and (3.2b) hold not only for velocity fields, but also for displacement fields, given that the following conditions are satisfied: 1) the rotation parameters are small, 2) the Z component of the translation is small relative to the distance of the object from the camera, and 3) the field of view is not very large. The conditions 1) and 2) are reasonable if the time interval between the two image frames is short enough or if the motion is slow. In the following sections, we restrict ourselves to conditions which allow us to employ (3.2a) and (3.2b) as the basis of our analysis.

B. The Task—Inputs and Outputs

The input utilized by our scheme for interpreting motion information is a flow field described by $\{(\alpha(x, y), \beta(x, y), W(x, y))\}$, where $(\alpha(x, y), \beta(x, y))$ is the flow vector at the (x, y) pixel in the image, and $W(x, y)$ is a corresponding weight between 0 and 1. High reliability of the flow vector is represented by a weight close to 1, and low reliability by a weight close to 0. The flow field can be either dense, thus defined at most of the pixels, or sparse, thus defined only on a sparse subset of the image pixels. If the flow field is undefined at a pixel (x, y) , then $W(x, y)$ is determined to be 0. A rough estimate of the noise level in the flow field is assumed to be known.

The interpretation process should result in three outputs: object masks, motion parameters, and depth. We wish to partition the set $\{(x, y) : W(x, y) > 0\}$ into disjoint sets of pixels where each set corresponds to a different rigid object. The pixels corresponding to the stationary environment, where the optical flow is induced only by the camera motion, should be grouped together.

The five recoverable motion parameters of each rigid object, relative to the camera, should be estimated. These parameters include the rotation parameters $(\Omega_X, \Omega_Y, \Omega_Z)$ and the direction of the translation vector defined by the unit vector $\mathbf{U} = \mathbf{T}/r$, where r is the length of the translation vector \mathbf{T} . Once the motion parameters are recovered, it is also possible to estimate the relative depth $Z(x, y)/r$ corresponding to each pixel (x, y) where a flow vector is defined, unless $r = 0$ or the location of the vector is exactly in the FOE.

IV. SEGMENTATION

In this section, we develop a method for segmenting the flow field into connected sets of flow vectors where each set is consistent with a rigid motion of a roughly planar patch. A segment satisfying this constraint is very likely to be associated with only one rigid object. Thus, the data are organized into coherent units which constitute the basis for formation of object hypotheses in the second stage. This organization makes it possible to deal with independently moving objects, while employing all the information associated with each object and

preventing the suppression of valuable data in distinct, but possibly small, surfaces. Another purpose of the segmentation is exclusion of incorrect flow vectors which are inconsistent with their neighbors.

A. Ψ Transformations—A Segmentation Constraint

In order to achieve a useful segmentation, we employ a few simple observations on the structure of optical flow fields. First, we examine the flow field induced by a rigid motion of a planar surface. Excluding the degenerate case in which the same plane contains both the surface and the nodal point (and, therefore, the corresponding region in the image is a straight line), the surface can be represented by the equation

$$k_X X + k_Y Y + k_Z Z = 1. \quad (4.1)$$

The coefficients k_X , k_Y , and k_Z can be any real numbers, except the case in which all of them are zero. Using (3.1), we obtain

$$1/Z = k_X x + k_Y y + k_Z. \quad (4.2)$$

Substituting (4.2) in (3.2), we realize that, given a relative motion $\{\mathbf{T}, \boldsymbol{\Omega}\}$, the flow field is

$$\alpha = a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \quad (4.3a)$$

$$\beta = a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \quad (4.3b)$$

where

$$a_1 = \Omega_Y + k_Z T_X \quad (4.4a)$$

$$a_2 = k_X T_X - k_Z T_Z \quad (4.4b)$$

$$a_3 = -\Omega_Z + k_Y T_X \quad (4.4c)$$

$$a_4 = -\Omega_X + k_Z T_Y \quad (4.4d)$$

$$a_5 = \Omega_Z + k_X T_Y \quad (4.4e)$$

$$a_6 = k_Y T_Y - k_Z T_Z \quad (4.4f)$$

$$a_7 = \Omega_Y - k_X T_Z \quad (4.4g)$$

and

$$a_8 = -\Omega_X - k_Y T_Z. \quad (4.4h)$$

Equations (4.3) correspond to what we shall call a Ψ transformation. They describe a 2-D motion in the image plane, represented by the eight parameters a_1, \dots, a_8 . Note that a similar representation of the optical flow produced by a moving planar surface is introduced in [30].

We proceed now with another observation, related to arbitrary surfaces in the environment. Given such a surface, it can be described as a function $Z = Z(x, y)$ defined on the image region R which corresponds to the projection of this surface. Let $Z' = Z'(x, y)$ be an approximation to the surface Z such that

$$|\Delta Z(x, y)| \stackrel{\text{def}}{=} |Z(x, y) - Z'(x, y)| \ll Z(x, y) \quad \text{for any } (x, y) \in R. \quad (4.5)$$

If (α_T, β_T) and (α'_T, β'_T) are the translational components of the flow fields induced by the same motion of the surfaces Z and Z' , respectively, then

$$\begin{aligned}\alpha'_T &= \frac{T_X - T_Z x}{Z'} = \frac{T_X - T_Z x}{Z - \Delta Z} \approx \frac{T_X - T_Z x}{Z} \left(1 + \frac{\Delta Z}{Z}\right) \\ &= \alpha_T \left(1 + \frac{\Delta Z}{Z}\right)\end{aligned}\quad (4.6a)$$

and

$$\begin{aligned}\beta'_T &= \frac{T_Y - T_Z y}{Z'} = \frac{T_Y - T_Z y}{Z - \Delta Z} \approx \frac{T_Y - T_Z y}{Z} \left(1 + \frac{\Delta Z}{Z}\right) \\ &= \beta_T \left(1 + \frac{\Delta Z}{Z}\right).\end{aligned}\quad (4.6b)$$

The rotational component of the flow field is independent of the structure of the environment. Hence, given (4.5), the flow field induced by the approximating surface Z' is very similar to the real flow in the region R . As a conclusion, if Z' is a planar surface which satisfies (4.5), then the flow field in R can be approximated by a Ψ transformation.

In a real-world environment, the surface can usually be approximated by a piecewise planar surface, containing only a few planar patches, for which the distance between the real surface and the approximating one is small relative to the distance from the sensor to the surface. If this is the case, then the flow field can be approximated, reasonably well, by a piecewise Ψ transformation. This suggests that a useful segmentation of the flow field can be based on finding connected sets of flow vectors (for a rigorous definition of connectivity in flow fields, see [4]), where the vectors in each set approximately satisfy the same Ψ transformation. Thus, each segment is consistent with a rigid motion of a roughly planar surface and can be assumed to be induced by the relative motion of only one rigid object.

Notice that different segmentation constraints could have been employed. For example, it is possible to use a stronger constraint based on consistency with affine transformations [see (4.7)]. However, in such a case, an oversegmentation is possible, as demonstrated by the results of the first step of the segmentation algorithm in experiment 1 [Fig. 3(b)]. Another option is to use a weaker constraint based, for example, on a transformation which includes all of the 12 coefficients of the second-order Taylor expansion of the flow field (a representation of the optical flow, based on these coefficients, is proposed in [30] and [31]). In this case, however, the space of all the possible transformations strictly contains the space of all the Ψ transformations; therefore, the probability of grouping together flow vectors induced by different rigid motions is increased. In the next section, we describe an algorithm for achieving a segmentation based on the Ψ transformation.

B. Segmentation Algorithm

The generalized Hough transform technique [6] is a useful tool for grouping together flow vectors which satisfy the same 2-D parameterized transformation [1], [2]. In this technique, the set of relevant transformations is represented by a discrete multidimensional parameter space, where each dimension corresponds to one of the transformation parameters. Each point in this space uniquely characterizes a transformation, defined by the corresponding parameter values. A flow vector "votes" for each point with an associated transformation consistent

with this vector. The points receiving the most votes are likely to represent transformations corresponding to large segments in the flow field.

As a global technique, the Hough transform is relatively insensitive to noise and partially incorrect or occluded data. However, high dimensionality of the parameter space requires large amounts of memory and computation time. In our case, the segmentation constraint is based on the eight-parameter Ψ transformations (4.3). The Hough technique can, in principle, be employed, but the computational cost required for such a number of parameters is very high. Therefore, a three-stage algorithm is proposed.

The first stage is based on grouping together adjacent flow vectors into *components* consistent with *affine transformations*. The affine transformations, represented by

$$\alpha = a_1 + a_2 x + a_3 y \quad (4.7a)$$

and

$$\beta = a_4 + a_5 x + a_6 y \quad (4.7b)$$

are subclasses of the Ψ transformations, parameterized by only six parameters. Furthermore, these parameters can be partitioned into two disjoint sets of three parameters each, corresponding to (4.7a) and (4.7b). Thus, the grouping problem in the first stage can be basically solved by applying the Hough technique to 3-D parameter spaces, as will be shown in Section IV-B-1.

In the second stage, components which are consistent with the same Ψ transformation are merged into segments. Given a set of adjacent components, optimal parameters are computed using the least-squares technique. Related error measures, associated with each component in the set, can be thus obtained. If these error values are not high (in a sense defined in Section IV-B-2), then the components are merged. Note that this hypothesize-and-test technique could not be employed without the initial grouping into components, which drastically reduces the number of hypotheses to be tested.

Sometimes, an overfragmentation may occur in the first stage of the segmentation, that is, a segment is partitioned into a large number of small components. This is demonstrated in experiment 1 in Section VI [see Fig. 3(b)], where the flow field in one of the segments contains large second-order terms. In order to reduce the computational cost of the first and second segmentation stages, the grouping of vectors belonging to small connected sets may be postponed, in such a case, to the third stage. In this stage, flow vectors which are not contained in any of the segments are merged into neighboring segments, if they are consistent with the corresponding Ψ transformations. If, after the third stage, some of these small sets are still not merged into the existing segments, then the first and second stages of the segmentation may be repeated, focused only on these sets, thus possibly creating new segments.

In the following subsections, the three stages of the segmentation are more fully described. Details, which for the sake of brevity are suppressed, can be found in [4].

1) First Stage—Grouping Based on Affine Transformations:

In the first stage of the segmentation, flow vectors are grouped into components, where a component is a connected set of

vectors which support the same affine transformation. The grouping is based on a multipass Hough approach [1], [2], [11] where, in each iteration, a modification of the generalized Hough technique is employed. This modified Hough technique will be presented in the next subsection, while the details of the multipass scheme will be described in Section IV-B-1-b.

a) A Modified Version of the Generalized Hough Technique: The affine transformations can be represented by a six-dimensional (6-D) parameter space where each dimension corresponds to one of the parameters a_1, \dots, a_6 in (4.7). For computational reasons, the parameter space must contain only a finite number of points. Therefore, minimal and maximal values are determined for each parameter and the corresponding interval is quantized. The parameter space is the Cartesian product of the discrete sets so obtained.

A flow vector $(\alpha(x, y), \beta(x, y))$ votes for a transformation (a_1, \dots, a_6) , if it approximately satisfies the constraint equations (4.7a, b), that is, if

$$\delta \stackrel{\text{def}}{=} \sqrt{\delta_x^2 + \delta_y^2} \leq \epsilon \quad (4.8a)$$

where

$$\delta_x = |\alpha - a_1 - a_2 x - a_3 y| \quad (4.8b)$$

and

$$\delta_y = |\beta - a_4 - a_5 x - a_6 y|. \quad (4.8c)$$

Note that ϵ is a function of the resolution in the parameter space and the noise level in the flow field, but to compensate for the fact that the image transformations cannot be expected to be exactly affine, it is never less than a given threshold, typically one pixel. The amount of support is determined by the function

$$V(a_1, a_2, a_3, a_4, a_5, a_6, x, y) = 1 - 0.758/\epsilon \quad (4.9)$$

which allows the support to range from 1 down to 0.25 for those flow vectors at the limit of the acceptable error range. The total amount of support given to each transformation (a_1, \dots, a_6) is the weighted sum

$$S(a_1, a_2, a_3, a_4, a_5, a_6) = \sum_{x, y} W(x, y) V(a_1, a_2, a_3, a_4, a_5, a_6, x, y) \quad (4.10)$$

where $W(x, y)$ is the weight of the flow vector at the pixel (x, y) .

Suppose now that we want to find the affine transformation, among those represented in the parameter space, which is maximally supported by a given set of flow vectors. Basically, we have to compute the support, according to (4.10), given to any of those transformations. A serious computational problem may arise, however, if the number of points in the parameter space is very high. For example, in the experiments described in Section VI, the minimal and maximal possible values of the parameters a_1 and a_4 are taken to be -64 pixels and +64 pixels, respectively, and the desired accuracy is 0.25 pixel. Thus, 512 samples are apparently needed for each of these parameters. Using an equal number of samples also for the other parameters, the parameter space should contain $512^6 \approx 16 \times 10^{15}$ points, and a straightforward Hough technique is computationally impractical.

This problem is alleviated by using two techniques. First, a multiresolution scheme in the parameter space is employed. The Hough technique is iteratively used, where in each iteration the parameter space is quantized around the values estimated in the previous iteration, using a finer resolution. Thus, utilizing a limited memory size, accurate parameter values can still be found. Other methods for achieving this goal are presented in [21] and [26].

The second technique is based on a decomposition of the parameter set into two disjoint subsets $\{a_1, a_2, a_3\}$ and $\{a_4, a_5, a_6\}$. The Hough technique is separately applied to the corresponding 3-D parameter spaces, using the relevant constraint (4.7a), or (4.7b). Sets of highly supported parameter triples $A_\alpha = \{(a_{1i}, a_{2i}, a_{3i}) : i = 1, \dots, N\}$ and $A_\beta = \{(a_{4i}, a_{5i}, a_{6i}) : i = 1, \dots, N\}$ are thus found, where N was experimentally determined to be 10. As a result, a set of N^2 hypothesized affine transformations

$$A_{\alpha\beta} = A_\alpha \times A_\beta = \{(a_{1i}, a_{2i}, a_{3i}, a_{4j}, a_{5j}, a_{6j}) : i = 1, \dots, N; j = 1, \dots, N\} \quad (4.11)$$

is obtained. The support function can then be directly applied to the set $A_{\alpha\beta}$, thus determining the maximally supported transformation T^* in this set. T^* is not necessarily the maximally supported transformation in the 6-D parameter space. However, large components in the flow field, corresponding to maxima points in the 6-D space, can be expected to produce maxima points also in each of the 3-D parameter spaces. Therefore, T^* is, at least, a near optimal transformation, as can also be concluded from the experimental results. The decomposition technique is employed in each iteration of the multiresolution scheme; together they create a very efficient algorithm.

b) Implementation of a Multipass Approach: The multipass Hough technique is an iterative procedure, where in each iteration the motivating goal is to find the largest component among the flow vectors which do not belong to any of the components created in previous iterations. More accurately, we consider the set of vectors which are assigned a positive weight, are not included yet in any component, and were not excluded from further consideration. (The circumstances under which vectors are excluded from further consideration are described below.) In this set, we first find the largest connected subset S . (The term *largest subset* refers here, as well as elsewhere in this section, to the subset with the maximal sum of weights.) An affine transformation, highly supported by the vectors in S , is then computed using the Hough technique, and a corresponding component C is determined. If the sum of weights in $C \cap S$ is higher than a given threshold, then C is added to the list of components and a new iteration begins. Otherwise, another affine transformation, which is also highly supported by vectors in S , is selected and a related component is found and checked. If after a few such trials no component is accepted, then the set S is excluded from further consideration and a new iteration begins.

The sum of weights of vectors in each component is required to exceed a given threshold h , which is related to the noise pattern in the flow field. By selecting a small value of h , com-

ponents corresponding to groups of compatible but errorful flow vectors are more likely to emerge. On the other hand, using a relatively large value of h may prevent detection of small components and, eventually, segments corresponding to distinct surfaces and/or independently moving objects. The selection of h should reflect a compromise between these two types of risk, and it heavily depends on the type of algorithm used for computing the flow field. If, for example, high and correlated values of noise are expected to be occasionally present in groups of adjacent flow vectors, then a selection of a large value of h may be appropriate.

Each iteration of the multipass technique is composed of the following steps.

1) Find the set S defined earlier in this section. If the sum of weights in this set is not above the threshold h , then there is no point in continuing the search for new components, and the merging stage begins. Sometimes an overfragmentation occurs, i.e., a segment is partitioned into a large number of small components. In order to prevent an excessive number of iterations in such a case, a new threshold h' , higher than h , is determined. The process is stopped if the sum of weights in S is below this threshold and the number of components already determined is higher than a given number, typically 10. The grouping of vectors in small sets is thus postponed to the third stage.

2) Partition the set S into a given number (typically 64) of roughly square windows, such that the sum of weights in each window is approximately the same. Then, from each window, select the flow vector with maximal weight. The Hough technique will be applied only to these vectors, and not to the whole set S , in order to reduce the computation time.

3) Use the modified Hough technique, described in Section IV-B-1-a, to find an affine transformation which receives high support from the flow vectors selected in the previous step.

4) For each vector in the flow field, compute the error measure δ (4.8a) associated with the computed affine transformation. If this deviation is below the threshold ϵ (Section IV-B-1-a) and the vector does not belong to any of the previously created components, then include the vector in a new set S' . In addition, even if the vector already belongs to a component, but the new value of the error measure δ is smaller than the old value corresponding to the existing component, then include the vector in S' . If the sum of weights in the set $S \cap S'$ does not exceed the threshold h then, assuming that other affine transformations will not have significantly more support in the set S , exclude S from further consideration and start a new iteration. Otherwise, continue with step 5).

5) Find in the set S' a connected subset C with a maximal sum of weights of vectors from S . If this sum is not above h , then we wish to try a new affine transformation, which may be supported by a larger *connected* subset of vectors (although the total support by vectors in S is expected to be smaller). The goal of selecting a new transformation is achieved by decreasing, only temporarily while executing again steps 2) and 3) in the current iteration, the weights of all the vectors in S which support the current affine transformation (in the experiments they are divided by two and going back to step 2). If after a few such cycles (three, in the experiments) no sufficiently large component is found, then the set S is excluded from further consideration and a new iteration begins.

6) If the sum of weights in the set $C \cap S$ exceeds h , then it is added to the list of components. Note that the new component may contain vectors which previously belonged to other components. These components must be examined to see whether they are still connected sets. If not, then the corresponding component is reduced to its largest connected subcomponent, and the other vectors are among the vectors to be examined in the first step of the next iteration.

2) *Second Stage—Merging of Components:* Components created in the first stage of the segmentation are atomic units which, if consistent with the same Ψ transformation, should be merged together to create a segment. Consistency with a Ψ transformation is detected by computing, using the least-squares technique, optimal parameters and related error values for sets of adjacent components. The merging decision is primarily based on these error values.

a) *Computing an Optimal Ψ Transformation:* Given a set of n flow vectors, we wish to compute, employing the least-squares criterion, the optimal Ψ transformation corresponding to this set. The error function to be minimized is

$$\begin{aligned} E(a_1, \dots, a_8) = & \sum_{i=1}^n W_i [(\alpha_i - a_1 - a_2 x_i - a_3 y_i \\ & - a_7 x_i^2 - a_8 x_i y_i)^2 \\ & + (\beta_i - a_4 - a_5 x_i - a_6 y_i \\ & - a_7 x_i y_i - a_8 y_i^2)^2] \end{aligned} \quad (4.12)$$

where, for each i between 1 and n , $(\alpha_i, \beta_i) = (\alpha(x_i, y_i), \beta(x_i, y_i))$ is a flow vector and W_i is the corresponding weight. Taking partial derivatives with respect to a_1, \dots, a_8 and equating to 0, a set of eight linear equations is obtained. If these equations are independent, their solution, denoted by a_1^*, \dots, a_8^* , represents the optimal Ψ transformation. Substituting this solution in (4.12) and using the normalization equation

$$\sigma = \sqrt{E(a_1^*, \dots, a_8^*) / \sum_{i=1}^n W_i} \quad (4.13)$$

an error value, corresponding to the given set of flow vectors, is obtained. σ is an estimate of the standard deviation of the actual flow values from those predicted by the optimal Ψ transformation.

b) *Criteria for a Merging Decision:* Let F denote a set of adjacent components $\{C_j : j = 1, \dots, m\}$. The decision whether to merge these components into one segment is based on the degree of their consistency with the same Ψ transformation. To determine such a consistency, an optimal Ψ transformation and a related error measure, denoted, respectively, by Ψ_j and σ_j , are separately computed for each component C_j , $j = 1, \dots, m$. In addition, an optimal Ψ transformation, denoted by Ψ_F , is computed for the set of vectors $\cup_{j=1}^m C_j$. Substituting the coefficients of Ψ_F and the flow values of the vectors contained in C_j , $1 \leq j \leq m$, in (4.12) and (4.13), a new error value σ'_j is obtained for each component C_j . In the following discussion, p_j is the ratio of the sum of vector weights in C_j to the total sum of weights in the set F .

For each $1 \leq j \leq m$, σ'_j is never less than σ_j , since Ψ_j can be adjusted to the local surface and noise associated with C_j . If, however, a merging decision is justified and p_j is close to 1,

then σ'_j can be expected to be very close to σ_j . On the other hand, if p_j is close to 0, then σ'_j should be allowed to be significantly higher than σ_j , up to a given upper bound. This upper bound is determined to be

$$l_j = \max \{l_a \sigma_j, l_b\} \quad (4.14)$$

where l_a and l_b are given thresholds. The value of l_a is typically around 1.5, while l_b represents a reasonable upper bound on the noise level. If, for example, the most significant noise is induced by using flow values rounded to integers and, therefore, the noise is uniformly distributed between -0.5 pixels and +0.5 pixels, then l_b will be $\sqrt{2}/0.5 \approx 0.7$. (The deviation between the measured and predicted flow values is measured as a distance in the 2-D image plane, therefore, 0.5 is multiplied by $\sqrt{2}$.) To reflect the dependence on the value of p_j , the allowed level of σ'_j is

$$\sigma'_j = p_j \sigma_j + (1 - p_j) l_j. \quad (4.15)$$

Thus, a merging decision is accepted if, and only if, for each component C_j in F

$$\sigma'_j \leq L_j. \quad (4.16)$$

c) *The Merging Procedure:* The algorithm for finding sets of components to be merged starts by computing an optimal Ψ transformation and a related error measure for each component. After this initial step, segments are sequentially created by testing possible mergings of components. In each cycle of the algorithm, a given segment is formed, and only components which are not yet assigned to any of the already created segments are considered as candidates for merging into this segment. Eventually, all the components are contained in one of the segments.

The process for creating a given segment starts by detecting the largest component, denoted by C^* , among the candidates for forming this segment. Then, other candidates are sequentially tested, in the order of their associated sums of weights, for merging with C^* . In general, given a set of components which are already assigned to the segment, neighboring components which have not been examined yet are sequentially tested for merging with this set. Once a merging decision is made, additional components are tested for merging with the new set. This process continues until all the candidates for forming the segment are examined. Results of the merging stage are demonstrated in Fig. 3(c).

3) *Third Stage—Assimilating Ungrouped Flow Vectors:* The purpose of the third stage of the segmentation is examination of flow vectors which were assigned positive weights but were not grouped into any of the components in the first stage of the segmentation and, therefore, do not belong to any of the segments. Such vectors, called 0-vectors, which are neighbors of one of the segments, are tested for consistency with the Ψ transformation corresponding to this segment and, if consistent, are merged into it. Then, 0-vectors, neighbors of the just segmented vectors, are examined in their turn. This process is iteratively executed until no new vector is merged into one of the segments. Results of this stage of the segmentation are demonstrated in Fig. 3(d).

It is possible that, after the third stage, connected sets of 0-vectors, which were not excluded from further consideration

in the first segmentation stage, are still not contained in any of the existing segments. In such a case, the first and the second stages of the segmentation are executed again, focused only on these sets, thus possibly creating new segments.

V. FORMING OBJECT HYPOTHESES AND RECOVERING THREE-DIMENSIONAL MOTION AND STRUCTURE

In the first phase of the interpretation process, described in the preceding section, the flow field is segmented into connected sets of flow vectors, where each set is consistent with a rigid motion of a roughly planar surface. Such a segment is assumed to correspond to a portion of only one rigid object. The next task is to group segments which are consistent with the same 3-D motion parameters. Such a group can be hypothesized, employing the rigidity assumption [28], to be induced by one rigidly moving object (or by the camera motion). In Section V-A, we describe a general algorithm for computing the motion parameters from a set of flow vectors generated by a rigid motion. In Section V-B, we combine this algorithm with the segmentation results to form object hypotheses and to estimate the corresponding 3-D motion and structure. Again, details can be found in [4].

A. Estimating Motion Parameters and Depth Information of a Rigid Object

1) *Optimization Constraint:* Given a set of flow vectors, assumed to be induced by a rigidly moving object, we wish to find the 3-D motion parameters and depth values which are maximally consistent with these data. Following [8], we employ a least-squares approach which minimizes the deviation between the measured flow field and that predicted from the estimated motion and structure. This approach is adopted because of its relative robustness in the presence of noise. Based on (3.2), the error function to be minimized is

$$\begin{aligned} \sum_{i=1}^n W_i & [(\alpha_i + \Omega_X x_i y_i - \Omega_Y (1 + x_i^2) \\ & + \Omega_Z y_i - (T_X - T_Z x_i)/Z_i)^2 \\ & + (\beta_i + \Omega_X (1 + y_i^2) - \Omega_Y x_i y_i \\ & - \Omega_Z x_i - (T_Y - T_Z y_i)/Z_i)^2] \end{aligned} \quad (5.1)$$

where $\mathbf{T} = (T_X, T_Y, T_Z)$ and $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ are the translation and rotation vectors, respectively, and for each i between 1 and n , (α_i, β_i) is the flow vector computed at the pixel (x_i, y_i) , W_i is its weight, and Z_i is the spatial depth of the corresponding point in the environment. The task is to determine $\mathbf{T}, \boldsymbol{\Omega}$, and $\{Z_i\}$ which minimize this function. Using the decomposition of the flow field into its rotational and translational components, denoted by (α_R, β_R) and (α_T, β_T) [see (3.3) and (3.4)], the error function can be more concisely represented by

$$\sum_{i=1}^n W_i [(\alpha_i - \alpha_{Ri} - \alpha_{Ti})^2 + (\beta_i - \beta_{Ri} - \beta_{Ti})^2]. \quad (5.2)$$

As can easily be seen, it is actually impossible to determine the absolute values of (T_X, T_Y, T_Z) and $\{Z_i : i = 1, \dots, n\}$. However, if the magnitude, denoted by r , of the translation is nonzero, then it is possible to estimate the direction of the 3-D translation, represented by the unit vector

$$(U_X, U_Y, U_Z) = (T_X, T_Y, T_Z)/r \quad (5.3)$$

and the relative depth values, represented by

$$\tilde{Z}_i = r/Z_i, \quad i = 1, \dots, n. \quad (5.4)$$

Introducing the abbreviations

$$\alpha_U = U_X - U_Z x = \alpha_T/\tilde{Z} \quad (5.5a)$$

and

$$\beta_U = U_Y - U_Z y = \beta_T/\tilde{Z} \quad (5.5b)$$

(5.2) can be rewritten as

$$\sum_{i=1}^n W_i [(\alpha_i - \alpha_{Ri} - \alpha_{Ui}\tilde{Z}_i)^2 + (\beta_i - \beta_{Ri} - \beta_{Ui}\tilde{Z}_i)^2]. \quad (5.6)$$

Thus, the task can be reformulated as one of finding the values of $(U_X, U_Y, U_Z), (\Omega_X, \Omega_Y, \Omega_Z)$, and $\{\tilde{Z}_i : i = 1, \dots, n\}$ which minimize this expression. In addition, the depth constraints

$$\tilde{Z}_i \geq 0, \quad i = 1, \dots, n \quad (5.7)$$

should be satisfied. Note that this error measure is different from the one actually employed in [8] where the contribution of each flow vector is multiplied by $\alpha_U^2 + \beta_U^2$.

For any given i , $1 \leq i \leq n$, we can find the optimal value of \tilde{Z}_i , as a function of the motion parameters, by examining the first derivative of (5.6) with respect to \tilde{Z}_i . This derivative is given by

$$2W_i[-(\alpha_i - \alpha_{Ri})\alpha_{Ui} - (\beta_i - \beta_{Ri})\beta_{Ui} + (\alpha_{Ui}^2 + \beta_{Ui}^2)\tilde{Z}_i]. \quad (5.8)$$

Setting it equal to 0 yields

$$\tilde{Z}_i = ((\alpha_i - \alpha_{Ri})\alpha_{Ui} + (\beta_i - \beta_{Ri})\beta_{Ui})/(\alpha_{Ui}^2 + \beta_{Ui}^2) \quad (5.9)$$

unless $\alpha_{Ui}^2 + \beta_{Ui}^2 = 0$, in which case \tilde{Z}_i can be assigned any non-negative value. If the expression in (5.9) is negative, then the corresponding depth constraint in (5.7) is unsatisfied. In such a case, to minimize the error function (5.6), \tilde{Z}_i should be set to 0, since the derivative (5.8) is nonnegative for nonnegative values of \tilde{Z}_i and, therefore, the error function is monotonically nondecreasing for these values. To summarize, the optimal value of \tilde{Z}_i is given by

$$\tilde{Z}_i = [((\alpha_i - \alpha_{Ri})\alpha_{Ui} + (\beta_i - \beta_{Ri})\beta_{Ui})/(\alpha_{Ui}^2 + \beta_{Ui}^2)]^+. \quad (5.10)$$

Substituting (5.10), for any $1 \leq i \leq n$, into (5.6), and expanding the resulting expression yields the following representation of the error, as a function of the motion parameters:

$$E(\mathbf{U}, \boldsymbol{\Omega}) = \sum_{i=1}^n W_i E_i \quad (5.11a)$$

where

$$E_i = \begin{cases} \frac{[(\alpha_i - \alpha_{Ri})\beta_{Ui} - (\beta_i - \beta_{Ri})\alpha_{Ui}]^2}{\alpha_{Ui}^2 + \beta_{Ui}^2} & \text{if } (\alpha_i - \alpha_{Ri})\alpha_{Ui} + (\beta_i - \beta_{Ri})\beta_{Ui} > 0 \\ (\alpha_i - \alpha_{Ri})^2 + (\beta_i - \beta_{Ri})^2 & \text{otherwise.} \end{cases} \quad (5.11b)$$

A normalized version of this error function, defined by

$$\sigma(\mathbf{U}, \boldsymbol{\Omega}) = \sqrt{E(\mathbf{U}, \boldsymbol{\Omega}) / \sum_{i=1}^n W_i} \quad (5.12)$$

will also be utilized. σ is an estimate of the standard deviation of the measured flow values from those predicted by the motion parameters and the corresponding depth values.

Note that the expression (5.11) for the error function was obtained by assuming a nonzero translation. In the case of a purely rotational motion, the appropriate error function to be minimized is

$$E_R(\boldsymbol{\Omega}) = \sum_{i=1}^n W_i [(\alpha_i - \alpha_{Ri})^2 + (\beta_i - \beta_{Ri})^2]. \quad (5.13)$$

The minimal value of this function is never less than the minimal value of $E(\mathbf{U}, \boldsymbol{\Omega})$, since the latter function is derived from (5.6), where the values of \tilde{Z}_i , $i = 1, \dots, n$, can be chosen to be 0, thus predicting a purely rotational flow field. If, however, the minimal value of $E_R(\boldsymbol{\Omega})$ is close to the minimal value of $E(\mathbf{U}, \boldsymbol{\Omega})$, then the 3-D motion is, possibly, purely rotational.

The task of finding the three rotation parameters which minimize the function $E_R(\boldsymbol{\Omega})$ can easily be accomplished by taking the partial derivatives of $E_R(\boldsymbol{\Omega})$ with respect to the rotation parameters, setting them equal to 0, and solving the linear equations so obtained [8]. In the next section we concentrate on the much more difficult task of finding values of \mathbf{U} and $\boldsymbol{\Omega}$ which minimize the error function $E(\mathbf{U}, \boldsymbol{\Omega})$ [or, equivalently, the function $\sigma(\mathbf{U}, \boldsymbol{\Omega})$], where \mathbf{U} can be any unit vector and $\boldsymbol{\Omega}$ is unconstrained.

2) Algorithm: The algorithm for recovering the motion parameters employs an error measure, derived from (5.12), corresponding to possible locations of the FOE in the image plane. For each hypothesized FOE, the optimal rotation parameters and a related error value are computed. A minimum value of the resulting error function is determined, using a multiresolution sampling technique. Notice the difference of this approach from the one employed in [23], where for each hypothesized *rotational* component the FOE of the corresponding translational field and a related error measure are computed. We believe that our approach is more efficient, since the search space is only 2-D, as opposed to 3-D in [23], and the error function which we employ is easier to compute. Before we proceed with a mathematical description of the algorithm, we refer the reader to the geometrical interpretation given in Fig. 2.

a) Reducing the Search Space: In this subsection, we show how to derive a new error function from $E(\mathbf{U}, \boldsymbol{\Omega})$. This function will be defined only on the unit hemisphere which is isomorphic to the image plane, thus reducing the dimensionality of the search space from five to two. Let us start the derivation of the new function with the observation that if the depth constraints (5.7) are ignored, then, for any hypothesized direction of translation, the optimal rotation parameters can easily be extracted by solving a set of three linear equations. To see that, notice that the error functions (5.11) can be reduced in this case to the function

$$E'(\mathbf{U}, \boldsymbol{\Omega}) = \sum_{i=1}^n W_i [((\alpha_i - \alpha_{Ri})\beta_{Ui} - (\beta_i - \beta_{Ri})\alpha_{Ui})^2 / (\alpha_{Ui}^2 + \beta_{Ui}^2)]. \quad (5.14)$$

Differentiating $E'(\mathbf{U}, \boldsymbol{\Omega})$ with respect to the rotation parameters and setting the derivatives equal to 0, yields three linear

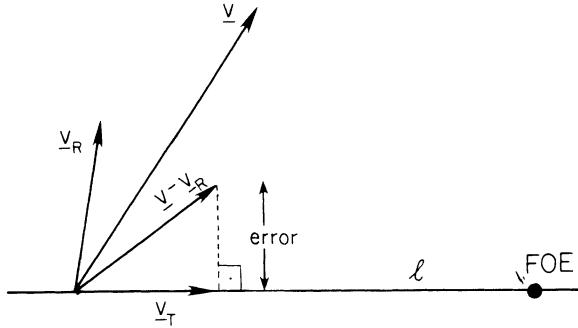


Fig. 2. A geometrical interpretation of the algorithm. l is the line connecting the initial point of a flow vector v to a hypothesized location of the FOE. We wish to find rotation parameters such that the difference $v - v_R$, where v_R is the corresponding rotational component, will be oriented toward or away from the FOE. The projection of $v - v_R$ on l is v_T , the estimated translational component of the flow vector. The related depth value \tilde{Z} is determined as the one which yields such a translational component. The error measure is the distance between the end points of $v - v_R$ and v_T . Note that the depth constraint means that if the Z component of the object translation is positive, then v_T should be oriented *toward* the FOE, while if the Z component is negative, v_T should be oriented *from* the FOE. If the depth constraint is unsatisfied, then we choose v_T to be $(0, 0)$.

equations with the rotation parameters as unknowns. Thus, ignoring the depth constraints (5.7), the search space can be limited to the unit sphere $\{\mathbf{U}: |\mathbf{U}| = 1\}$.

Moreover, changing the sign of any unit vector \mathbf{U} has no effect on the value of $E'(\mathbf{U}, \boldsymbol{\Omega})$ since it only affects the sign of α_U and β_U . Therefore, the search space can be further restricted to the hemisphere

$$H = \{\mathbf{U}: |\mathbf{U}| = 1 \text{ and } U_Z \geq 0\}. \quad (5.15)$$

When an optimal value of \mathbf{U} is found in H , its preferred sign can be determined as the one which gives $\tilde{Z}_i \geq 0$ for most indexes i [8]. Still, we wish to incorporate these constraints or, equivalently, the equations (5.11b) in a more rigorous way. Hence, for each \mathbf{U} in H , we define the error measure

$$\sigma_H(\mathbf{U}) = \min_{s, \boldsymbol{\Omega}} \sigma(s\mathbf{U}, \boldsymbol{\Omega}) \quad (5.16)$$

where s can have the values +1 or -1. The goal is to find a vector \mathbf{U} in H which minimizes the function σ_H . The associated values of s and $\boldsymbol{\Omega}$ are, respectively, the determined sign of the translation vector and the estimated rotation parameters. The function σ_H is, however, difficult to compute. Therefore, in the proposed algorithm we compute an approximation to σ_H which, in the experiments, was usually found to be very accurate. A few main steps can be distinguished in the procedure for computing this approximation as follows.

1) Given a vector \mathbf{U} in H , estimate the optimal rotation vector $\boldsymbol{\Omega}^*$ by minimizing $E'(\mathbf{U}, \boldsymbol{\Omega})$ with respect to $\boldsymbol{\Omega}$, and compute the corresponding normalized error measure $\sigma'(\mathbf{U}, \boldsymbol{\Omega}^*)$. This error value, denoted by $\sigma_H^\downarrow(\mathbf{U})$, is a lower bound of $\sigma_H(\mathbf{U})$ since it minimizes the error function $\sigma(\mathbf{U}, \boldsymbol{\Omega})$, with respect to $\boldsymbol{\Omega}$ and the sign of \mathbf{U} , without considering the depth constraints (5.7).

2) Compute $\sigma(\mathbf{U}, \boldsymbol{\Omega}^*)$ and $\sigma(-\mathbf{U}, \boldsymbol{\Omega}^*)$. Determining the minimum of these two error values yields the preferred sign, denoted by s^* , of \mathbf{U} . The value $\sigma(s^*\mathbf{U}, \boldsymbol{\Omega}^*)$, denoted by $\sigma_H^\uparrow(\mathbf{U})$, is an upper bound of $\sigma_H(\mathbf{U})$, because it gives the actual error

measure for some values of s and $\boldsymbol{\Omega}$ in (5.16).

3) Compute an approximation of $\sigma_H(\mathbf{U})$ by averaging its lower and upper bounds

$$\hat{\sigma}_H(\mathbf{U}) = (\sigma_H^\downarrow(\mathbf{U}) + \sigma_H^\uparrow(\mathbf{U}))/2. \quad (5.17)$$

The relative deviation of $\sigma_H(\mathbf{U})$ from $\hat{\sigma}_H(\mathbf{U})$ is bounded by

$$(\sigma_H^\uparrow(\mathbf{U}) - \sigma_H^\downarrow(\mathbf{U}))/(2\hat{\sigma}_H(\mathbf{U})). \quad (5.18)$$

In the experiments, this value is usually very small, typically much less than 0.01.

Notice that

$$\min_U \sigma_H^\downarrow(\mathbf{U}) \leq \min_U \sigma_H(\mathbf{U}) \leq \min_U \sigma_H^\uparrow(\mathbf{U}) \quad (5.19a)$$

and

$$\min_U \sigma_H^\downarrow(\mathbf{U}) \leq \min_U \hat{\sigma}_H(\mathbf{U}) \leq \min_U \sigma_H^\uparrow(\mathbf{U}). \quad (5.19b)$$

Therefore, if the difference between the minimal values of the lower and upper bounds of σ_H is very small, then $\min \hat{\sigma}_H \approx \min \sigma_H$, and (see [4]) the solution obtained by minimizing $\hat{\sigma}_H$ is guaranteed to be optimal or very close to optimal. In rare situations (which never occurred in our experiments) where this difference is relatively large, we can still eliminate, from further examination, all the vectors \mathbf{U} such that $\sigma_H^\downarrow(\mathbf{U})$ is not significantly smaller than the minimal value of the function σ_H^\downarrow . If some of the vectors are not eliminated, then the corresponding values of σ_H should be more accurately approximated. This goal can be achieved by searching for optimal rotation parameters using, for example, a sampling technique, and substituting the results in (5.12).

b) Search Strategy: The search for an optimal vector in H consists of a sampling of the error measure $\hat{\sigma}_H$. A multiresolution scheme is employed where in the first iteration the set H is coarsely sampled (similar to [14]) and in each additional iteration only the neighborhood of the vector giving a minimum value in the previous iteration is sampled, using a finer resolution. Note that solutions near the boundary of H require a vector \mathbf{U}' to be defined as a “neighbor” of a vector \mathbf{U} if either \mathbf{U}' or $-\mathbf{U}'$ is close to \mathbf{U} . Another way to obtain the same effect while using the normal definition of a neighborhood is to extend the domain of definition of the function $\hat{\sigma}_H$ to the whole unit sphere, employing exactly the same definition used for the domain H . In this case, $\hat{\sigma}_H(-\mathbf{U}) = \hat{\sigma}_H(\mathbf{U})$ for each unit vector \mathbf{U} , and hence, computationally, it makes no difference which domain of definition is used.

In [14] the unit sphere is uniformly sampled in the first step of the search process. We propose a different approach where the density of the sampling in a given region of H is proportional to the rate of change of the flow field, as a function of \mathbf{U} , when \mathbf{U} varies in this region. Using this approach, the relative number of samples is decreased in areas of the hemisphere where the error function is expected to be flat, but it is increased in areas where this function is expected to vary rapidly. Thus, for a given number of samples, the probability of missing the correct solution, because of a too sparse sampling around it, becomes smaller. This idea is more quantitatively explored in [4].

The solution of \mathbf{U} , found in the last iteration of the sampling procedure, and the corresponding sign s^* and the rotation pa-

rameters Ω^* , defined in the procedure for computing $\hat{\sigma}_H$, are the determined motion parameters. By substituting these parameters in (5.10), the relative depth corresponding to each flow vector can be estimated as well.

3) *Comments on Inherent Ambiguity and Instability:* The flow field equations (3.2) may, in general, have more than one solution. If, for example, the surface is planar, then usually two solutions exist [27], [30]. Thus, multiple solutions should be searched for on the error surface. This can be done by locating locally minimal error values which are also very close to the global minimum.

Another problem is the inherent instability which may exist in recovering 3-D motion and structure from *noisy* flow fields. Such a situation can be identified as one in which the error function $\hat{\sigma}_H$ (5.17) is very close to its minimal value in a large portion of the search space [see, for example, Fig. 4(e)]. In this case, reasonably reliable estimates of the motion parameters may be impossible to obtain. The influence of certain parameters on this instability is analyzed in [4]. From this analysis it can be concluded that the following conditions contribute to instability: the field of view is small, the algorithm is locally applied to the image, the resolution of the image is low, the noise level (in pixels) is high, the flow field is sparse and/or the noise samples in adjacent vectors are highly correlated, the depth variation is small, the translation is small, and the object is far away from the camera.

Two complementary approaches may be taken in order to deal with this instability. First, *constraints* on the motion parameters and the environmental depth, rather than values, can still be recovered, using, for example, the coefficients of the related Ψ transformations [see (4.4)]. Second, possible values of the motion parameters can be represented by a *probabilistic* distribution function. Such a function can be defined, for example, on the set H , using the computed values of $\hat{\sigma}_H$.

B. Forming Object Hypotheses

Segments of the flow field which are consistent with the same motion parameters can be hypothesized, using the rigidity assumption [28], to be induced by one rigidly moving object (or by the camera motion). The process for detecting such sets of segments is similar to the second stage of the segmentation process, where components are merged into segments. Optimal motion parameters and a related error measure ϵ_i are computed for each segment S_i , using the algorithm described in the previous section. In addition, given any set of segments, the algorithm is applied to this set and the corresponding motion parameters are computed. Then, for each segment S_i in the set, an error measure ϵ'_i is obtained by substituting these parameters and the related flow data in (5.17). Based on the error values $\{\epsilon_i\}$ and $\{\epsilon'_i\}$, consistency of the set with rigid motion is determined, employing a decision procedure similar to the one described in Section IV-B-2 (for more details see [4]).

Actually, each segment is sampled, using the method in step 2) of the multipass Hough technique (Section IV-B-1-b), and only the selected vectors are used for forming object hypotheses and computing the corresponding motion parameters. This sampling procedure considerably reduces the computation

time. Notice that, because each segment is sampled, all the distinct surfaces and independently moving objects, even the small ones, are appropriately represented, thus preventing the suppression of valuable data.

In addition to the ambiguity described in the previous section, another ambiguity may exist in the decomposition of the environment into independently moving objects. For example, two independently moving objects induce, in some cases, optical flows which can be interpreted as resulting from one rigidly moving object [4]. In order to deal with this ambiguity, one may have to find a *set* of possible decompositions, not only one.

VI. EXPERIMENTS

In this section, we present four experiments which demonstrate our proposed scheme for the interpretation of optical flow fields. The first two experiments are based on simulated data, and the last two are based on images taken from a video camera in the University of Massachusetts Robotics Lab. An additional experiment based on real data is described in [3]. In all the experiments, values to appear in translation vectors and surface equations are given in focal units, whereas rotation parameters are given in degrees and flow vectors are given in pixel units. Actually, the flow values in the experiments based on simulated data are rounded to integers, thus inducing noise uniformly distributed between $-\frac{1}{2}$ and $+\frac{1}{2}$ pixels. The methods employed for computing the real data in experiments 3 and 4 also produce flow values given in integer units, hence, the noise level in these experiments should be at least as high as in experiments 1 and 2 (actually it is higher). The image, in all the experiments, contains 128×128 pixels. The field of view of the camera is 45° in the experiments with simulated data and 30° in the experiments with real data.

A. Experiment 1

The first experiment simulates a translatory motion of the camera, represented by the vectors $T_C = (0, 0.02, 1)$ and $\Omega_C = (0^\circ, 0^\circ, 0^\circ)$. The environment consists of two distinct surfaces: a plane described by the equation $Z = 50Y + 100$ and an ellipsoid represented by $(X - 2)^2 + [(Y - 2)/4]^2 + (Z - 5)^2 = 1$. A flow vector is computed for each pixel, unless the corresponding ray of light does not intersect any of the surfaces, in which case the related weight is assumed to be 0 (otherwise it is 1). A sample of the flow field is shown in Fig. 3(a).

The results of the three stages of the segmentation, shown in Fig. 3(b)-(d), demonstrate the role and importance of each of these stages. The two segments found in this process were determined to be consistent with the same rigid motion. The error function $\hat{\sigma}_H$ (5.17) was computed using 64 vectors from each segment. Employing a spherical coordinate system (r, ϕ, θ) , where

$$X = r \sin(\phi) \cos(\theta) \quad (6.1a)$$

$$Y = r \sin(\phi) \sin(\theta) \quad (6.1b)$$

and

$$Z = r \cos(\phi) \quad (6.1c)$$

the domain of definition of $\hat{\sigma}_H$, that is, the hemisphere $\{U: |U| =$

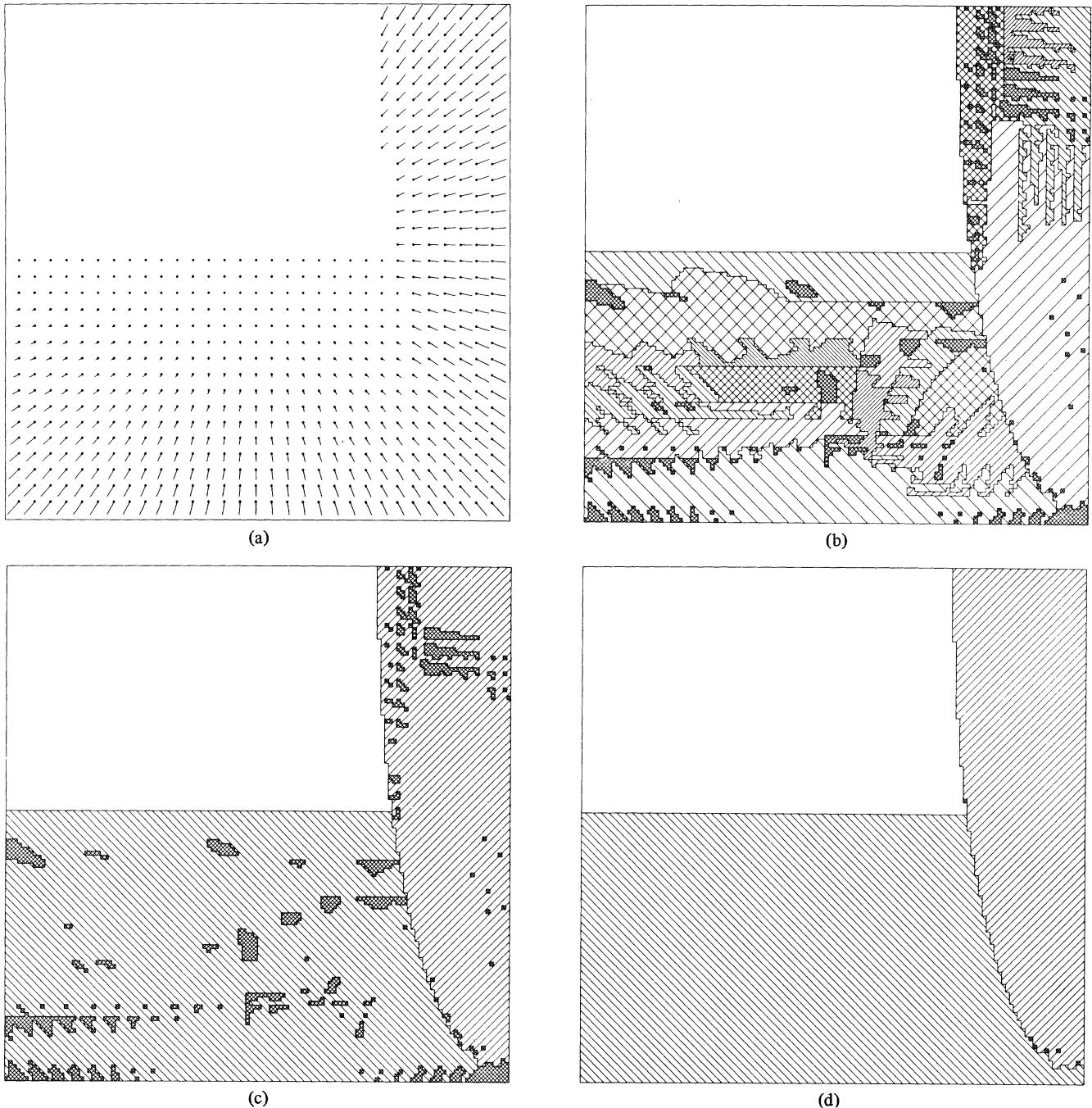


Fig. 3. Experiment 1. (a) A sample of the flow field. The initial point of each vector is marked by a dot. The length of the vectors is scaled by 0.25. (b) Components, represented by line patterns, which were determined in the first step of the segmentation. The small areas with the densest pattern correspond to vectors which are not contained in any of the components. The irregular shapes of the components were caused by the roundoff error. (c) Segments obtained by merging components consistent with the same Ψ transformation. (d) Final segmentation. Note that almost all the flow vectors which, after the first stage, had not been included in any of the components, were merged into the segments.

$1, U_Z \geq 0\}$, can be represented by the set

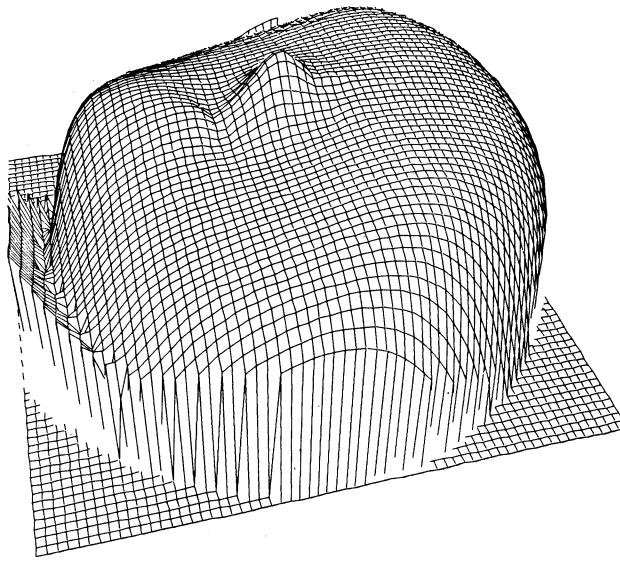
$$\{(\phi, \theta) : 0 \leq \phi \leq 90^\circ, 0^\circ \leq \theta < 360^\circ\}. \quad (6.2)$$

This representation is utilized for displaying the function $\hat{\sigma}_H$ in Fig. 3(e), where (ϕ, θ) are used as polar coordinates. Employing the sampling procedure for minimizing $\hat{\sigma}_H$, the motion parameters were determined, after two iterations, to be $\mathbf{U} = (0.0017, -0.0204, -0.9998)$ and $\boldsymbol{\Omega} = (-0.02^\circ, -0.01^\circ, -0.02^\circ)$. Note that, assuming a stationary environment, the camera motion is given by $-\mathbf{U}$ and $-\boldsymbol{\Omega}$. These results are in a good agreement with the correct values. Substituting the computed

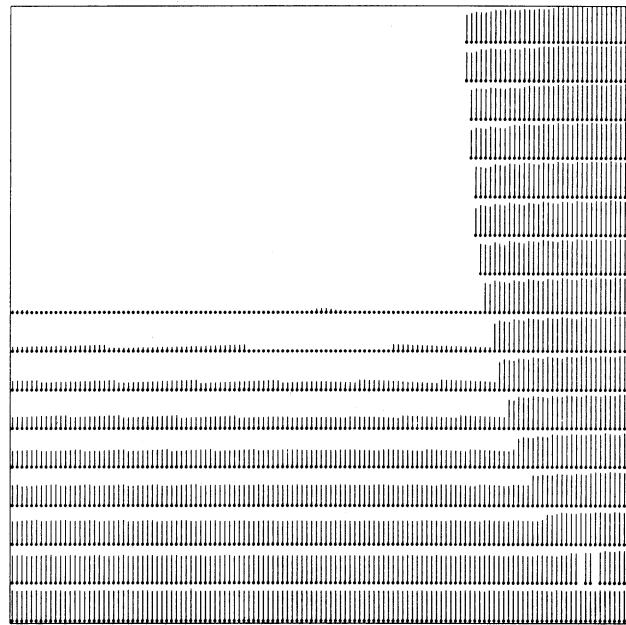
values in (5.10), the “reciprocal depth” map, that is, the function r/Z shown in Fig. 3(f), was obtained. Notice the effect of the roundoff error, especially near the FOE which is close to the center of the image. The average of the relative errors in the estimated values of r/Z is 12.1 percent.

B. Experiment 2

In the second experiment, the camera motion is composed of both translation and rotation, described by $T_C = (0.5, 0.5, 1)$ and $\boldsymbol{\Omega}_C = (1.15^\circ, -1.15^\circ, 2.86^\circ)$. The environment contains



(e)



(f)

Fig. 3. (Continued.) (e) The error function \hat{o}_H , shown inverted, defined on the hemisphere $\{\mathbf{U} : |\mathbf{U}| = 1, U_Z \geq 0\}$. The spherical coordinates (ϕ, θ) , employed in (6.2) for representing this hemisphere, are used here as polar coordinates. The range of the function is [0.275, 8.605] (in pixels), that is, 0.275 is the error value associated with the peak of the displayed surface (this peak corresponds to the estimated translation axis), and 8.605 is the error value associated with the reference plane at the bottom of the surface. (f) The function r/Z , where r is the length of the translation vector and Z is the environmental depth. The length of each bar represents the relative value of r/Z at the image pixel corresponding to the attached dot.

an independently moving sphere, defined by $(X - 9)^2 + (Y - 9)^2 + (Z - 30)^2 = 4$. An object coordinate system is defined which is parallel to the camera coordinate system, but its origin is in the sphere center (9, 9, 30). The motion of the object in this coordinate system is represented by $T_O = (0.5, -0.5, 0)$ and $\Omega_O = (0^\circ, 0^\circ, -11.46^\circ)$. The stationary environment is composed of two surfaces: a plane described by $Z = X + 0.5Y + 50$, and an ellipsoid described by $[(X + 3)/2]^2 + [(Y + 1)/5]^2 + [(Z - 20)/2]^2 = 1$. A 32×32 sample of the flow field corresponding to this scene is shown in Fig. 4(a).

The segments found in the experiment are shown in Fig. 4(b). The two segments associated with the stationary environment were determined to be consistent with the same rigid motion, while no rigid motion compatible with the third segment was also found to be consistent with one of the other segments. Thus, the decomposition of the flow field into sets corresponding to independently moving objects could be uniquely (and correctly) determined. The error function $\hat{\sigma}_H$ corresponding to the stationary environment is displayed in Fig. 4(c). The associated motion parameters of the camera were determined to be $-U = (0.3899, 0.4037, 0.8277)$ (the corresponding actual values were $U_C = (0.4082, 0.4082, 0.8164)$) and $-\Omega = (1.17^\circ, -1.12^\circ, 2.83^\circ)$. The related depth map is represented by the function r/Z in Fig. 4(d). Again, the relative errors in r/Z are mainly a direct consequence of the roundoff errors in the corresponding flow values. Their average value, in this experiment, is 14.7 percent.

The error function corresponding to the independently moving object is shown in Fig. 4(e). This function is very close to its minimal value in a large portion of the search space, thus demonstrating the instability discussed in Section V-A-3.

C. Experiment 3

The third experiment, taken from [24], is based on real data shown in Fig. 5(a) and (b). The scene contains two textured cylinders in front of a textured plane parallel to the image plane. The camera was translated roughly in the direction of the Z axis, and then rotated about its Y axis. Fig. 5(c) shows the flow vectors determined for a set of interesting points extracted from the image; for more details on how the flow field was extracted, see [24]. The weight assigned to each vector is 1, since no reliability measure was computed.

The results of the segmentation process are shown in Fig. 5(d). The three segments found in this process are compatible with the same camera motion. Fig. 5(e) displays the corresponding error function $\hat{\sigma}_H$. Assuming a stationary environment, the recovered motion parameters of the camera, $-U = (-0.0079, 0.0181, 0.9998)$ and $-\Omega = (-0.1^\circ, -1.16^\circ, 0.01^\circ)$, are consistent with the specifications of the experiment. The three distinct surfaces in the environment and their relative distances from the camera can be perceived in the depth map in Fig. 5(f). Notice, however, the large depth variation in the central part of the image, whereas the actual values are approximately constant in this area. These errors are caused by the presence of noise in the flow values near the focus of expansion and are unavoidable in such circumstances. Note that this experiment demonstrates the ability of our scheme to interpret sparse flow fields.

D. Experiment 4

Fig. 6(a) and (b) shows images taken from a camera translated in the direction of its X axis and then rotated about its Y axis 1.5° . In addition, an object (a “dinosaur” toy) was rotated

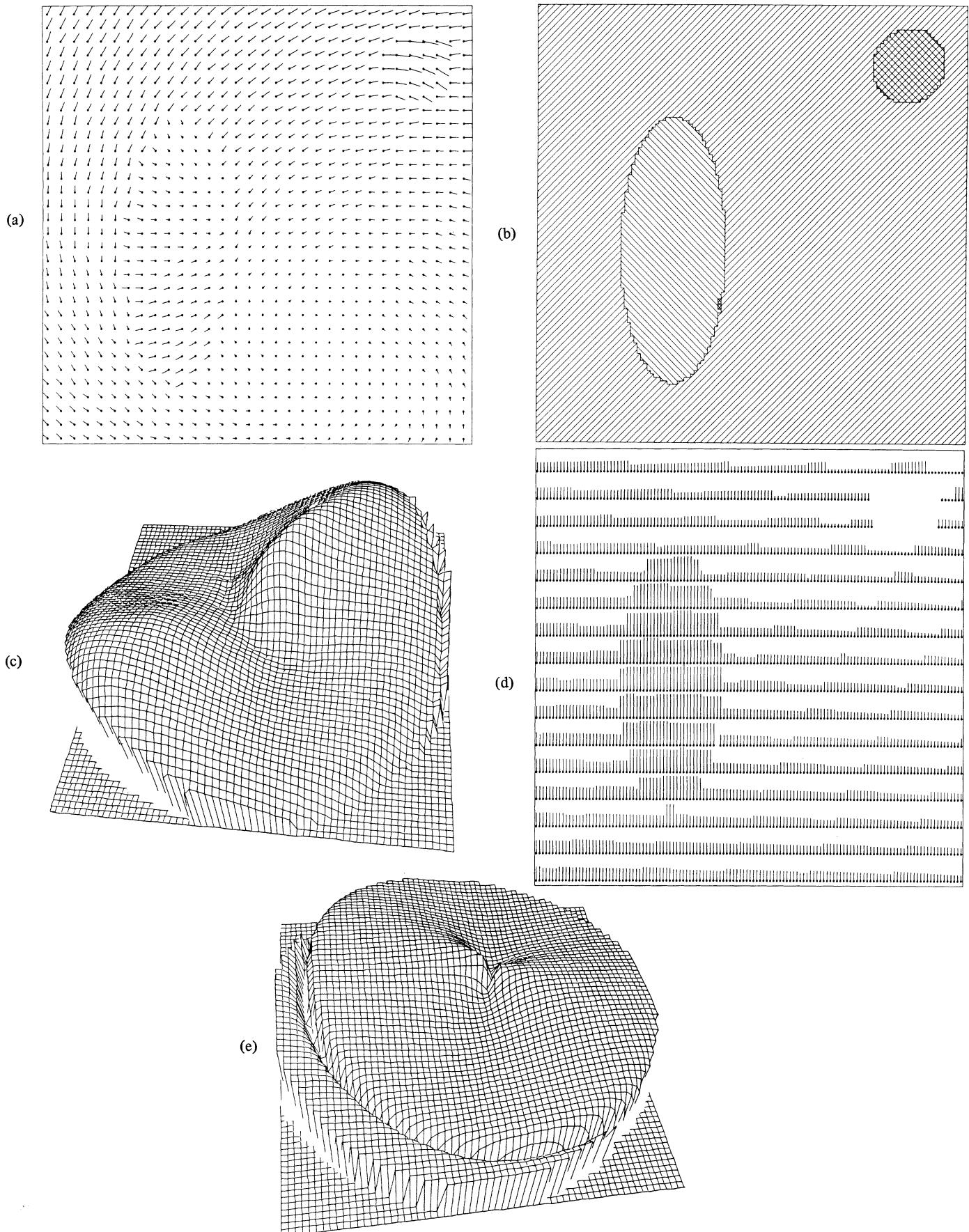


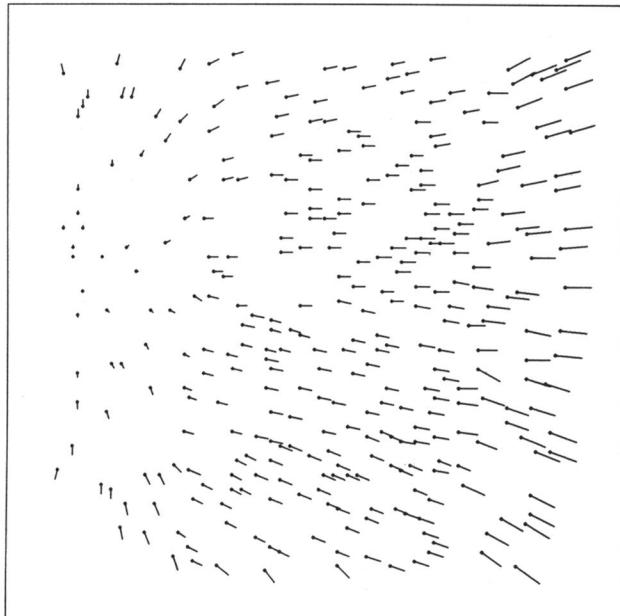
Fig. 4. Experiment 2. (a) A 32×32 sample of the flow field. The vectors are scaled by 0.5. (b) Final segmentation. (c) The error function $\hat{\sigma}_H$, shown inverted, corresponding to the stationary environment. The range of the function is [0.281, 2.710]. (d) The depth function r/Z corresponding to the stationary environment. The roundoff error has a strong effect, especially in the upper right corner near the focus of expansion. (e) The error function $\hat{\sigma}_H$, shown inverted, corresponding to the moving object. The range of this function is [0.257, 0.390].



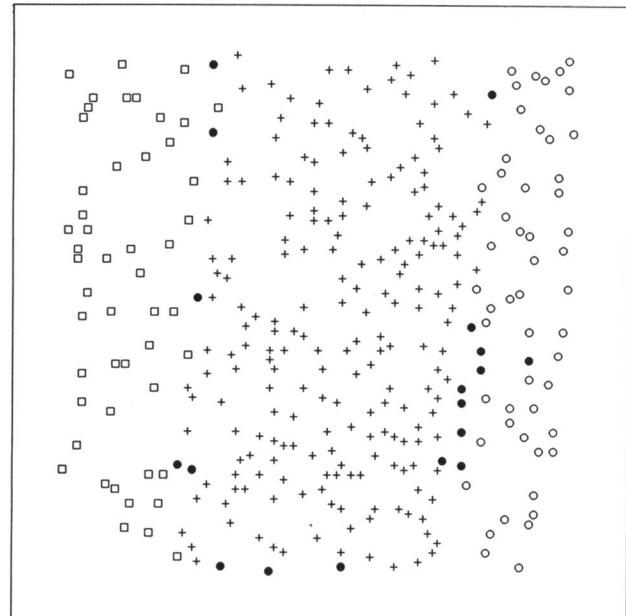
(a)



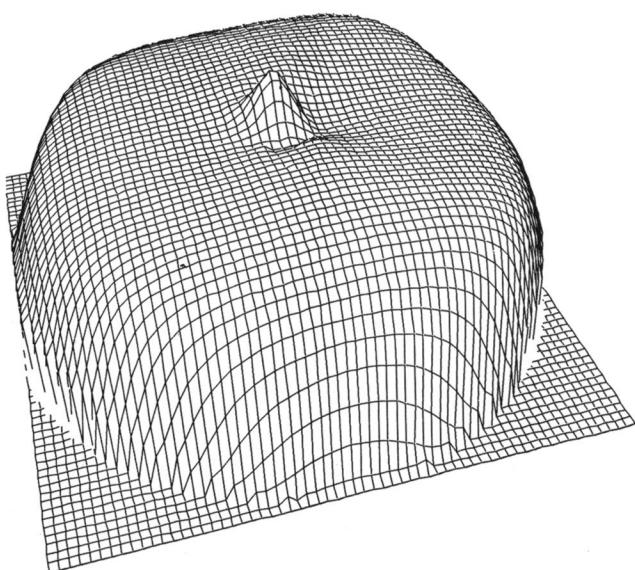
(b)



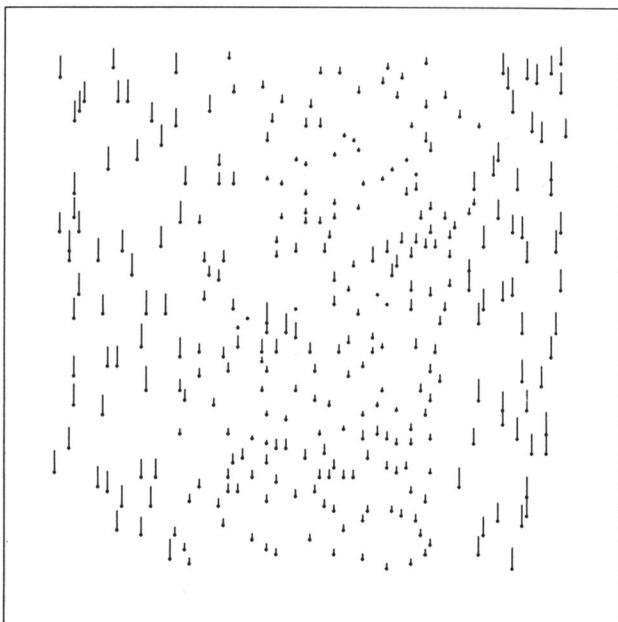
(c)



(d)



(e)



(f)

Fig. 5. Experiment 3. (a) The first intensity image. (b) The second intensity image. (c) The flow field produced in [24]. The vectors are scaled by 0.5. (d) Final segmentation. Each segment is represented by a distinct shape; the black dots correspond to flow vectors which are not contained in any of the segments. (e) The error function $\hat{\sigma}_H$ shown inverted. The range of the function is [0.356, 3.571]. (f) The estimated depth function r/Z .

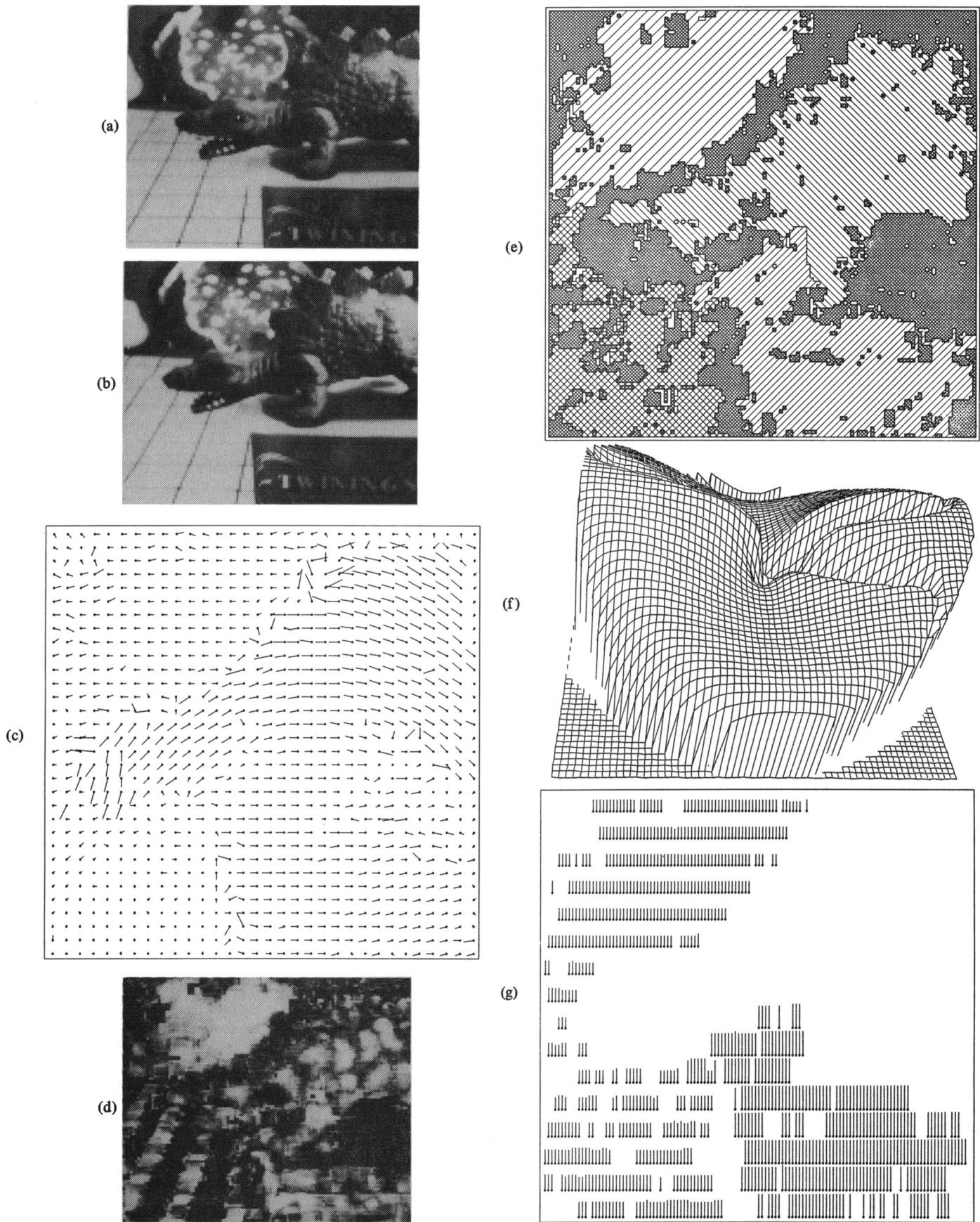


Fig. 6. Experiment 4. (a) The first intensity image. (b) The second intensity image. (c) A 32×32 sample of the computed flow field. The vectors length is scaled by 0.5. (d) The weight plane. High values are represented by bright gray levels. (e) Final segmentation. The white areas correspond to flow vectors assigned weight 0. The areas with the densest pattern correspond to unsegmented vectors. (f) The error function $\hat{\sigma}_H$, shown inverted, corresponding to the stationary environment. Note the two peaks which actually correspond to the same translation because $\hat{\sigma}_H$ is invariant to sign change in the translation vector. The range of the error function is [0.297, 2.216]. (g) The estimated depth function r/Z .

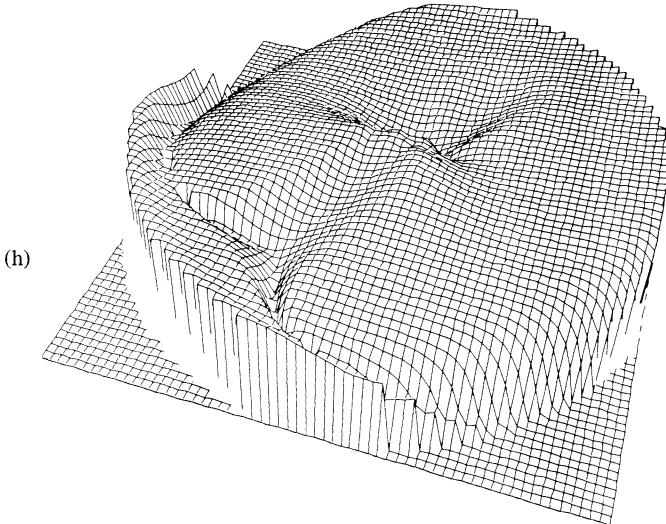


Fig. 6. (Continued.) (h) The error function, shown inverted, corresponding to the independently moving object. The range of this function is [0.241, 0.707].

about an axis parallel to the Z axis of the camera. The flow field [Fig. 6(c)] and a related confidence measure [Fig. 6(d)] were computed employing a technique developed by Anandan [5].

Four segments [Fig. 6(e)] were found in the first phase of the interpretation process. A correct and unique grouping of the segments into objects was then determined. An error function, based on the three segments corresponding to the stationary environment, is displayed in Fig. 6(f). Minimizing this function, the translation axis and the rotation parameters of the camera were determined to be $(0.9996, -0.0258, -0.0059)$ and $(-0.05^\circ, -1.68^\circ, 0.61^\circ)$, respectively, in a reasonable agreement with the actual values. In Fig. 6(g), which shows the "reciprocal depth" map r/Z , three distinct surfaces can be detected. Two of these surfaces are roughly parallel to the image plane, and the third one is slanted; these orientations are consistent with the actual environment.

The error function associated with the independently moving object is displayed in Fig. 6(h). As in experiment 2, the translation axis cannot reliably be determined, thus demonstrating again the instability which may exist in recovering the motion parameters.

VII. SUMMARY

We have presented a new approach for the interpretation of optical flow fields which are induced by motion of the camera as well as motion of several rigid objects in the environment. The interpretation goals of decomposing the flow field into sets corresponding to independently moving objects, recovery of motion parameters, and estimation of relative depth of environmental surfaces were shown to be feasible. An algorithm based on our approach, was demonstrated to work with sparse, noise, and partially incorrect data, derived from both artificial and real images.

A hierarchical structure, based on four levels of organization in the flow field, has been employed. In the interpretation process, units from each level are combined into larger units in the next level based on their consistency with appropriate parameter values. Thus, flow vectors, consistent with an affine transformation, are combined into one component; then, com-

ponents that are compatible with the same Ψ transformation are merged into a segment; and, finally, sets of segments which satisfy the same 3-D motion parameters are assumed to correspond to one rigid object. The techniques for computing the parameter values in each level have been based, whenever possible, on solving linear equations derived from the least-squares criterion. Otherwise, sampling techniques combined with multi-resolution search schemes, have been employed. Combining all these techniques, an effective and efficient algorithm has been developed.

In some situations, however, there exists an inherent ambiguity in the interpretation of noisy flow fields, as was briefly discussed and demonstrated in Sections V and VI. In [4], we analyze such cases and present constraints and parameters which can be extracted even in unstable situations. Integration of such information over a time sequence of flow fields may eventually resolve the ambiguity and result in a unique interpretation.

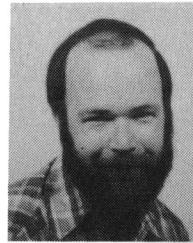
ACKNOWLEDGMENT

I am grateful to many members of the Computer Vision Group at the University of Massachusetts for their support and generous help. A. Hanson, E. Riseman, P. Anandan, and the reviewers made useful comments for improving this paper.

REFERENCES

- [1] G. Adiv, "Recovering 2-D motion parameters in scenes containing multiple moving objects," in *Proc. DARPA Image Understanding Workshop*, Arlington, VA, 1983, pp. 285-292 (also, TR 83-11, Dep. Comput. Inform. Sci., Univ. Mass., Amherst).
- [2] —, "Recovering motion parameters in scenes containing multiple moving objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Washington, DC, 1983, pp. 399-400.
- [3] —, "Determining 3-D motion and structure from optical flow generated by several moving objects," in *Proc. DARPA Image Understanding Workshop*, New Orleans, LA, 1984, pp. 113-129 (also, TR 84-07, Dep. Comput. Inform. Sci., Univ. Mass., Amherst).
- [4] —, "Interpreting optical flow," Ph.D. dissertation (in preparation), Dep. Comput. Inform. Sci., Univ. Mass., Amherst, 1985.
- [5] P. Anandan, "Computing dense displacement fields with confidence measures in scenes containing occlusion," in *Proc. DARPA Image Understanding Workshop*, New Orleans, LA, 1984, pp. 236-246.
- [6] D. H. Ballard, "Parameter networks: Towards a theory of low-level vision," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, Canada, 1981, pp. 1068-1078.
- [7] D. H. Ballard and O. A. Kimball, "Rigid body motion from depth and optical flow," Dep. Comput. Sci., Univ. Rochester, Rochester, NY, TR 70,1981.
- [8] A. R. Bruss and B. K. P. Horn, "Passive navigation," Mass. Inst. Technol., A.I. Memo 662, 1981.
- [9] J.-Q. Fang and T. S. Huang, "Solving three-dimensional small-rotation motion equations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Washington, DC, 1983, pp. 253-258.
- [10] —, "Estimating 3-D movement of a rigid object: Experimental results," in *Proc. Int. Joint Conf. Artif. Intell.*, Karlsruhe, West Germany, 1983, pp. 1035-1037.
- [11] C. L. Fennema and W. B. Thompson, "Velocity determination in scenes containing several moving objects," *Comput. Graph. Image Processing*, vol. 9, pp. 301-315, Apr. 1979.
- [12] A. Hanson and E. Riseman, Eds., *Computer Vision Systems*. New York: Academic, 1978, pp. 303-334.
- [13] C. Jerian and R. Jain, "Determining motion parameters for scenes with translation and rotation," in *Proc. Workshop Motion: Representation Perception*, Toronto, Canada, 1983, pp. 71-77.
- [14] D. T. Lawton, "Motion analysis via local translational processing," in *Proc. Workshop Comput. Vis.*, Rindge, NH, 1982, pp. 59-72.

- [15] —, "Processing dynamic image sequences from a moving sensor," Ph.D. dissertation, Dep. Comput. Inform. Sci., Univ. Mass., Amherst, TR 84-05, 1984.
- [16] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," in *Proc. Roy. Soc. London*, B, vol. 208, July 1980, pp. 385-397.
- [17] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133-135, Sept. 1981.
- [18] H. H. Nagel, "On the derivation of 3-D rigid point configurations from image sequences," in *Proc. IEEE Conf. Pattern Recog. Image Proc.*, Dallas, TX, 1981, pp. 103-108.
- [19] H. H. Nagel and B. Neumann, "On 3-D reconstruction from two perspective views," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, Canada, 1981, pp. 661-663.
- [20] B. Neumann, "Motion analysis of image sequences for object grouping and reconstruction," in *Proc. Int. Conf. Pattern Recog.*, Miami, FL, 1980, pp. 1262-1265.
- [21] J. O'Rourke, "Motion detection using Hough techniques," in *Proc. IEEE Conf. Pattern Recog. Image Proc.*, Dallas, TX, 1981, pp. 82-87.
- [22] K. Prazdny, "Egomotion and Relative depth map from optical flow," *Biol. Cybern.*, vol. 36, pp. 87-102, 1980.
- [23] —, "Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer," in *Proc. IEEE Conf. Pattern Recog. Image Proc.*, Dallas, TX, 1981, pp. 109-114.
- [24] J. H. Rieger and D. T. Lawton, "Determining the instantaneous axis of translation from optic flow generated by arbitrary sensor motion," in *Proc. Workshop Motion: Representation Perception*, Toronto, Canada, 1983, pp. 33-41.
- [25] J. W. Roach and J. K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 554-562, Nov. 1980.
- [26] K. R. Sloan, "Dynamically quantized pyramids," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, Canada, 1981, pp. 734-736.
- [27] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 13-27, Jan. 1984.
- [28] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: M.I.T. Press, 1979.
- [29] —, "Analysis of visual motion by biological and computer systems," *Computer*, vol. 14, pp. 57-69, Aug. 1981.
- [30] A. M. Waxman and S. Ullman, "Surface structure and 3-D motion from image flow: A kinematic approach," Center Automation Res., Univ. Maryland, CAR-TR-24, 1983.
- [31] A. M. Waxman, "An image flow paradigm," in *Proc. Workshop Comput. Vis.*, Annapolis, MD, 1984, pp. 49-57.
- [32] A. M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation and global image flow: Planar surfaces in motion," Center Automation Res., Univ. Maryland, CAR-TR-58, 1984.
- [33] T. D. Williams, "Computer interpretation of a dynamic image from a moving vehicle," Ph.D. dissertation, Dep. Comput. Inform. Sci., Univ. Mass., TR 81-22, 1981.



Gilad Adiv was born in Israel, in 1951. He received the B.S. degree in mathematics and statistics in 1971, and the M.S. degree in mathematics in 1973, both from the Hebrew University of Jerusalem, Israel. In 1983 he received the M.S. degree in computer and information science from the University of Massachusetts, Amherst, where he is currently pursuing the Ph.D. degree.

From 1973 to 1981 he was employed in Rafael, Israel. His major research interests lie in computer vision, especially motion analysis, and image processing.