Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects

Gilad Adiv

COINS Technical Report 84–07*

April 1984

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 segmented into connected sets of flow vectors, where each set is consistent with a rigid motion of a roughly planar surface. In the second stage sets of segments are hypothesized to be induced by the same rigidly moving object. Each of these hypotheses is tested by searching for 3-D motion parameters which are compatible with all the segments in the corresponding set. Once the motion parameters are recovered, the relative environmental depth can be estimated as well. Experiments based on real and simulated data are presented.

This work was supported by DARPA under Grant N00014-82-K-0464.

* The report is sponsored by Allen R. Hanson and Edward M. Riseman.

Table of Contents

1. 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 dynamic segmentation, recovering the motion parameters of the sensor and each moving object, and structure determination. The results of this interpretation can be used to control behaviour, as in robotics or navigation. They can also be integrated, as an additional knowledge source, into an image understanding system, such as the VISIONS sys.em [HAN78].

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. A new scheme is proposed, which allows motion of the camera as well as rigid objects in the scene. Furthermore, the flow field is allowed to be sparse, noisy and partially incorrect. The information in only one flow field, as opposed to a time sequence of such fields, is utilized.

Our approach is based on two main stages. In the first stage 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. In the second stage sets of segments are hypothesized to be induced by the same rigidly moving object. Each of these hypotheses is tested by searching for 3-D motion parameters which are compatible with all the segments in the corresponding set. Once the motion parameters are recovered, the relative environmental depth can be estimated as well.

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 3. In subsequent sections, algorithms for flow field segmentation, estimation of motion parameters, and structure determination are developed. Preliminary experiments based on real and simulated data are described in section 6.

2. Literature Review

In this section we review methods existing in the literature for interpreting optical flow fields. We concentrate on techniques which assume rigid motion and basically rely on the information contained in one flow field. Two main issues are emphasised:

a) **Scene Complexity**. Some researchers assume that the scene contains only one object, or, equivalently, that the sensor is moving but the environment is stationary (e.g., [BRU81], [LAW82], [TSA84]). Others allow the scene to contain several independently moving objects (e.g., [ULL79], [NEU80]).

b) **Robustness.** Optical flow fields produced from real images by existing techniques are noisy and partially incorrect (see the discussion in [ULL81]). Many of the algorithms described in the literature for interpretation of flow fields fail under such conditions. Other algorithms are less sensitive and work reasonably well on real world images.

In the first class of techniques, discussed in this review, only one rigid object (or camera motion) is assumed. A few researchers [ROA80, PRA80, NAG81a,b, FAN83a,b] present sets of nonlinear equations with motion parameters as unknowns. Methods for solving such equations are usually iterative and require initial guesses of the unknowns. Sensitivity to noise is indicated by experiments reported in [ROA80, PRA80, FAN83a,b].

Longuet-Higgins [LON81] and Tsai and Huang [TSA84] develop 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 still reported [TSA84].

Bruss and Horn [BRU81] employ a least squares approach which minimises some measure of the discrepancy between the measured flow and that predicted from the computed motion parameters. In the case of general rigid motion this approach leads to a system of nonlinear equations from which the motion parameters can be computed numerically.

This method is computationally more complicated than the methods offered in [LON81] and [TSA84], but seems to be more robust in the presence of noise.

Assuming a purely translational motion, all the flow vectors are oriented towards 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 [WIL81]. 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 [LAW82] 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. Results for other restricted cases of motion are presented in [LAW84].

Prazdny [PRA81] 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 3 rotation parameters is obtained and the solution can be determined by minimizing this function. Jerian and Jain [JER83] report on difficulties with applying a similar approach to noisy data.

Rieger and Lawton [RIE83] develop a relatively robust and simple procedure for computing the motion parameters, based on the fact that the differences between optic flow vectors near occlusion boundaries are oriented towards the FOE of the translational field. However, the environment is assumed to contain occlusion boundaries which endow the flow field with strong discontinuities.

A number of methods, presented in the literature, allow (at least in principle) unconstrained sensor motion and independently moving objects in the environment. Ullman, in his somewhat pioneer work [ULL79], 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. This approach seems to be very sensitive to noise because of its local nature.

Longuet-Higgins and Prazdny [LON80] and Waxman and Ullman [WAX83] 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. However, 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.

More global approaches are proposed in [NEU80] and [BAL81b]. Neumann [NEU80] 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 heavily relies on the assumption of-orthographic projection.

Ballard and Kimball [BAL81b] apply the generalized Hough technique to the optical flow field and thus extract the motion parameters. This is a global approach which is relatively insensitive to noise. In principle, it can also be used in scenes containing independently moving objects. However, the depth information is assumed to be known, thus making the task much easier.

This review demonstrates typical constraints and weaknesses of algorithms reported in the literature. No algorithm for interpretation of optical flow fields in scenes containing several, independently moving, rigid objects, has been shown to work with noisy, real world data, unless severe constraints are assumed or additional information is utilized.

3. The Model and the Task — A Mathematical Formulation

3.1 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 model 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 figure 3.1) 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:

$$x = X/Z, \qquad y = Y/Z. \tag{3.1a,b}$$

The motion, relative to the camera, of a rigid object in the scene can be decomposed into two components: translation $\underline{T} = (T_X, T_Y, T_Z)$ and rotation $\underline{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$. 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 [LON80]:

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

and

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

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

$$(\alpha,\beta) = (\alpha_R,\beta_R) + (\alpha_T,\beta_T), \tag{3.3}$$

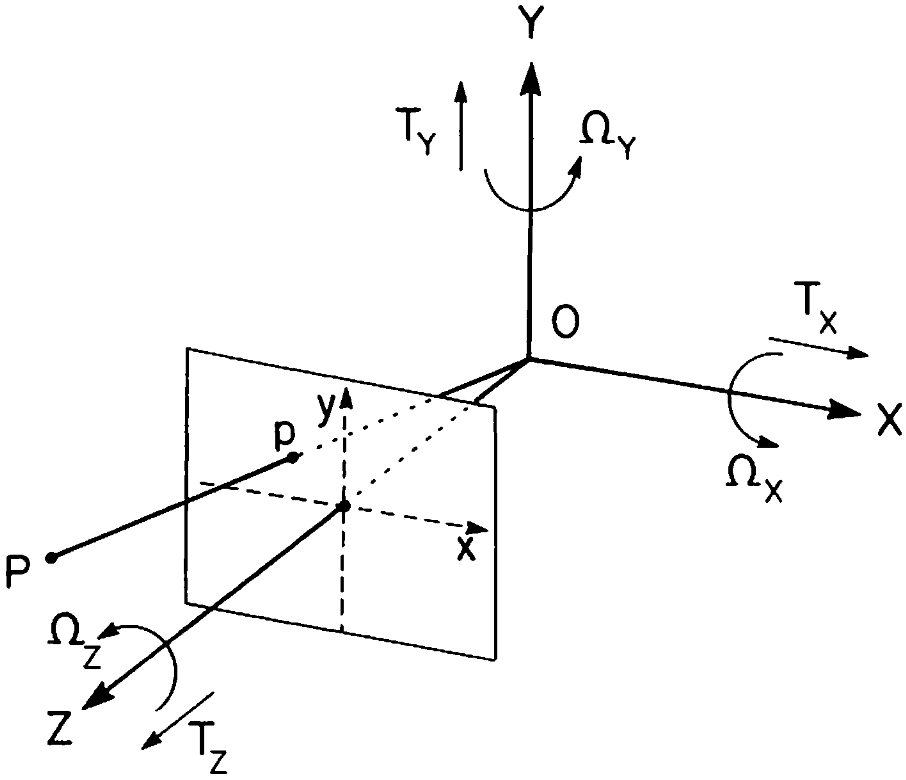

Figure 3.1(redrawn from [LON80]): A coordinate system (X, Y, Z) attached to the camera, and the corresponding image coordinates (x, y). The image position p is the perspective projection of the point P in the environment. $\underline{T} = (T_X, T_Y, T_Z)$ and $\underline{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ represent the relative translation and rotation of a given object in the scene.

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

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

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

In the displacement-based scheme, let (X, Y, Z) be the camera coordinates at time t_1

of a point on the object and let (X', Y', Z') be the corresponding coordinates at time t_2.

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \underline{T}, \tag{3.5}$$

where the rotation matrix R can be approximated, assuming small values of the rotation parameters, by:

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

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

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

and

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

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

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

and

$$\beta = \frac{-\Omega_X(1 + y^2) + \Omega_Y xy + \Omega_Z x + (T_Y - T_Z y)/Z}{1 + \Omega_X y - \Omega_Y x + T_Z/Z}. \tag{3.8b}$$

If $|T_Z/Z| \ll 1$ and the field of view of the camara, 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 equations (3.2).

To conclude: equations (3.2) hold not only for velocity fields, but also for displacement fields, given that the rotation parameters are small and that the Z-component of the translation is small relative to the distance of the object from the image plane. Such assumptions 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 equations (3.2) as the basis of our analysis.

3.2 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 want 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 5 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 $\underline{U} = \underline{T}/r$, where r is the length of the translation vector $\underline{T}$. 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.

4. Segmentation

In this section we develop a method for segmentation of 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 correspond to a portion of only one rigid object. Thus, the data is organized into coherent units which form the basis for further processing. Another purpose of the segmentation is exclusion of flow vectors which are inconsistent with their neighbors, hence assumed to be incorrect.

4.1 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

$$k_1X + k_2Y + k_3Z = 1. \tag{4.1}$$

The coefficients k_1, k_2 and k_3 can be any real numbers, except the case in which all of them are zero. Using (3.1), we obtain:

$$1/Z = k_1x + k_2y + k_3. \tag{4.2}$$

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

$$\alpha = a_1 + a_2x + a_3y + a_7x^2 + a_8xy, \tag{4.3a}$$

$$\beta = a_4 + a_5x + a_6y + a_7xy + a_8y^2, \tag{4.3b}$$

where:

$$a_1 = \Omega_Y + k_3T_X, \tag{4.4a}$$

$$a_2 = k_1T_X - k_3T_Z, \tag{4.4b}$$

$$a_3 = -\Omega_Z + k_2T_X, \tag{4.4c}$$

$$a_4 = -\Omega_X + k_3T_Y, \tag{4.4d}$$

$$a_5 = \Omega_Z + k_1T_Y, \tag{4.4e}$$

$$a_6 = k_2T_Y - k_3T_Z, \tag{4.4f}$$

$$a_7 = \Omega_Y - k_1T_Z \tag{4.4g}$$

and

$$a_8 = -\Omega_X - k_2 T_Z. \tag{4.4h}$$

Equations (4.3) represent what we shall call a Ψ *transformation*. This is a 2-D transformation of the image into itself based on the 8 parameters $a_1, \ldots, a_8$.

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

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

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

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

and

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

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 equation (4.5), then the flow field in R can be approximated by a Ψ transformation.

In a real world environment the surface can be usually approximated by a piecewise planar surface, containing only a few planar patches, where 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, where each set approximately satisfies 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. In the next section we describe an algorithm for achieving such a segmentation.

4.2 Segmentation Algorithm

The generalized Hough transform technique [BAL81a] is a useful tool for grouping together flow vectors which satisfy the same 2-D parameterized transformation [ADI83]. 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 8-parameter Ψ transformations (equations (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

$$\alpha = a_1 + a_2x + a_3y \qquad (4.7a)$$

and

$$\beta = a_4 + a_5x + a_6y, \qquad (4.7b)$$

are sub-class of the Ψ transformations, parameterised by only 6 parameters. Furthermore, these parameters can be partitioned into two disjoint sets of 3 parameters each, corresponding to equations (4.7a) and (4.7b). Thus, the grouping problem in the first stage can be basically solved by applying the Hough technique to 3-dimensional parameter spaces, as will be shown in sub-section 4.2.1.

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 then obtained. If these error values are not high (in a sense defined in sub-section 4.2.2), then the components are merged.

Sometimes over-fragmentation may occur in the first stage of the segmentation, that is, a segment is partitioned into a large number of small components, as demonstrated in experiment 1 in section 6.1 (see figure 6.1b). 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 sub-sections the three stages of the segmentation are more fully described, but, for the sake of brevity, many details are still suppressed.

4.2.1 First Stage — Grouping Based on Affine Transformations

4.2.1.1 A Modified Version of the Generalised Hough Technique

The grouping of flow vectors into components consistent with affine transformations is based on a modification of the generalised Hough technique. The affine transformations can be represented by a 6-dimensional parameter space where each dimension corresponds to one of the parameters $a_1, \ldots, a_6$ in equations (4.7). For computational reasons the parameter space should 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 obtained sets.

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

$$\delta \overset{\text{def}}{=} \sqrt{\delta_x^2 + \delta_y^2} \leq \epsilon, \tag{4.8a}$$

where

$$\delta_x = |\alpha - a_1 - a_2 x - a_3 y| \tag{4.8b}$$

and

$$\delta_y = |\beta - a_4 - a_5 x - a_6 y|. \tag{4.8c}$$

Note that ϵ is a function of the resolution in the parameter space and the noise level in the flow field, but it is never less than a given threshold, typically one pixel. In this case, the amount of support is determined by the function

$$V(a_1, a_2, a_3, a_4, a_5, a_6, x, y) = 1 - 0.75\delta/\epsilon \tag{4.9}$$

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 $(a_1, \ldots, a_6)$, is the weighted sum

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

where $W(x, y)$ is the weight of the flow vector in 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 equation (4.10), given to any of those transformations. A serious computational problem may arise if the number of points in the parameter space is very high. If, for example, the minimal and maximal possible values of the parameter a_1 are -64 pixels and $+64$ pixels, respectively, and the desired accuracy is 0.25 pixel, then 512 samples are apparently needed for this parameter. If an equal number of samples is also required for the other parameters, then the parameter space should contain $512^6 \approx 16 \times 10^{15}$ points. In such a case, a straightforward Hough technique is computationally impractical.

This problem is alleviated by using two techniques. First, a multi-resolution scheme in the parameter space is employed. The Hough technique is iteratively used, where in each iteration the parameter space is quantised 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 [ORO81, SLO81].

The second technique is based on 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-dimensional parameter spaces, using the relevant constraint, (4.7a) or (4.7b). Sets of highly supported parameter triples, $A_\alpha = \{(a_{1_i}, a_{2_i}, a_{3_i}) : i = 1, \ldots, N\}$ and $A_\beta = \{(a_{4_i}, a_{5_i}, a_{6_i}) : i = 1, \ldots, N\}$, are thus found, where N was experimentally determined to be 10. As a result, a set of N^2 hypothesized affine transformations,

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

is obtained. The support function can be then 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-dimensional parameter space. However, large components in the flow field, corresponding to maxima points in the 6-dimensional space, can be expected to produce maxima points also in each of the 3-dimensional 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 multi-resolution scheme; together they create a very efficient algorithm.

4.2.1.2 Implementation of a Multipass Approach

The components which we try to locate are connected sets of flow vectors which support the same affine transformation. The algorithm for obtaining this goal is based on a multipass Hough approach, where a basic cycle of operations is repeatedly executed [FEN79, ADI83]. The input to each cycle includes masks of the components which were already detected during the previous cycles and a mask of those vectors which were excluded from further consideration. The cycle is composed of the following steps:

1) Consider the set of pixels which are assigned a positive weight, do not belong to any of the previously found components and were not excluded from further consideration. Find in this set a connected subset E with maximal sum of weights. If this sum is below a given threshold L, which is related to the noise level in the flow field, then stop searching for new components and start the merging stage. Sometimes over-fragmentation occurs, i.e., a segment is partitioned into a large number of small components . In order to prevent an excessive number of cycles in such a case, a new threshold, higher than L, is determined and the process is stopped if the sum of weights is below this threshold. The grouping of vectors in small sets is thus postponed to the third stage.

2) Partition the set E into a given number (typically 64) of square windows, such that the sum of weights in each window is roughly 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 E, in order to reduce the computation time.

3) Use the modified Hough technique, described in section 4.2.1.1, to find the affine transformation which receives the maximal support from the flow vectors selected in the previous step.

4) Determine the set F of all the vectors in E which are consistent with the computed affine transformation. If the sum of weights corrsponding to F is below the threshold L, then exclude the set E from further consideration and start a new cycle. Otherwise, find in F a connected subset G with maximal sum of weights. Then, if this sum exceeds L, add G to the list of components; otherwise, just exclude G from further consideration (to prevent an infinite loop).

4.2.2 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. Two main steps can be observed in the merging process. In the first step an optimal Ψ transformation is estimated for each component, employing the least squares technique. If the component contains n flow vectors, then the error function to be minimized is

$$E(a_1,\ldots,a_8)=\sum_{i=1}^{n}Wi\Big[(\alpha_i-a_1-a_2x_i-a_3y_i-a_7x_i^2-a_8x_iy_i)^2 \\ +(\beta_i-a_4-a_5x_i-a_6y_i-a_7x_iy_i-a_8y_i^2)^2\Big], \qquad (4.12)$$

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 $a_1,\ldots,a_8$ and equating to 0, a set of linear equations is obtained. Their solution, $a_1^*,\ldots,a_8^*$, is the optimal Ψ transformation. Substituting this solution in (4.12) and using the normalization equation

$$\sigma=\sqrt{E(a_1^*,\ldots,a_8^*)\Big/\sum_{i=1}^{n}Wi}, \qquad (4.13)$$

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

In the second step Ψ transformations, corresponding to merged sets of adjacent components, are computed. Based on related error values, associated with each component in such a set, it is decided whether to merge the components. In order to formulate the conditions for a merging decision, some notations are used. First, S denotes a set of adjacent components and Ψ_S is the corresponding optimal Ψ transformation. For each component C_j in S, σ_j is the error value found in the first step of the merging process, σ_j' is the error value obtained by substituting the coefficients of Ψ_S in (4.13), and p_j is the ratio between the sum of vector weights in C_j and the total sum of weights in the set S.

The ratios $\{\sigma_j'/\sigma_j\}$ are a major factor in the merging decision. If these ratios are only slightly higher than 1, then a merging decision seems to be justified. Note that σ_j' is never less than σ_j, because the optimal Ψ transformation corresponding to the component C_j can be adjusted to the local surface and noise associated with C_j. If, however, p_j is close to 1, then we expect σ_j' to be very close to σ_j, especially when a merging decision is justified. Hence, the allowed level of σ_j'/σ_j will be defined to be a monotonically decreasing function of p_j:

$$L_a(p_j) = \tau_1 - (\tau_1 - 1)p_j, \tag{4.14}$$

where τ_1 is a given threshold (typically $\tau_1 \approx 1.5$). Thus, $L_a(p_j)$ ranges from values close to 1 for components with relatively large weight, up to almost τ_1 for small components.

Sometimes, however, σ_j can not be computed because the linear equations derived from (4.12) are linearly dependent. In addition, if the component C_j is small, then σ_j may be unreliable as a basis for evaluation of σ'_j values. Therefore, an absolute threshold $L_b(p_j)$ of allowed values of σ'_j is also employed. $L_b(p_j)$ is given by

$$L_b(p_j) = \tau_3 - (\tau_3 - \tau_2)p_j, \tag{4.15}$$

where τ_2 and τ_3 are pre-determined thresholds related to the expected noise level in the flow field and $\tau_3 > \tau_2$. Thus $L_b(p_j)$ ranges from τ_2 for very large components up to τ_3 for small components. The reason for this dependency on p_j is related to the effect of statistical averaging of the noise. In large objects, such averaging is likely to take place and thus τ_2 represents the estimated standard deviation of the noise. The threshold τ_3, on the other hand, represents some reasonable upper bound of 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 τ_2 will be taken to be the corresponding standrad deviation, that is, approximately 0.3 pixels and τ_3 will be 0.5 pixels. To conclude, a merging decision is accepted if and only if, for each component C_j in S, $\sigma'_j/\sigma_j \leq L_a(p_j)$ or $\sigma'_j \leq L_b(p_j)$, i.e.,

$$\sigma'_j \leq \max\{L_a(p_j)\sigma_j,\ L_b(p_j)\}. \tag{4.16}$$

The algorithm, for finding sets of components to be merged, starts with detection of the component with the maximal sum of vector weights. Then, merging of this component with its neighbors is sequentially tested, in the order of their associated sums of weights. If one of these merging trials is successful, then merging of additional components with the already merged pair is examined. In general, given a set of already merged components, neighboring components are tested as candidates for adjoining this set. This process continues until all the candidates for merging are examined. Then, the process starts again, considering only the components which are not yet assigned to any of the already created segments. Evantually, all the components are contained in one of the segments.

4.2.3 Third Stage

The purpose of the third stage of the segmentation is examination of flow vectors which were assigned positive weights and were not grouped into any of the components in the first stage of the segmentation, and thus 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.

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.

5. Forming Object Hypotheses and Recovering 3-D Motion and Structure

In the first stage 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 detect sets of segments which are consistent with the same 3-D motion parameters. Such a set can be hypothesized, employing the rigidity assumption [ULL79], to be induced by one rigidly moving object (or by the camera motion). In sub-section 5.1 we describe an algorithm for computing the motion parameters of any set of flow vectors generated by a rigid motion. In section 5.2

we combine this algorithm with the segmentation results to form object hypotheses and estimate the corresponding 3-D motion and structure. Again, for the sake of brevity, many details are suppressed.

5.1 Estimating Motion Parameters and Depth Information of a Rigid Object

5.1.1 Optimisation Constraint

Given a set of flow vectors, assumed to be induced by a rigidly moving object, we want to find the 3-D motion parameters which are maximally consistent with this data. Following [BRU81], we employ the least-squares approach because of its relative robustness in the presence of noise. Based on (3.2), the error function to be minimised is

$$\sum_{i=1}^{n} W_i\Big[\Big(\alpha_i + \Omega_X x_i y_i - \Omega_Y(1+x_i^2) + \Omega_Z y_i - (T_X - T_Z x)/Z_i\Big)^2 + \Big(\beta_i + \Omega_X(1+y_i^2) - \Omega_Y x_i y_i - \Omega_Z x_i - (T_Y - T_Z y)/Z_i\Big)^2\Big], \tag{5.1}$$

where $\underline{T} = (T_X, T_Y, T_Z)$ and $\underline{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ 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 $\underline{T}$, $\underline{\Omega}$ 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 equations (3.4)), the error function can be more concisely represented by

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

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

$$(U_X, U_Y, U_Z) = (T_X, T_Y, T_Z)/r, \tag{5.3}$$

and the relative depth values, represented by

$$\tilde{Z}_i = r/Z_i, \quad i = 1, \ldots, n. \tag{5.4}$$

Introducing the abbreviations

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

and

$$\beta_U = U_Y - U_Z y = \beta_T/\tilde{Z}, \tag{5.5b}$$

(5.2) can be rewritten as

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

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

$$\tilde{Z}_i \geq 0, \quad i = 1, \ldots, n, \tag{5.7}$$

should be satisfied. Note that this error measure is different from the one employed in [BRU81] 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

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

Setting it equal to 0 yields

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

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 non-negative for non-negative values of $\tilde{Z}_i$ and, therefore, the error function is monotonically non-decreasing for these values. To summarise, the optimal value of $\tilde{Z}_i$ is given by:

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

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:

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

where

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

A normalized version of this error function, defined by

$$\sigma(\underline{U}, \underline{\Omega}) = \sqrt{E(\underline{U}, \underline{\Omega}) \Big/ \sum_{i=1}^{n} Wi}, \tag{5.12}$$

will be also 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 non-zero translation. In the case of a purely rotational motion, the appropriate error function to be minimized is:

$$E_R(\underline{\Omega}) = \sum_{i=1}^{n} Wi\Big((\alpha_i - \alpha_{Ri})^2 + (\beta_i - \beta_{Ri})^2\Big). \tag{5.13}$$

If the minimal value of this function is close to the minimal value of the function (5.11), then the motion is, possibly, purely rotational.

The task of finding the rotation parameters which minimize the function $E_R(\underline{\Omega})$ can be easily accomplished by solving a set of three linear equations [BRU81] and will not be considered further in this paper. Thus, in the next section, we concentrate on the much more difficult task of finding values of $\underline{U}$ and $\underline{\Omega}$ which minimise the error function $E(\underline{U},\underline{\Omega})$ (or, equivalently, the function $\sigma(\underline{U},\underline{\Omega})$), where $\underline{U}$ can be any unit vector and $\underline{\Omega}$ is unconstrained.

5.1.2 Algorithm

The algorithm for recovering the motion parameters employs an error measure, derived from (5.12), corresponding to possible directions of the translation vector. A minimum value of this function is determined, using a multi-resolution sampling scheme.

Let us start the derivation of this error measure with the observation that if the depth constraints (5.7) are ignored, then, for any hypothesized direction of translation, the optimal rotation parameters can be easily extracted by solving a set of three linear equations. To see that, notice that the error function (5.11) can be reduced in this case to the function

$$E'(\underline{U},\underline{\Omega}) = \sum_{i=1}^{n} Wi\left[\left((\alpha_i - \alpha_{Ri})\beta_{Ui} - (\beta_i - \beta_{Ri})\alpha_{Ui}\right)^2 / (\alpha_{Ui}^2 + \beta_{Ui}^2)\right]. \qquad (5.14)$$

Differentiating $E'(\underline{U},\underline{\Omega})$ with respect to the rotation parameters and setting the derivatives equal to 0 yields three linear equations with the rotation parameters as unknowns. Thus, ignoring the depth constraints (5.7), the search space can be limited to the unit sphere $\{\underline{U} : |\underline{U}| = 1\}$.

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

$$HS = \{\underline{U} : |\underline{U}| = 1 \text{ and } U_Z \geq 0\}. \qquad (5.15)$$

The preferred sign of $\underline{U}$ can be then determined, as proposed in [BRU81], as the one which gives $\tilde{Z}_i \geq 0$ for most indices i. Still, we wish to incorporate these constraints or, equivalently, the equations (5.11b) in a more rigorous way. Hence, for each $\underline{U}$ in HS, we define the error measure

$$\sigma_1(\underline{U}) = \min_{B,\underline{\Omega}} \sigma(B\underline{U}, \underline{\Omega}), \tag{5.16}$$

where B can have the values $+1$ or -1. The goal is to find a vector $\underline{U}$ in HS which minimizes the function σ_1. The associated values of B and $\underline{\Omega}$ are, respectively, the determined sign of the translation vector and the estimated rotation parameters. The function σ_1 is, however, difficult to compute. Therefore, in the proposed algorithm we compute an approximation to σ_1 which is experimentally shown to be very accurate. A few main steps can be distinguished in the procedure for computing this approximation:

1) Given a vector $\underline{U}$ in HS, estimate the optimal rotation vector $\underline{\Omega}^*$ by minimising $E'(\underline{U}, \underline{\Omega})$ with respect to $\underline{\Omega}$, and compute the corresponding normalised error measure $\sigma'(\underline{U}, \underline{\Omega}^*)$. This error value is a lower bound of $\sigma_1(\underline{U})$ since it minimizes the error function $\sigma(\underline{U}, \underline{\Omega})$, with respect to $\underline{\Omega}$ and the sign of $\underline{U}$, without considering the depth constraints (5.7).

2) Compute $\sigma(\underline{U}, \underline{\Omega}^*)$ and $\sigma(-\underline{U}, \underline{\Omega}^*)$. Determining the minimum of these two error values yields the preferred sign, denoted by μ, of $\underline{U}$. Using the notation $\underline{U}^* = \mu\underline{U}$, $\sigma(\underline{U}^*, \underline{\Omega}^*)$ is an upper bound of $\sigma_1(\underline{U})$, because it gives the actual error measure for some values of B and $\underline{\Omega}$ in equation (5.16).

3) Compute an approximation to $\sigma_1(\underline{U})$ by averaging its lower and upper bounds:

$$\hat{\sigma}_1(\underline{U}) = \left(\sigma'(\underline{U}, \underline{\Omega}^*) + \sigma(\underline{U}^*, \underline{\Omega}^*)\right)/2. \tag{5.17}$$

The relative deviation of $\hat{\sigma}_1(\underline{U})$ from $\sigma_1(\underline{U})$ is bounded by

$$\left(\sigma(\underline{U}^*, \underline{\Omega}^*) - \sigma'(\underline{U}, \underline{\Omega}^*)\right) / \left(2\hat{\sigma}_1(\underline{U})\right). \tag{5.18}$$

In the experiments, this value was found to be very small, typically much less than 0.01.

The search for an optimal vector in HS consists of a sampling (similar to [LAW82]) of the error measure $\hat{\sigma}_1$. A multi-resolution scheme is employed, where in the first iteration the set HS is coarsly sampled 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 HS require a vector $\underline{U}'$ to be defined as a 'neighbor' of a vector $\underline{U}$ if either $\underline{U}'$ or $-\underline{U}'$ is close to $\underline{U}$. 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 $\hat{\sigma}_1$ to the whole unit sphere, employing exactly the same definition used for the domain HS. In this case, $\hat{\sigma}_1(-\underline{U}) = \hat{\sigma}_1(\underline{U})$ for each unit vector $\underline{U}$, thus, computationally, it makes no difference which domain of definition is used.

The final solution of $\underline{U}$, and the corresponding sign μ and the rotation parameters $\underline{\Omega}^*$, defined in the procedure for computing $\hat{\sigma}_1$, are the determined motion parameters. Substituting these parameters in equations (5.10), the relative depth, corresponding to each flow vector, can be estimated as well.

We should mention that sometimes the error function $\hat{\sigma}_1$ is very close to its minimal value in a large portion of the search space (see figure 6.2e). Hence, in the presence of noise, it may be impossible to obtain reasonably accurate estimates of the motion parameters. Two complementary approaches may be taken in order to deal with this ambiguity. First, *constraints* on the motion parameters and the environmental depth, rather than values, can be still recovered, using, for example, the coefficients of the related Ψ transformations (see equations (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 HS, using the computed values of $\hat{\sigma}_1$. Investigation of situations which may lead to this ambiguity is under way.

5.2 Forming Object Hypotheses

Segments of the flow field, which are consistent with the same motion parameters, can be hypothesized, using the rigidity assumption [ULL79], 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 M_i are computed for each segment SEG_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 SEG_i in the set, an error measure M_i'

is obtained by substituting these parameters and the related flow data in equation (5.17). Based on the error values $\{M_i\}$ and $\{M_i'\}$, consistency of the set with rigid motion is determined, employing a decision procedure similar to the one described in section 4.2.2.

Actually, each segment is sampled, using the method in step (2) of the multipass Hough technique (section 4.2.1.2), 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, a flow field which can be interpreted as resulting from one rigidly moving object. In order to deal with this ambiguity, one may have to find a *set* of possible decompositions, not only one. Analysis of this ambiguity is also under way.

6. 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 real data. In all the experiments, values to appear in translation vectors and surface equations are given in focal units, whereas rotation parameters are given in radians 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 $-1/2$ and $+1/2$ 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.

6.1 Experiment 1

The first experiment simulates a translatory motion of the camera, represented by the vectors $\underline{T}_C = (0., 0.02, 1.)$ and $\underline{\Omega}_C = (0., 0., 0.)$. 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 figure 6.1a.

The results of the three stages of the segmentation, shown in figures 6.1b, 6.1c and 6.1d, 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 $\hat{\sigma}_1$ (equation 5.17) was computed using 64 vectors from each segment. Employing a spherical coordinate system (r, ϕ, θ), where

$$X = r\sin(\phi)\cos(\theta), \tag{6.1a}$$

$$Y = r\sin(\phi)\sin(\theta) \tag{6.1b}$$

and

$$Z = r\cos(\phi), \tag{6.1c}$$

the domain of definition of $\hat{\sigma}_1$, that is, the hemisphere $\{\underline{U} : |\underline{U}| = 1,\ U_Z \geq 0\}$, can be represented by the set

$$\{(\phi, \theta) : 0 \leq \phi \leq 90°, 0° \leq \theta < 360°\}. \tag{6.2}$$

This representation is utilized for displaying the function $\hat{\sigma}_1$ in figure 6.1e, where (ϕ, θ) are used as polar coordinates. Employing the sampling procedure for minimizing $\hat{\sigma}_1$, the motion parameters were determined, after two iterations, to be $\underline{U} = (0.0017, -0.0204, -0.9998)$ and $\underline{\Omega} = (-0.0004, -0.0003, -0.0004)$. Note that, assuming a stationary environment, the camera motion is given by $-\underline{U}$ and $-\underline{\Omega}$. These results are in a good agreement with the correct values. Substituting the computed values in equation (5.10), the 'reciprocal depth' map, that is, the function r/Z shown in figure 6.1f, was obtained.

6.2 Experiment 2

In the second experiment, the camera motion is composed of both translation and rotation, described by $\underline{T}_C = (0.5, 0.5, 1.)$ and $\underline{\Omega}_C = (0.02, -0.02, 0.05)$. The environment contains 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 $\underline{T}_O = (0.5, -0.5, 0.)$ and $\underline{\Omega}_O = (0., 0., -0.2)$. 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 figure 6.2a.

The segments found in the experiment are shown in figure 6.2b. 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 $\hat{\sigma}_1$ corresponding to the stationary environment is displayed in figure 6.2c. The associated motion parameters of the camera were determined to be $-\underline{U} = (0.3897, 0.4017, 0.8287)$ (the corresponding actual values were $\underline{U}_C = (0.4082, 0.4082, 0.8164)$) and $-\underline{\Omega} = (0.0204, -0.0196, 0.0494)$. The related depth map is represented by the function r/Z in figure 6.2d.

The error function corresponding to the independently moving object is shown in figure 6.2e. This function is very close to its minimal value in a large portion of the search space, thus, demonstrating the ambiguity discussed in section 5.1.2.

6.3 Experiment 3

The third experiment, taken from [RIE83], is based on real data, shown in figures 6.3a and 6.3b. In this experiment the camera was translated roughly ir the direction of the Z-axis, between two textured cylinders, towards a textured plane parallel to the image plane, and then rotated about its Y-axis a few degrees. Figure 6.3c 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 [RIE83]. The weight assigned to each vector is 1, since no reliability measure was computed.

The results of the segmentation process are shown in figure 6.3d. The three segments found in this process are compatible with the same camera motion. Figure 6.3e displays the corresponding error function $\hat{\sigma}_1$. Assuming stationary environment, the recovered motion parameters of the camera, $-\underline{U} = (-0.0079, 0.0181, 0.9998)$ and $-\underline{\Omega} = (-0.0018, 0.0203, -0.0006)$, are consistent with the specifications of the experiment. The estimated values of r/Z, shown in figure 6.3f, have a large 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.

6.4 Experiment 4

Figures 6.4a and 6.4b are images taken from a camera translated in the direction of its X-axis. The scene mainly contains a coffee can in the front and a plant in the background. The flow field, shown in figure 6.4c, was computed using a modified version of the algorithm proposed in [GLA83]. The new version, as well as a reliability measure assigned to each flow vector, was developed by Anandan [ANA84]. Based on this reliability measure, a weight plane, shown in figure 6.4d, was computed.

The four segments in figure 6.4e were determined to be compatible with the same motion parameters. The corresponding error function is shown in figure 6.4f. The optimal motion parameters of the camera, obtained by minimising this function, are $-\underline{U} = (1., 0., 0.)$ and $-\underline{\Omega} = (0.0000, 0.0019, 0.0001)$. These results are very close to the correct ones. Figure 6.4g shows the corresponding 'reciprocal depth' map, namely, r/Z.

7. 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, noisy and partially incorrect data, derived from both artificial and real images.

An 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, components 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 hypothesized to correspond to one rigid object. The techniques for computing the parameter values in each level has 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 together, 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 5 and 6. In our future work we will characterize such situations and propose appropriate modifications of the interpretation goals, based on the ideas already mentioned in section 5. These modifications will be basically aimed at issues of representation of information which can be extracted even in ambiguous cases. Integration of such information over a time sequence of flow fields may, eventually, resolve the ambiguity and result in a unique interpretation.

Acknowledgements

I am grateful to many members of the Computer Vision group at UMASS for their support and generous help. Al Hanson, Ed Riseman and P. Anandan made useful comments for improving this paper.

References

[ADI83] Adiv, G., *Recovering 2-D Motion Parameters in Scenes Containing Multiple Moving Objects*, Proc. of DARPA Image Understanding Workshop, Arlington, VA (also TR 83-11, Computer and Information Science Dept., Univ. of Mass.) (1983).

[ANA84] Anandan, P., *A Confidence Measure for Correlation Matching*, Technical Report in preparation, Computer and Information Science Dept., Univ. of Mass. (1984).

[BAL81a] Ballard, D.H., *Parameter Networks: Towards a Theory of Low-Level Vision*, Proc. of 7th IJCAI, Vancouver, Canada (1981).

[BAL81b] Ballard, D.H. and Kimball, O.A., *Rigid Body Motion from Depth and Optical Flow*, TR70, Computer Science Dept., Univ. of Rochester (1981).

[BRU81] Bruss, A.R. and Horn, B.K.P, *Passive Navigation*, MIT A.I. Memo 662 (1981).

[FAN83a] Fang, J.-Q. and Huang, T.S., *Solving Three Dimensional Small-Rotation Motion Equations*, Proc. of CVPR, Washington, D.C. (1983).

[FAN83b] Fang, J.-Q. and Huang, T.S., *Estimating 3-D Movement of a Rigid Object: Experimental Results*, Proc. of 8th IJCAI, Karlsruhe, West Germany (1983).

[FEN79] Fennema, C.L. and Thompson, W.B., *Velocity Determination in Scenes Containing Several Moving Objects*, CGIP 9 (1979).

[GLA83] Glaser, F., Reynolds, G. and Anandan, P., *Scene Matching by Hierarchical Correlation*, Proc. of CVPR, Washington, D.C. (1983).

[HAN78] Hanson, A. and Riseman, E. (Eds.), "Computer Vision Systems", Academic Press Inc., New York, NY, 1978.

[JER83] Jerian, C. and Jain, R., *Determining Motion Parameters for Scenes with Translation and Rotation*, Proc. of the Workshop on Motion, Toronto, Canada (1983).

[LAW82] Lawton, D.T., *Motion Analysis via Local Translational Processing*, Proc. of the Workshop in Computer Vision, Rindge, NH (1982).

[LAW84] Lawton, D.T., *Processing Dynamic Image Sequences from a Moving Sensor*, Ph.D. Dissertation (TR 84-05), Computer and Information Science Dept., Univ. of Mass. (1984).

[LON80] Longuet-Higgins, H.C. and Prasdny, K., *The interpretation of a Moving Retinal Image*, Proc. Roy. Soc. Lond., B **208** (1980.).

[LON81] Longuet-Higgins, H.C., *A Computer Algorithm for Reconstructing a Scene from Two Projections*, Nature **293** (1981).

[NAG81a] Nagel, H.H., *On the Derivation of 3D Rigid Point Configurations from Image Sequences*, Proc. of PRIP, Dalas, Texas (1981).

[NAG81b] Nagel, H.H. and Neumann, B., *On 3D Reconstruction from Two Perspective Views*, Proc. of 7th IJCAI, Vancouver, Canada (1981).

[NEU80] Neumann, B., *Motion Analysis of Image Sequences for Object Grouping and Reconstruction*, Proc. of Pattern Recognition, Miami, Florida (1980).

[ORO81] O'Rourke, J., *Motion Detection Using Hough Techniques*, Proc. of PRIP, Dalas, Texas (1981).

[PRA80] Prasdny, K., *Egomotion and Relative Depth Map from Optical Flow*, Biol. Cybernetics **36** (1980).

[PRA81] Prasdny, K., *Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinearly Moving Observer*, Proc. of PRIP, Dalas, Texas

(1981).

[RIE83] Rieger, J.H. and Lawton, D.T., *Determining the Instantaneous Axis of Translation from Optic Flow Generated by Arbitrary Sensor Motion*, Proc. of the Workshop on Motion, Toronto, Canada (1983).

[ROA80] Roach, J.W. and Aggarwal, J.K., *Determining the Movement of Objects from a Sequence of Images*, PAMI **2** (1980).

[SLO81] Sloan, K.R., *Dynamically Quantized Pyramids*, Proc. of 7th IJCAI, Vancouver, Canada (1981).

[TSA84] Tsai, R.Y. and Huang, T.S., *Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces*, PAMI **6** (1984).

[ULL79] Ullman, S., "The Interpretation of Visual Motion", MIT Press, Cambridge, Mass., 1979.

[ULL81] Ullman, S., *Analysis of Visual Motion by Biological and Computer Systems*, Computer **14** (1981).

[WAX83] Waxman, A.M. and Ullman, S., *Surface Structure and 3-D Motion from Image Flow: A Kinematic Approach*, CAR-TR-24, Center for Automation Research, Univ. of Maryland (1983).

[WIL81] Williams, T.D., *Computer Interpretation of a Dynamic Image from a Moving Vehicle*, Ph.D. Dissertation (TR 81-22), Computer and Information Science Dept., Univ. of Mass. (1981).

Figures*

Figure 6.1: Experiment 1.

Figure 6.2: Experiment 2.

Figure 6.3: Experiment 3.

Figure 6.4: Experiment 4.

* Figure 3.1 appears in page 6.

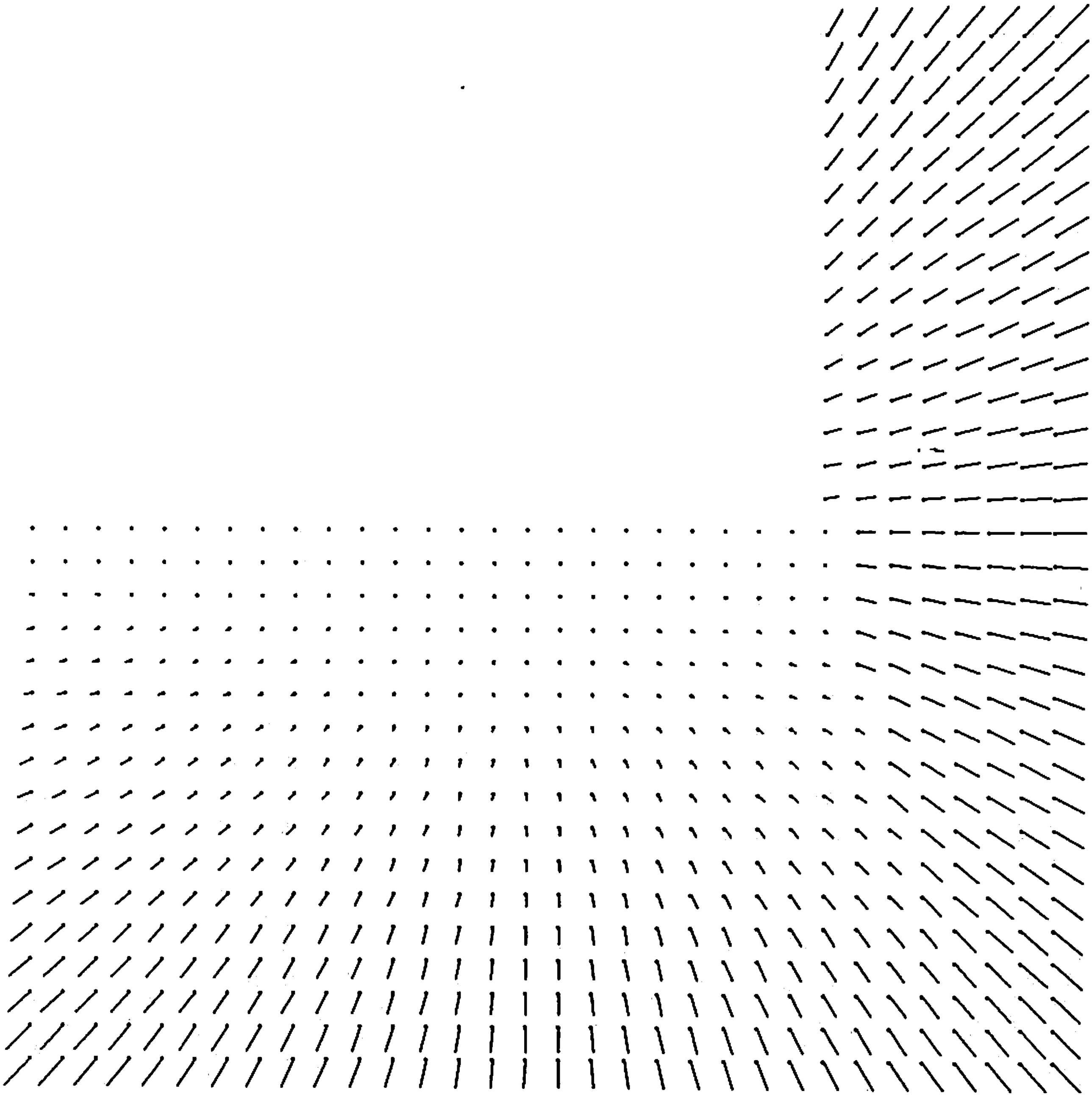

Figure 6.1a: A sample of the flow field.

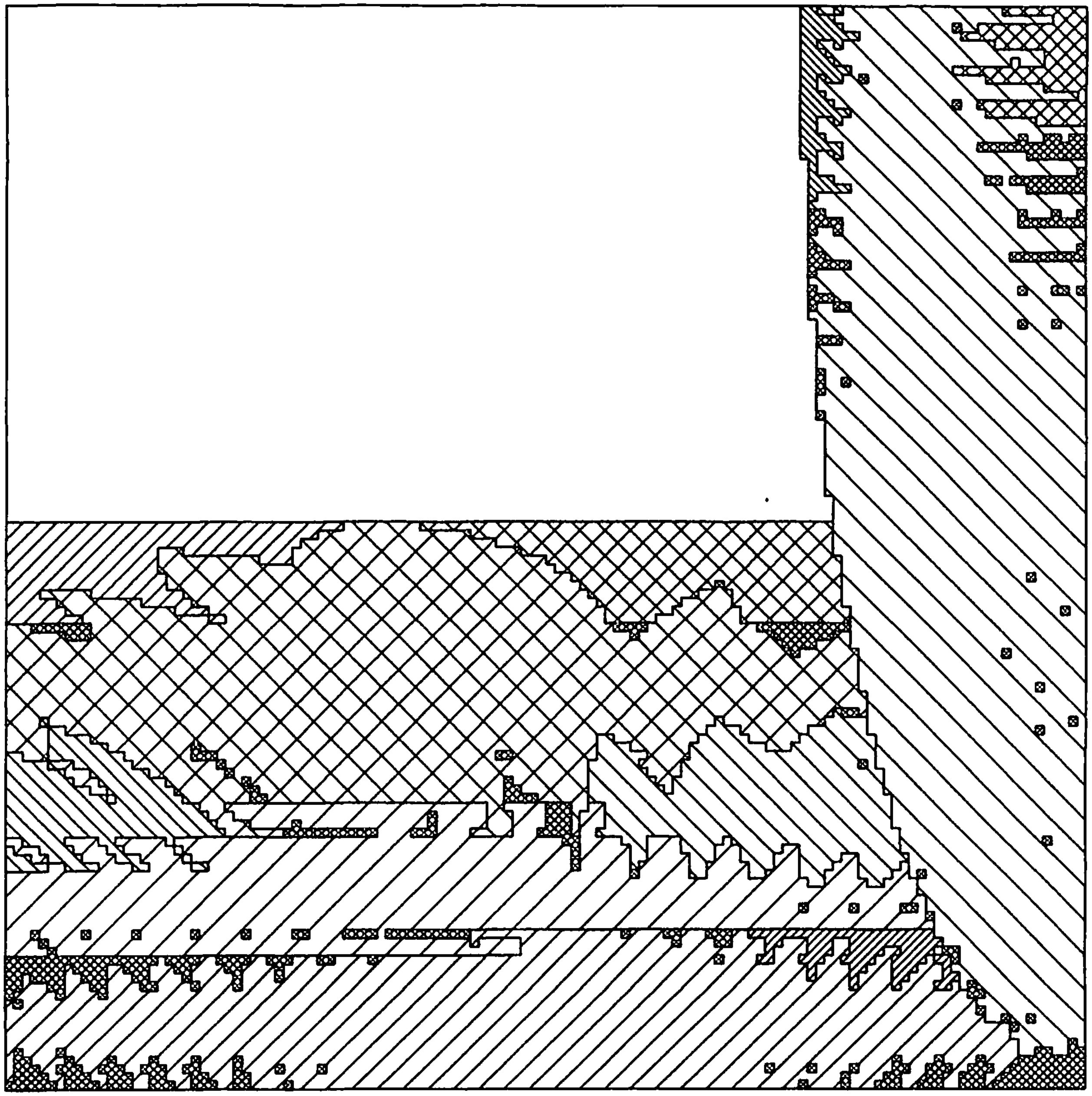

Figure 6.1b: Components determined in the first step of the segmentation. Each component is represented by a specific pattern. The small areas with the densest pattern correspond to vectors which are not contained in any of the components. The irregular shapes of the components were caused by the round-off error.

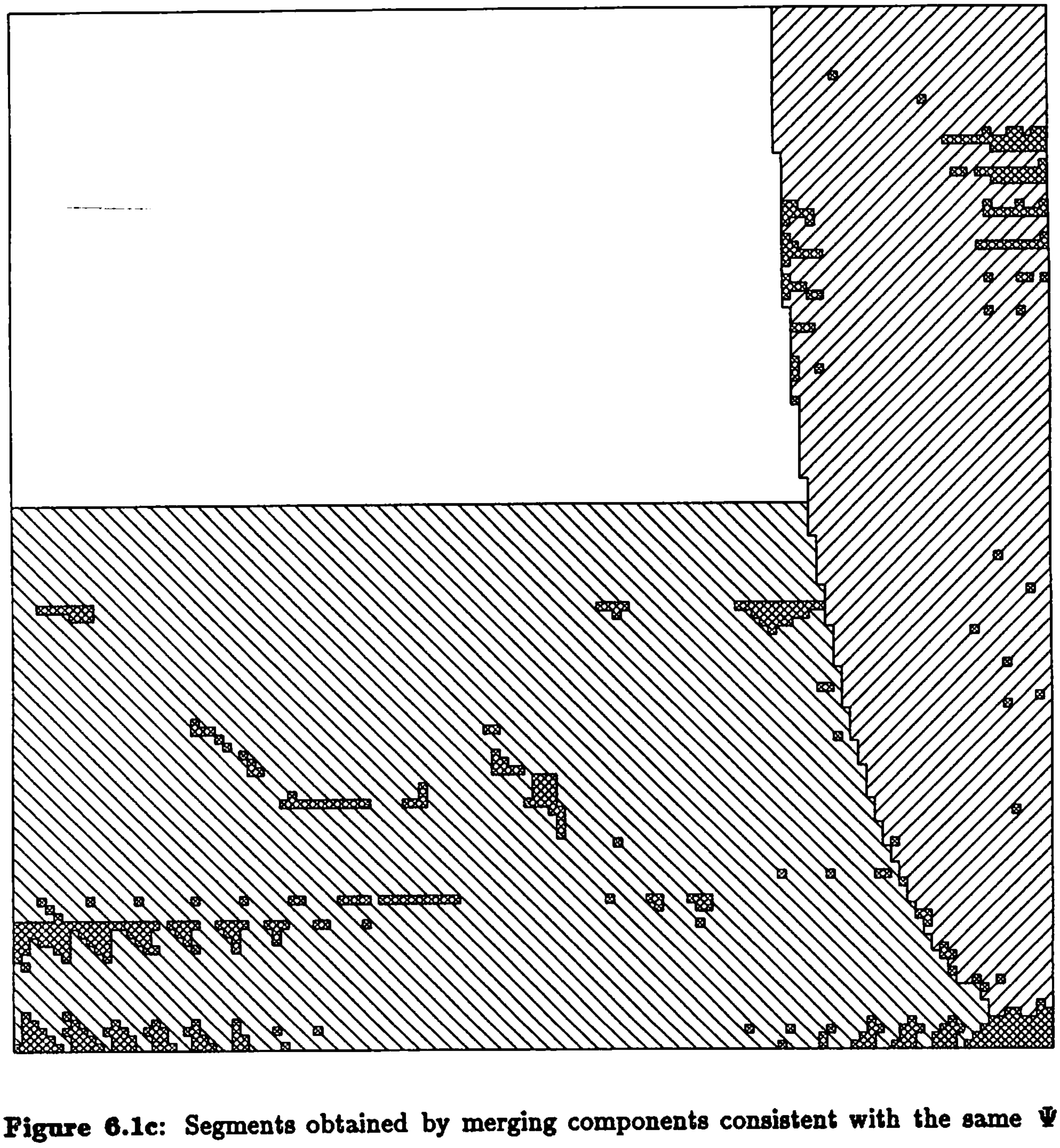

Figure 6.1c: Segments obtained by merging components consistent with the same Ψ transformation.

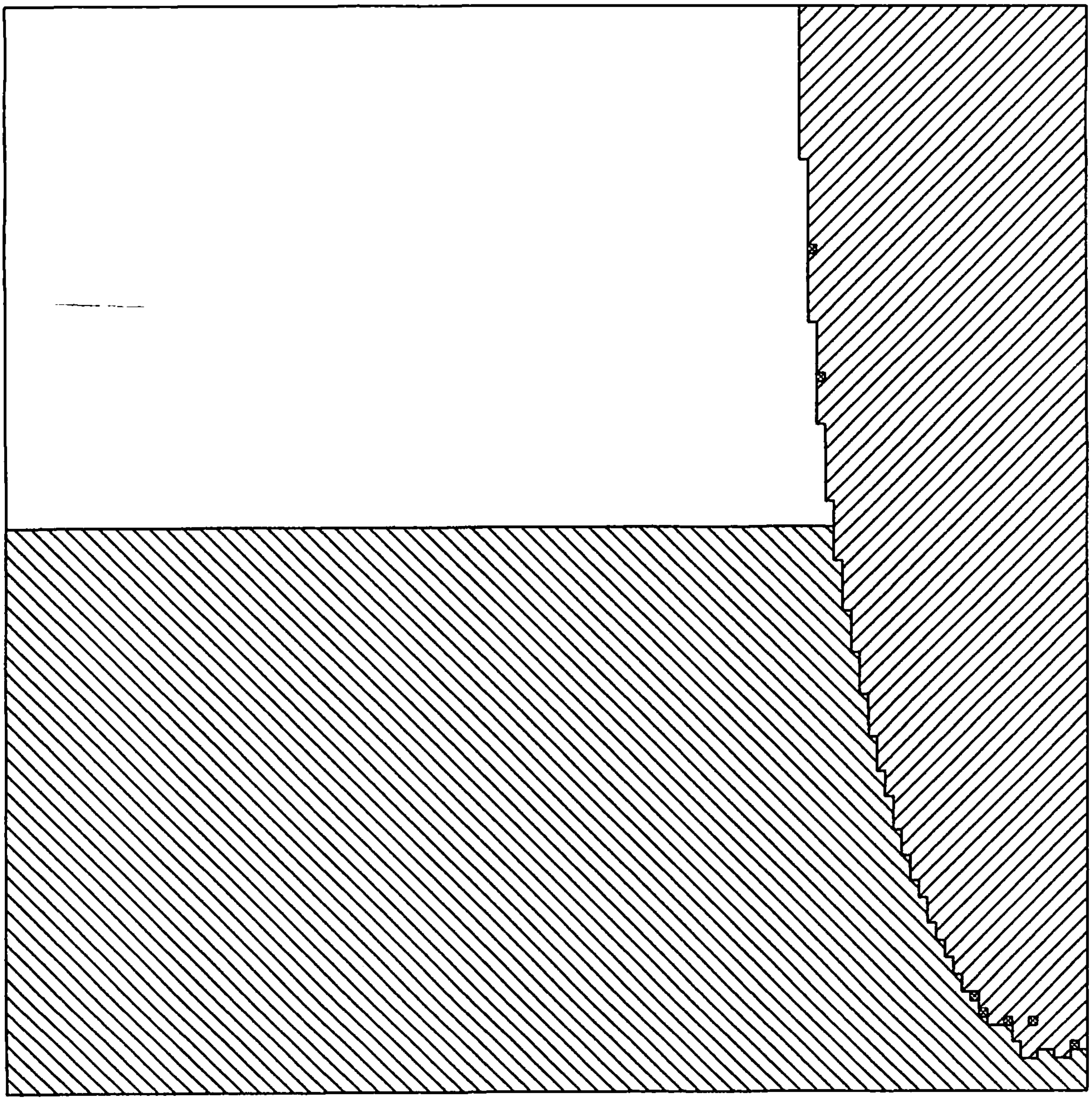

Figure 6.1d: Final segmentation. Note that almost all the flow vectors which, after the first stage, had not been included in any of the components, were merged into the segments.

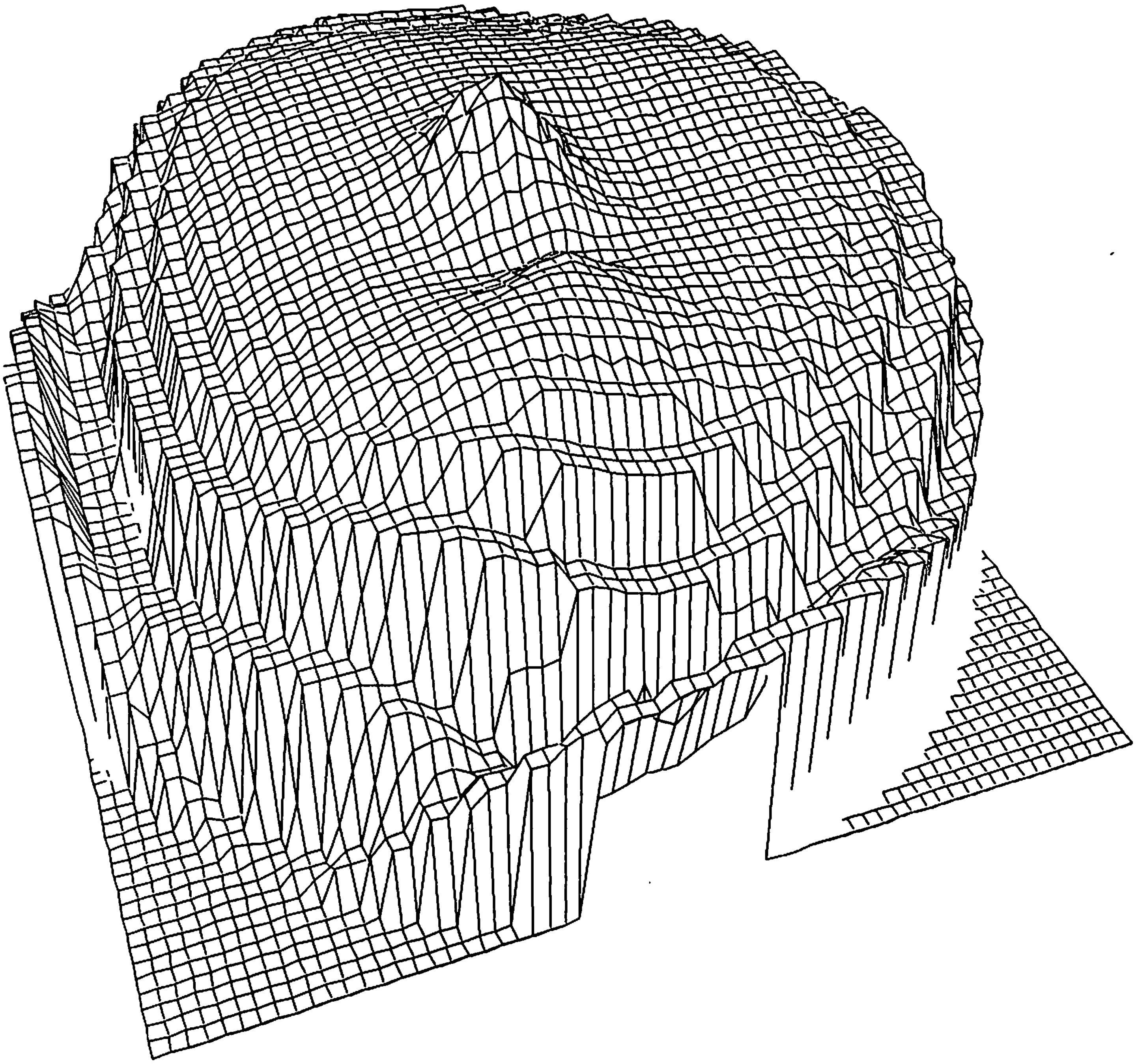

Figure 6.1e: The error function $\hat{\sigma}_1$, shown upside-down, defined on the hemisphere $\{ \underline{U} : |\underline{U}| = 1, U_Z \geq 0 \}$. The spherical coordinates (ϕ, θ), employed in equation (6.2) for representing this hemisphere, are used here as polar coordinates.

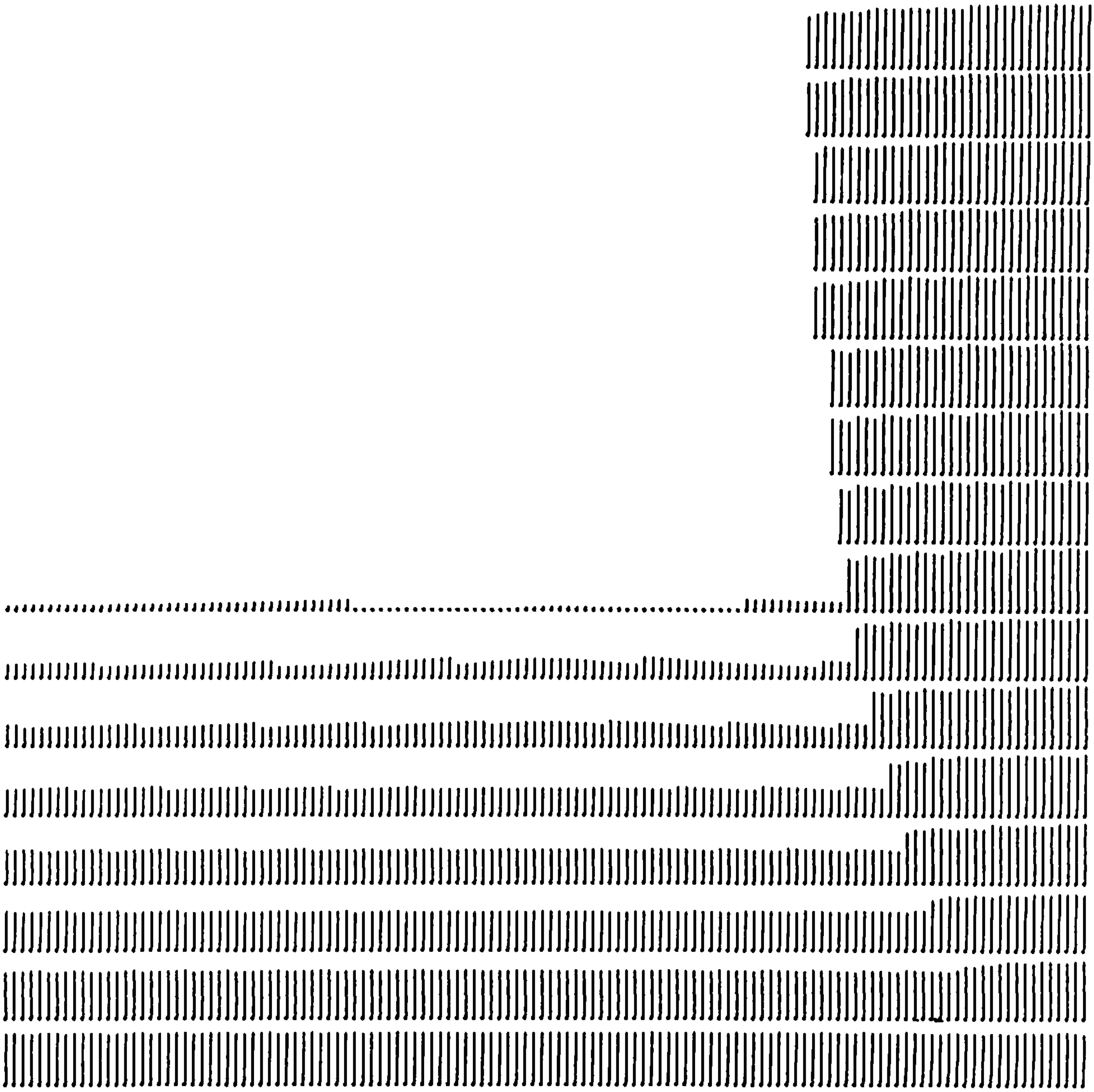

Figure 6.1f: The function r/Z, where r is the length of the translation vector and Z is the environmental depth. The length of each bar represents the relative value of r/Z at the image pixel corresponding to the attached dot.

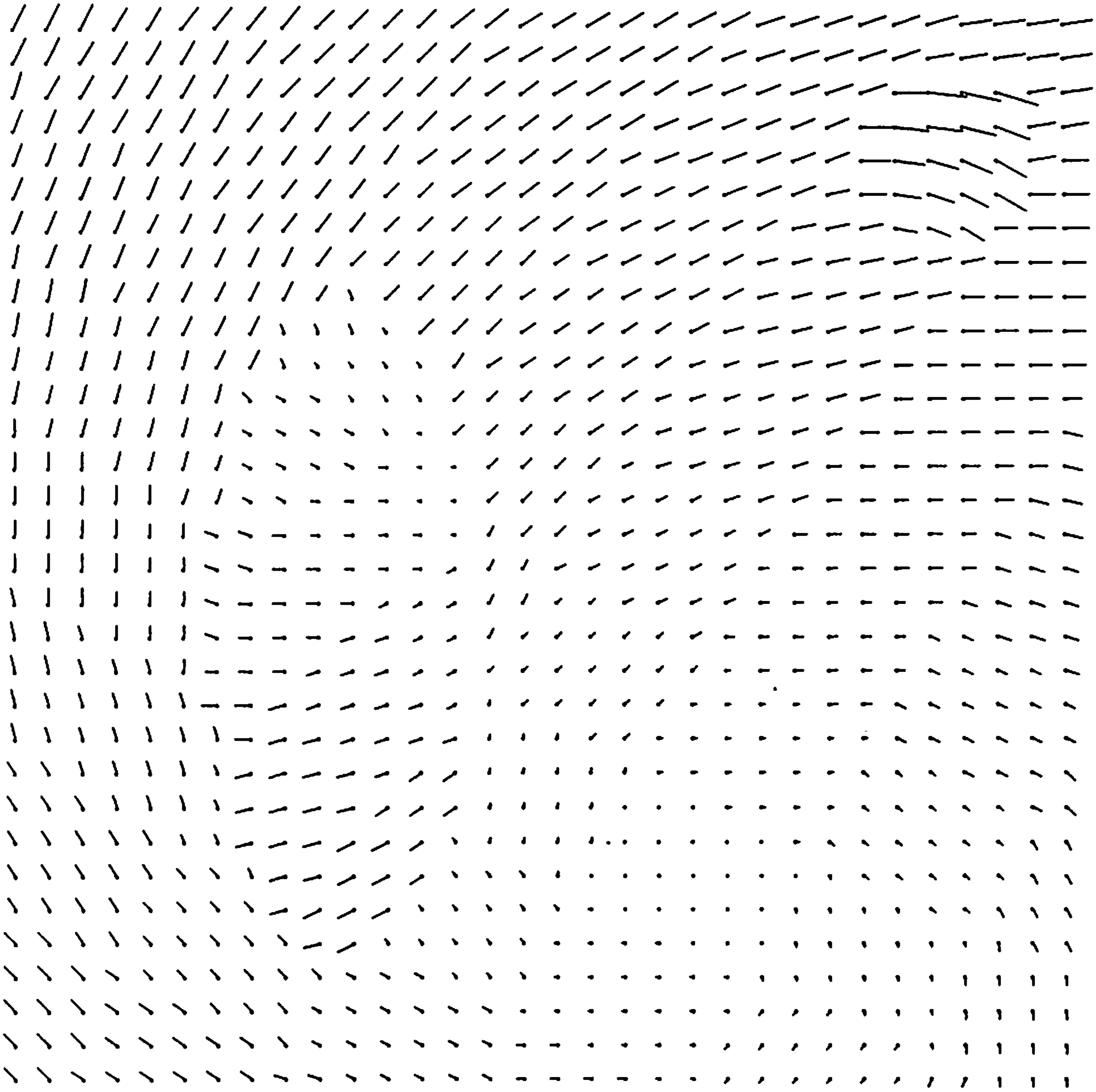

Figure 6.2a: A 32 × 32 sample of the flow field.

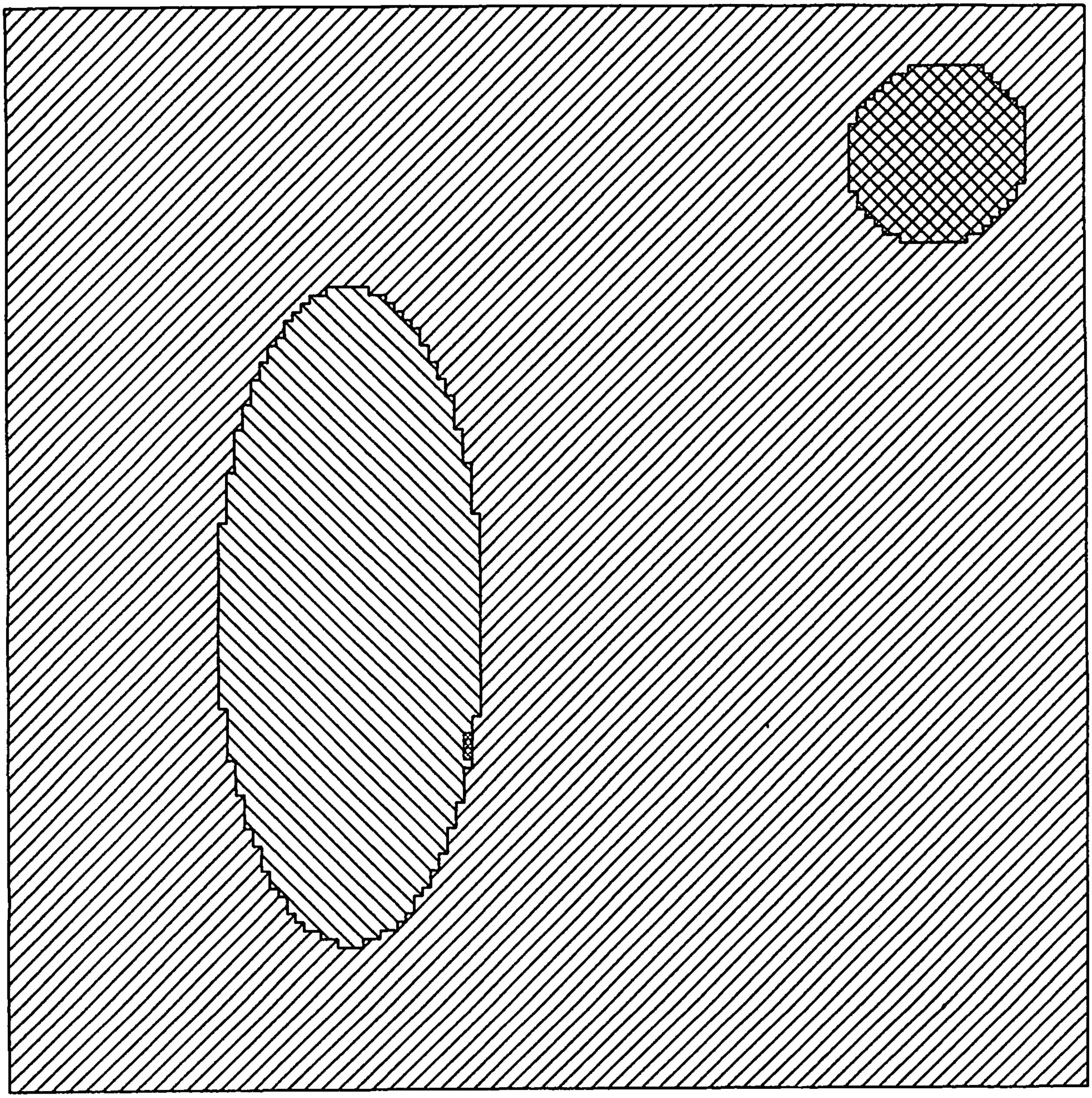

Figure 6.2b: Final segmentation.

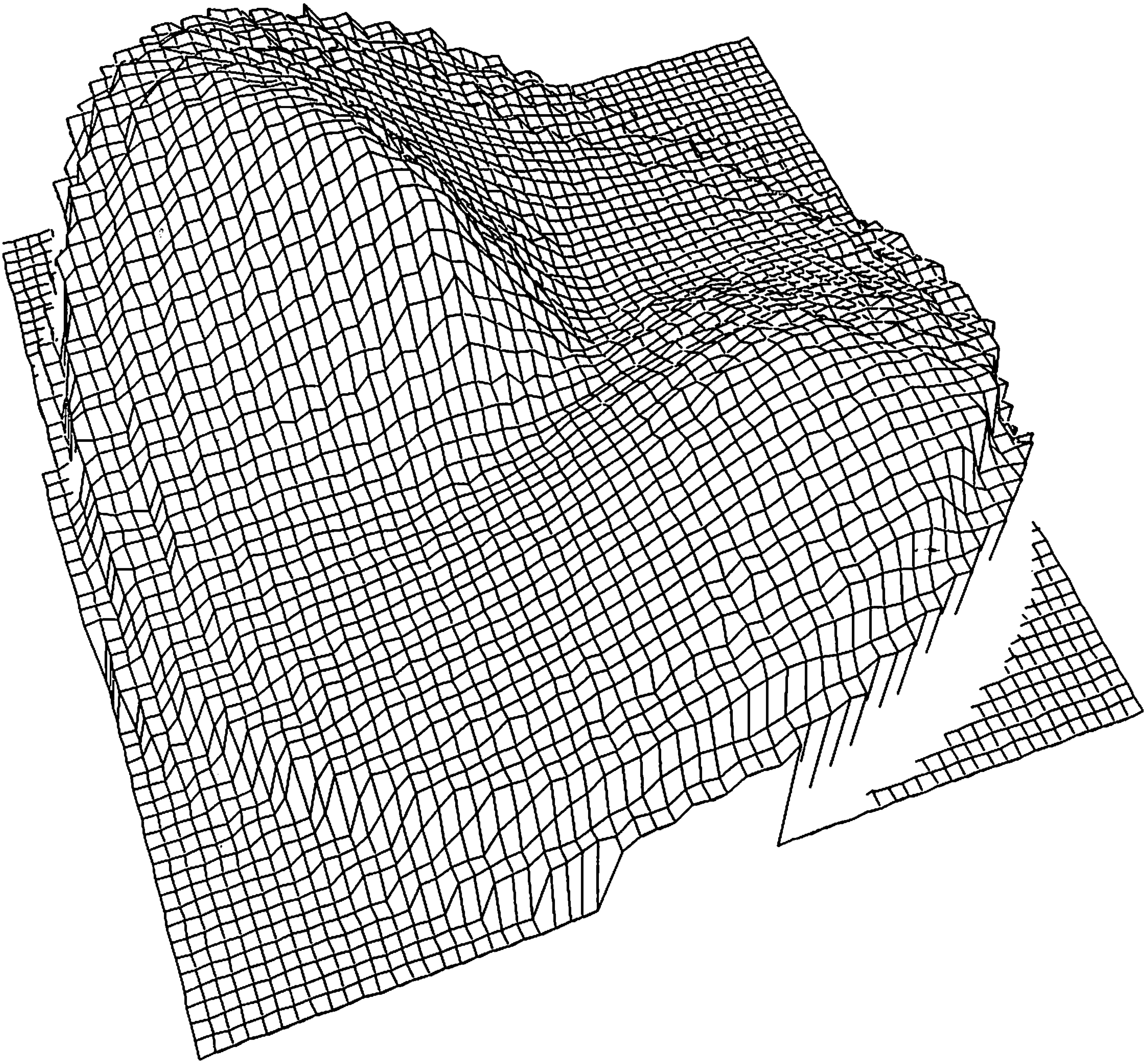

Figure 6.2c: The error function $\hat{\sigma}_1$, shown upside-down, corresponding to the stationary environment.

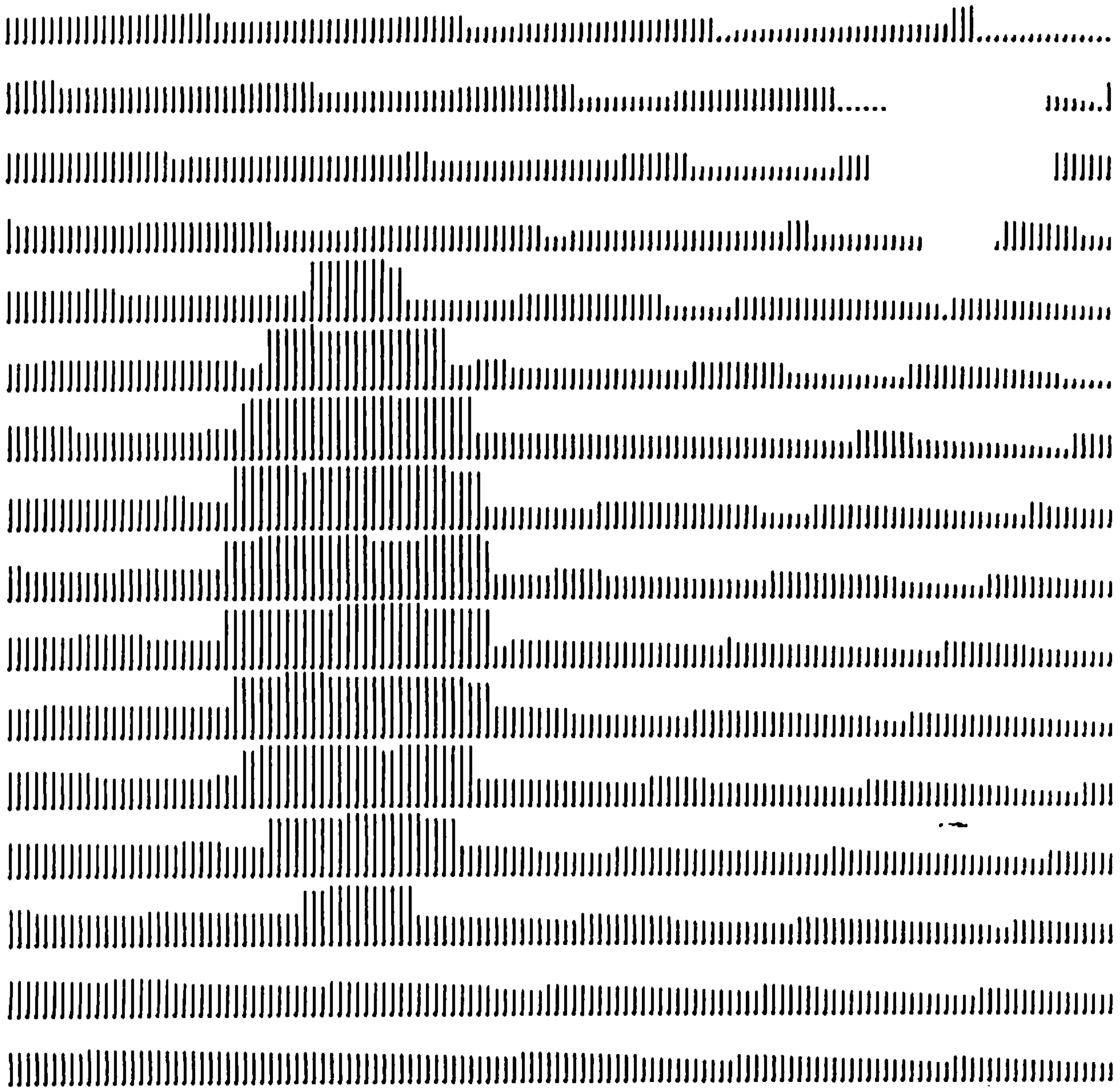

Figure 6.2d: The depth function r/Z corresponding to the stationary environment. The round-off error has a strong effect, especially in the upper right corner, near the focus of expansion.

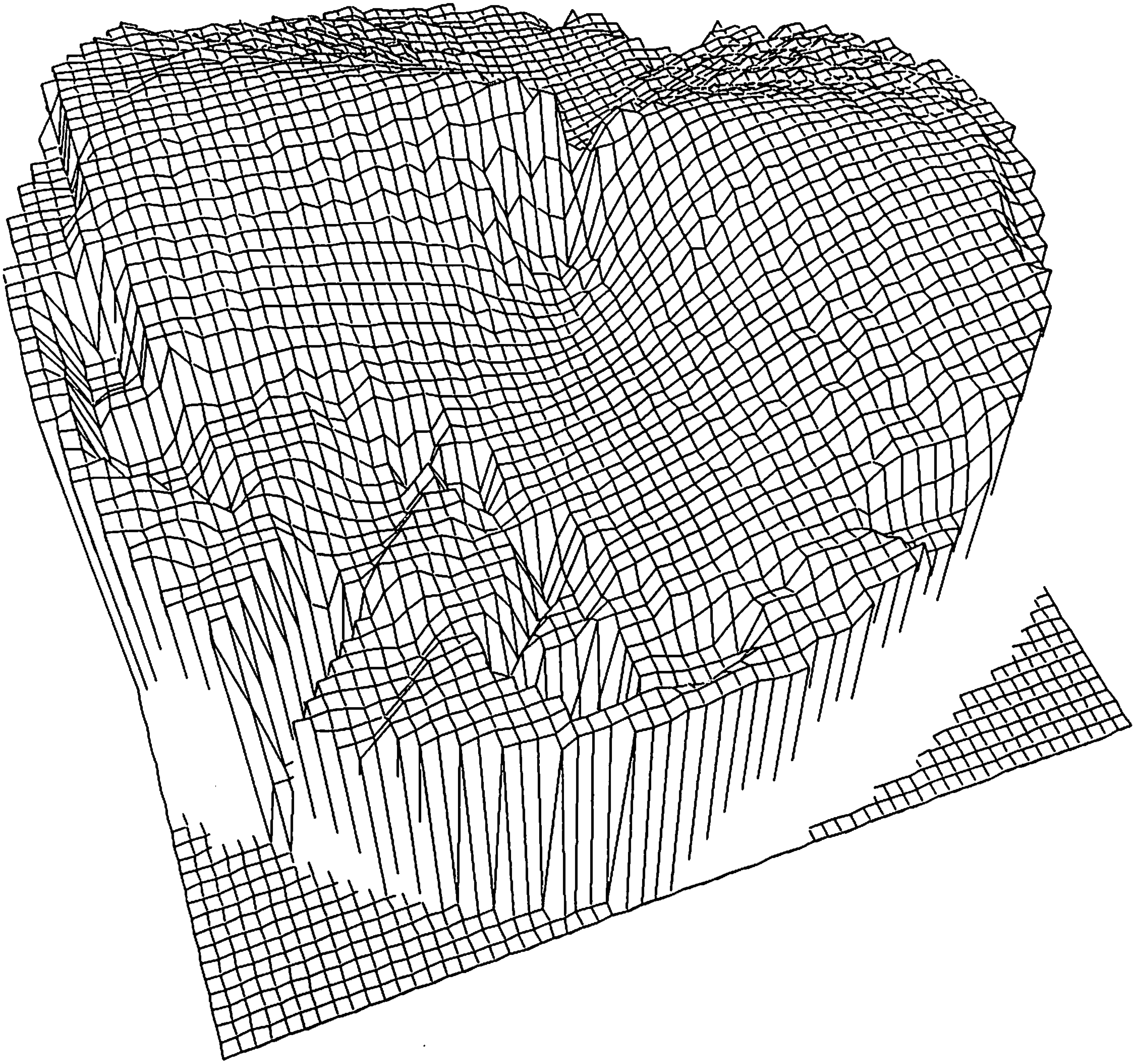

Figure 6.2e: The error function $\hat{\sigma}_1$, shown upside-down, corresponding to the moving object.

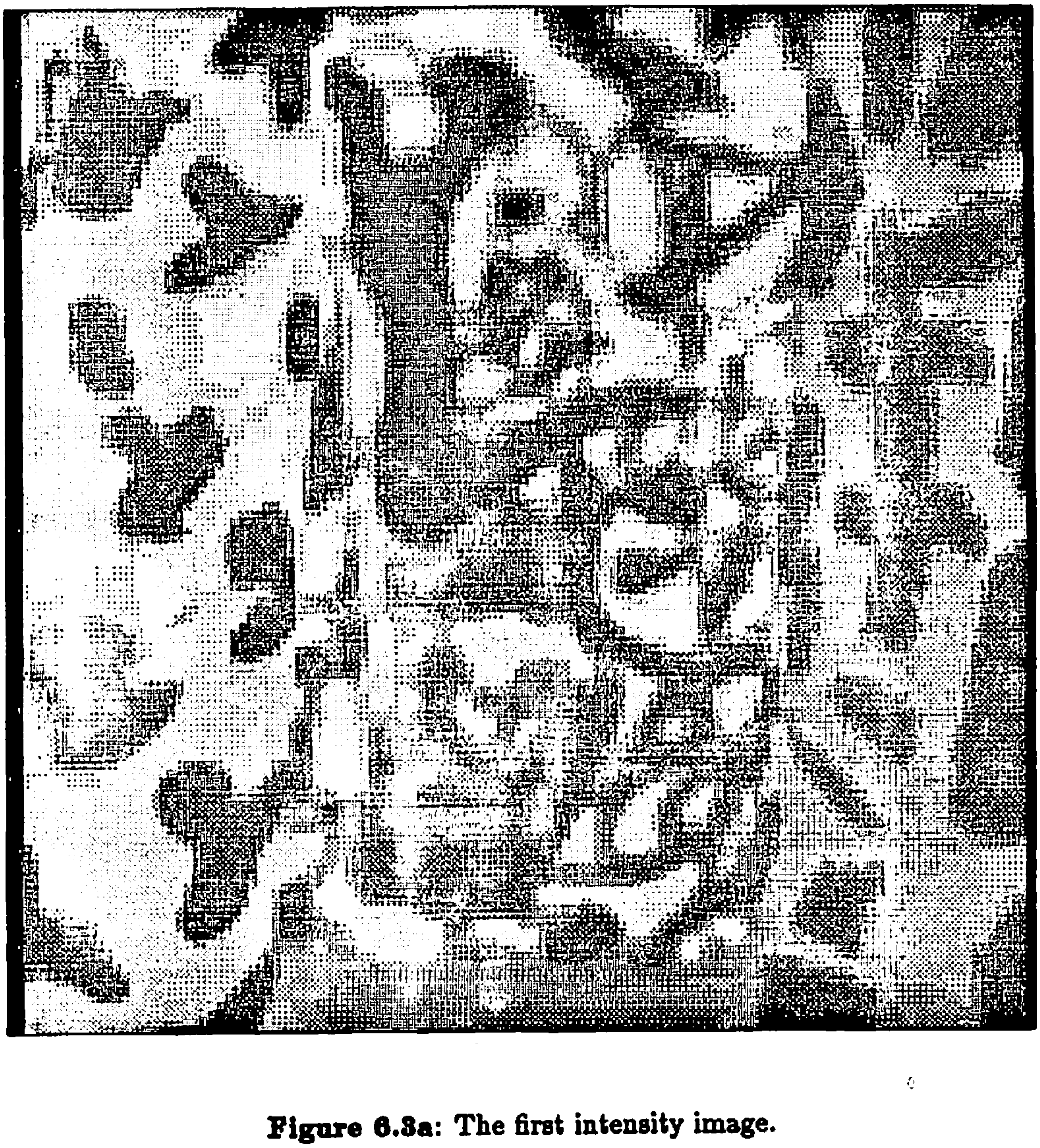

Figure 6.3a: The first intensity image.

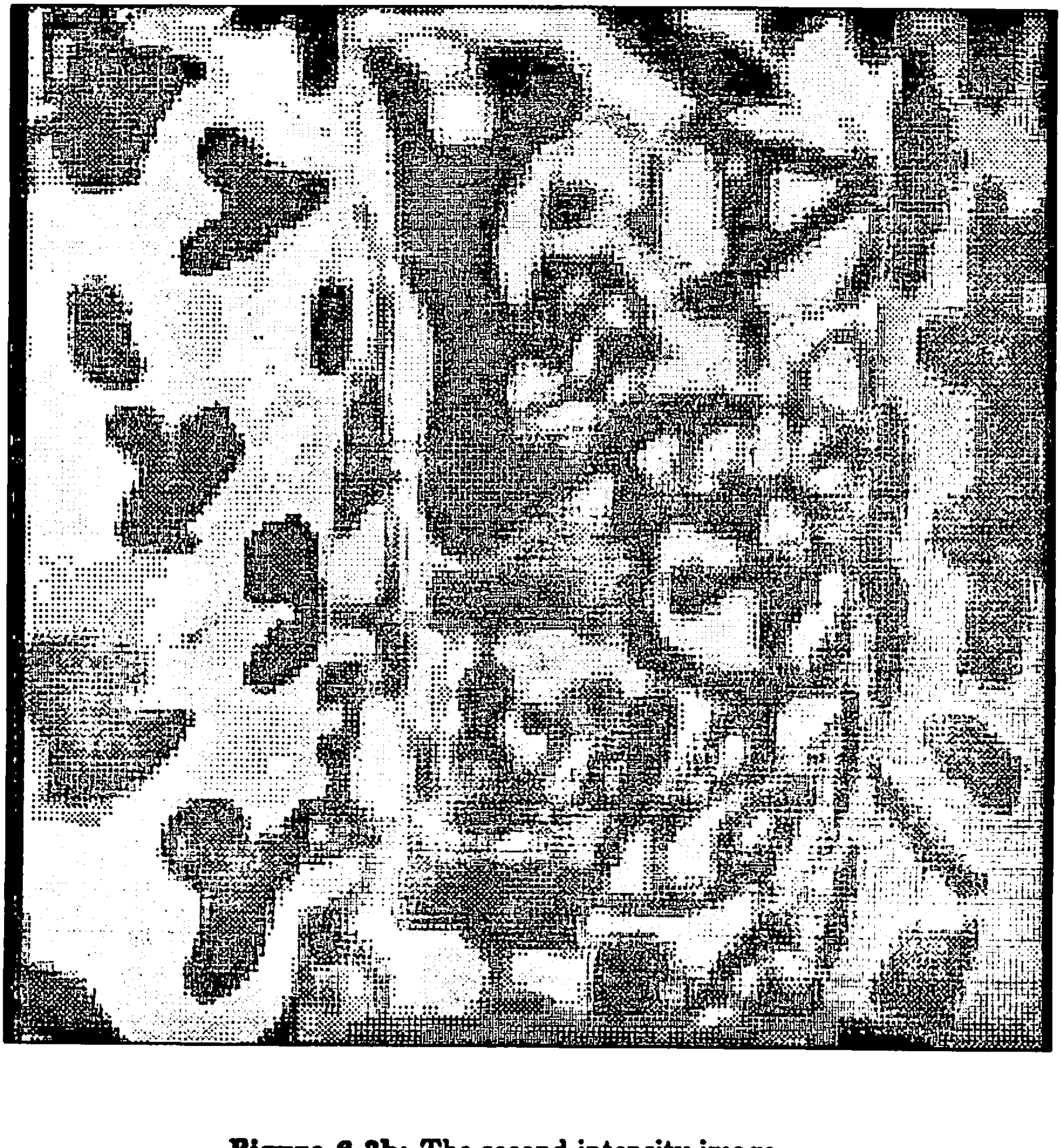

Figure 6.3b: The second intensity image.

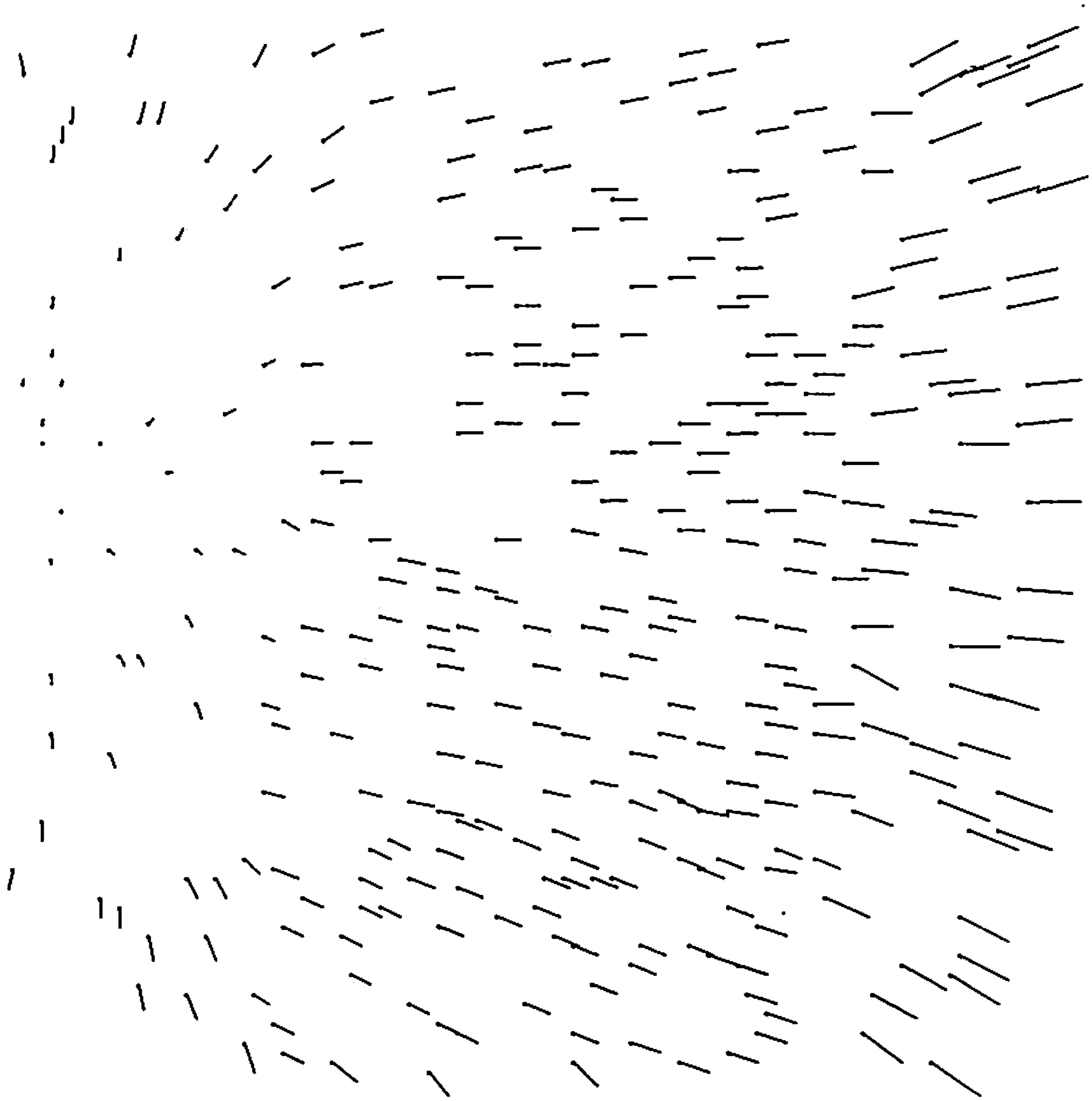

Figure 6.3c: The flow field produced in [RIE83].

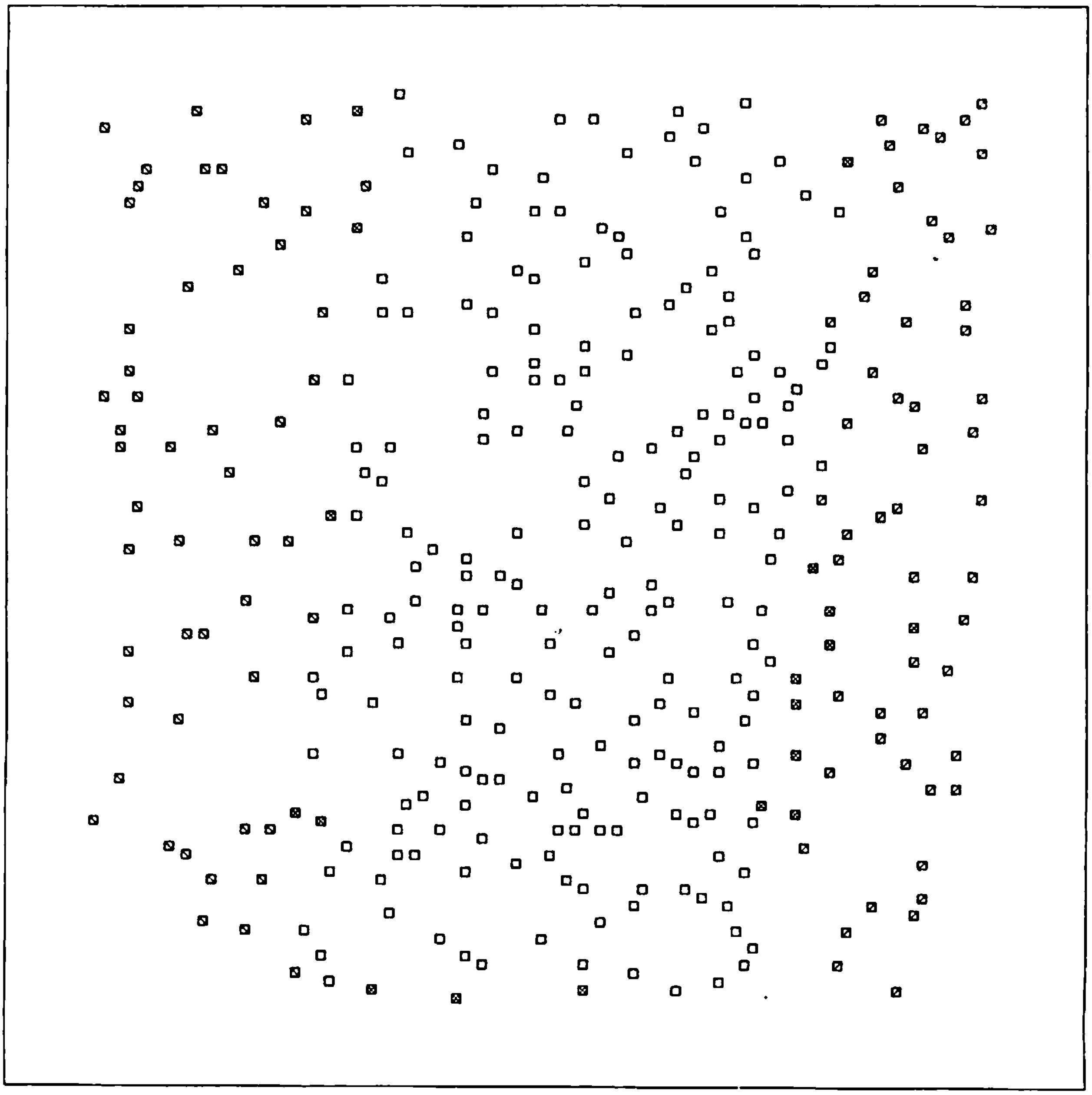

Figure 6.3d: Final segmentation. Again, each segment is represented by a distinct pattern and the points with the densest pattern correspond to flow vectors which are not contained in any of the segments.

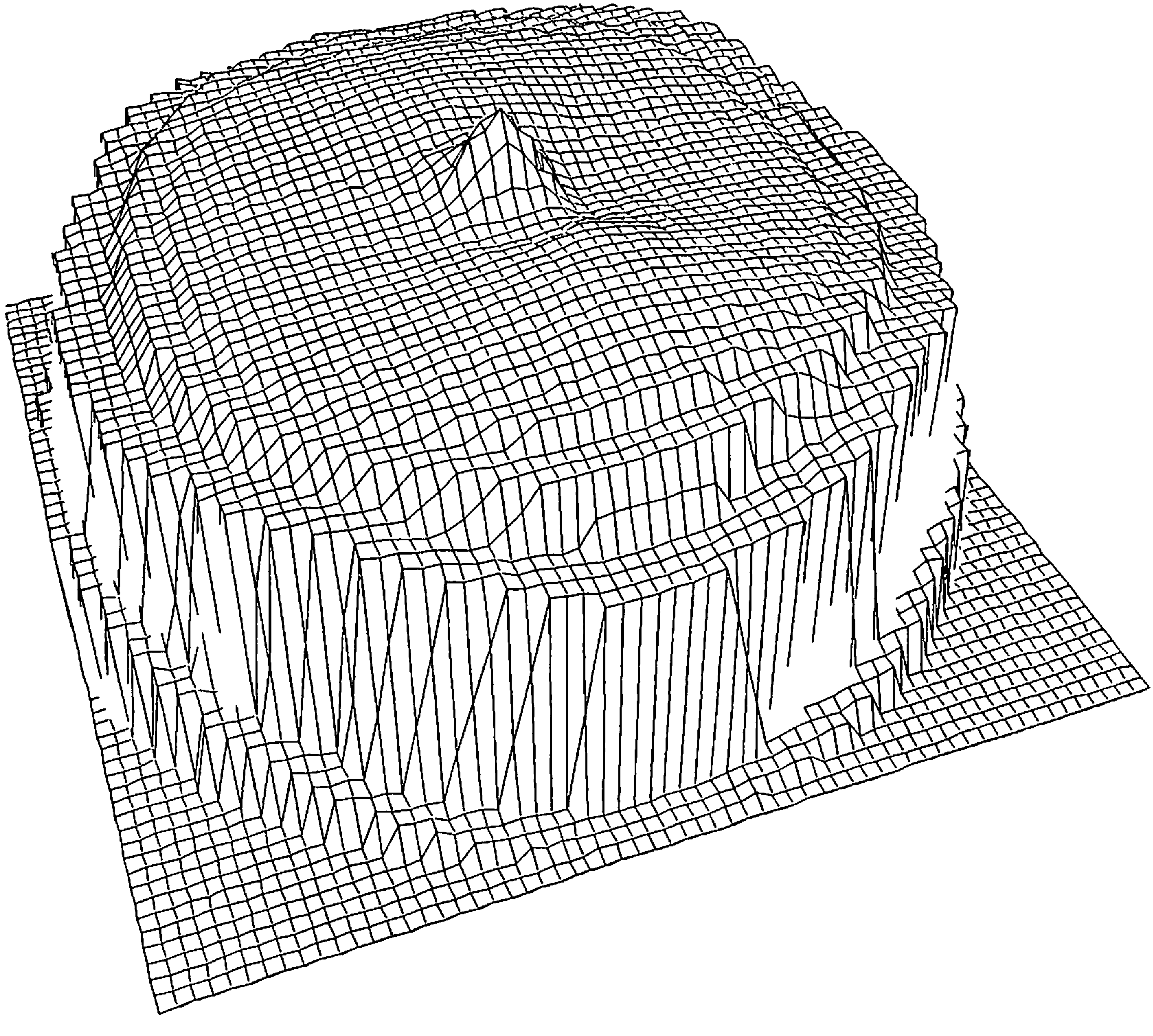

Figure 6.3e: The error function $\hat{\sigma}_1$ shown upside-down.

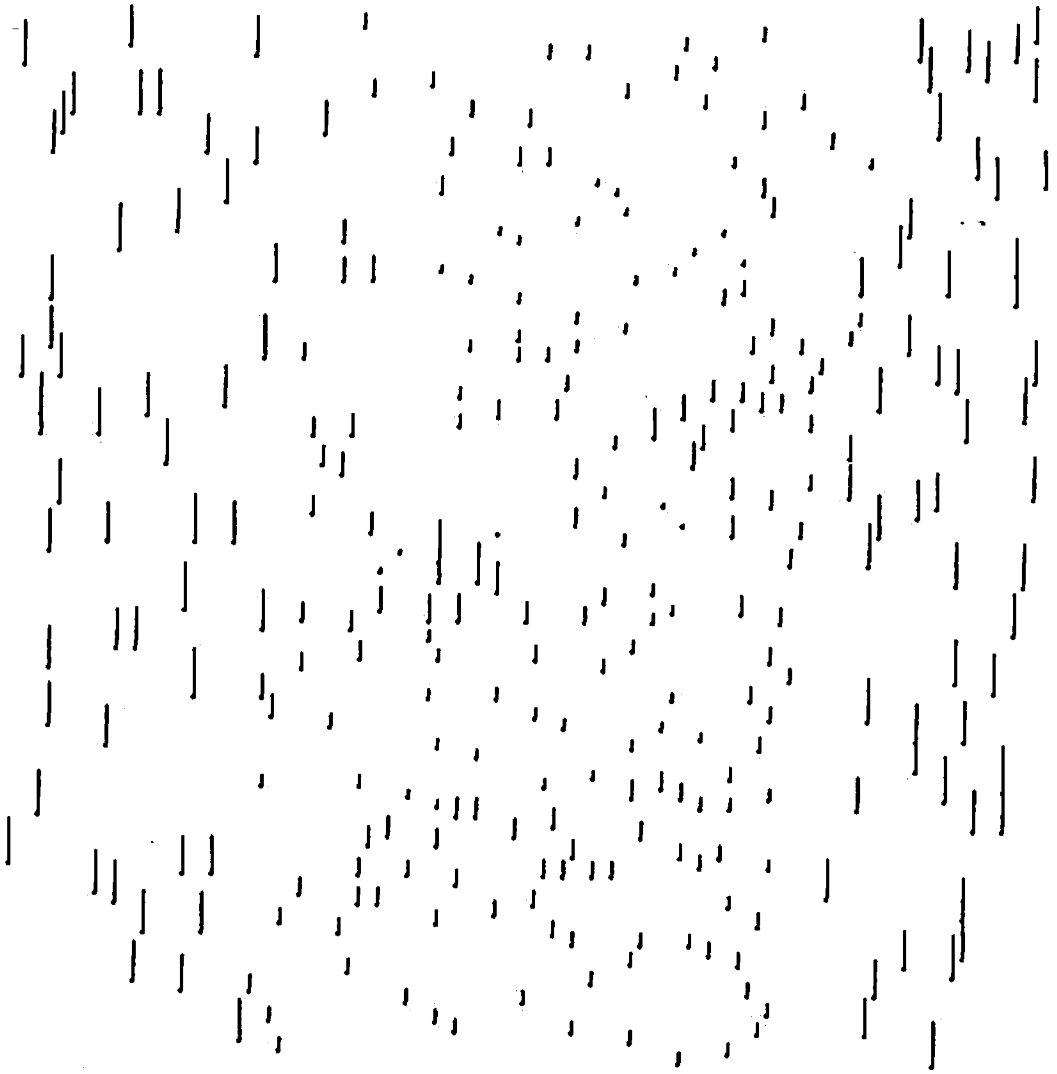

Figure 6.3f: The estimated depth function r/Z.

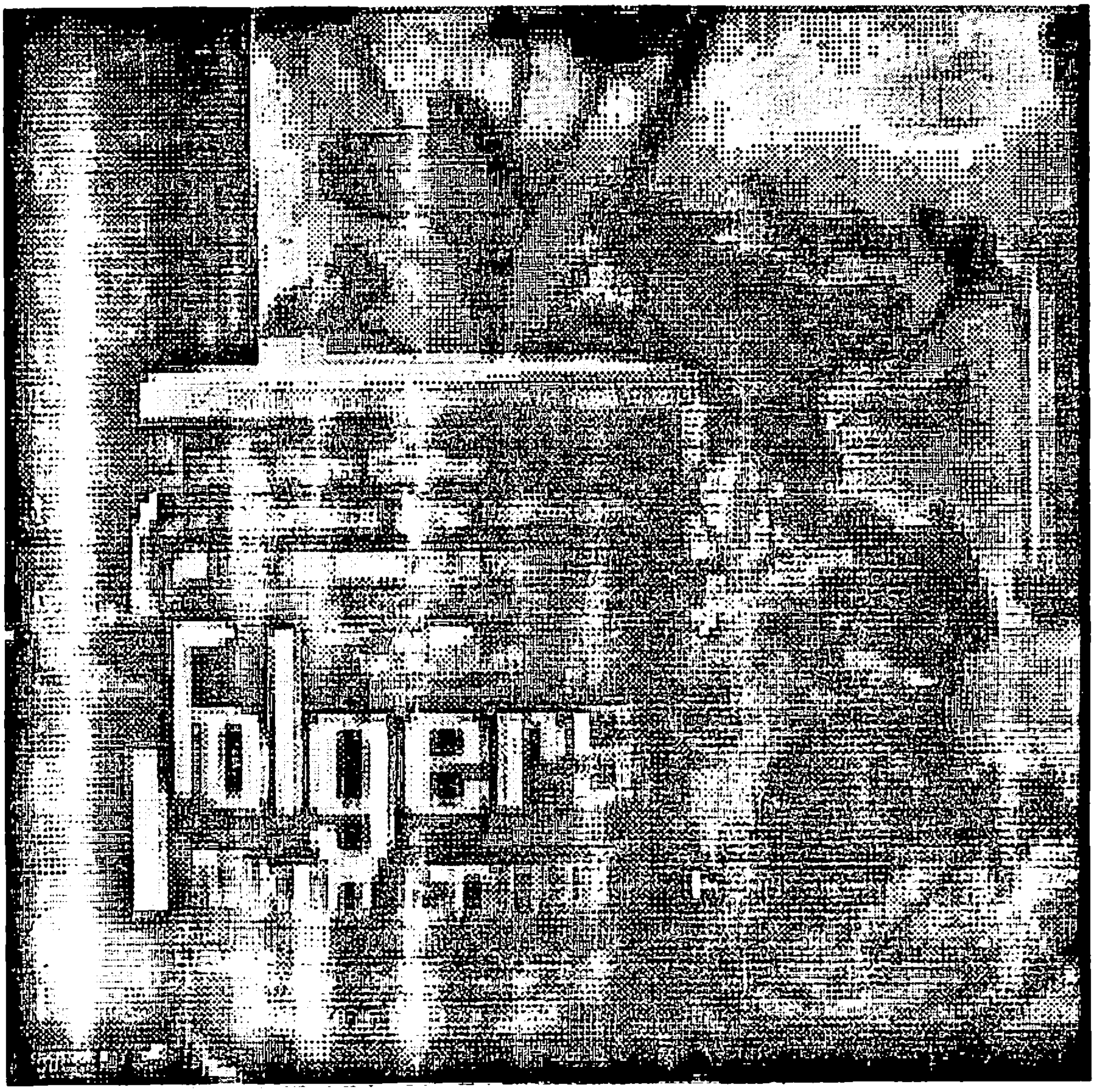

Figure 6.4a: The first intensity image.

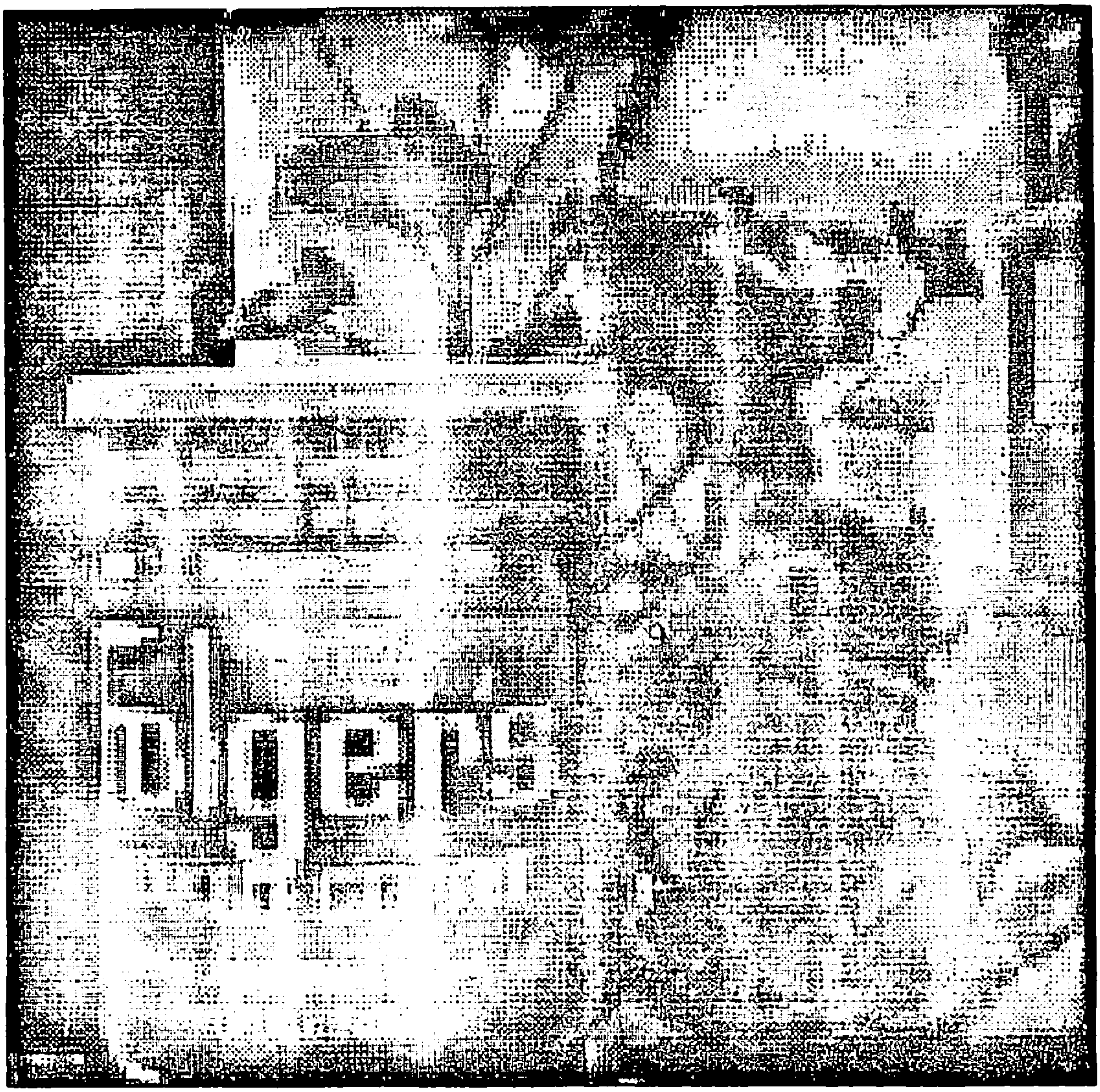

Figure 6.4b: The second intensity image.

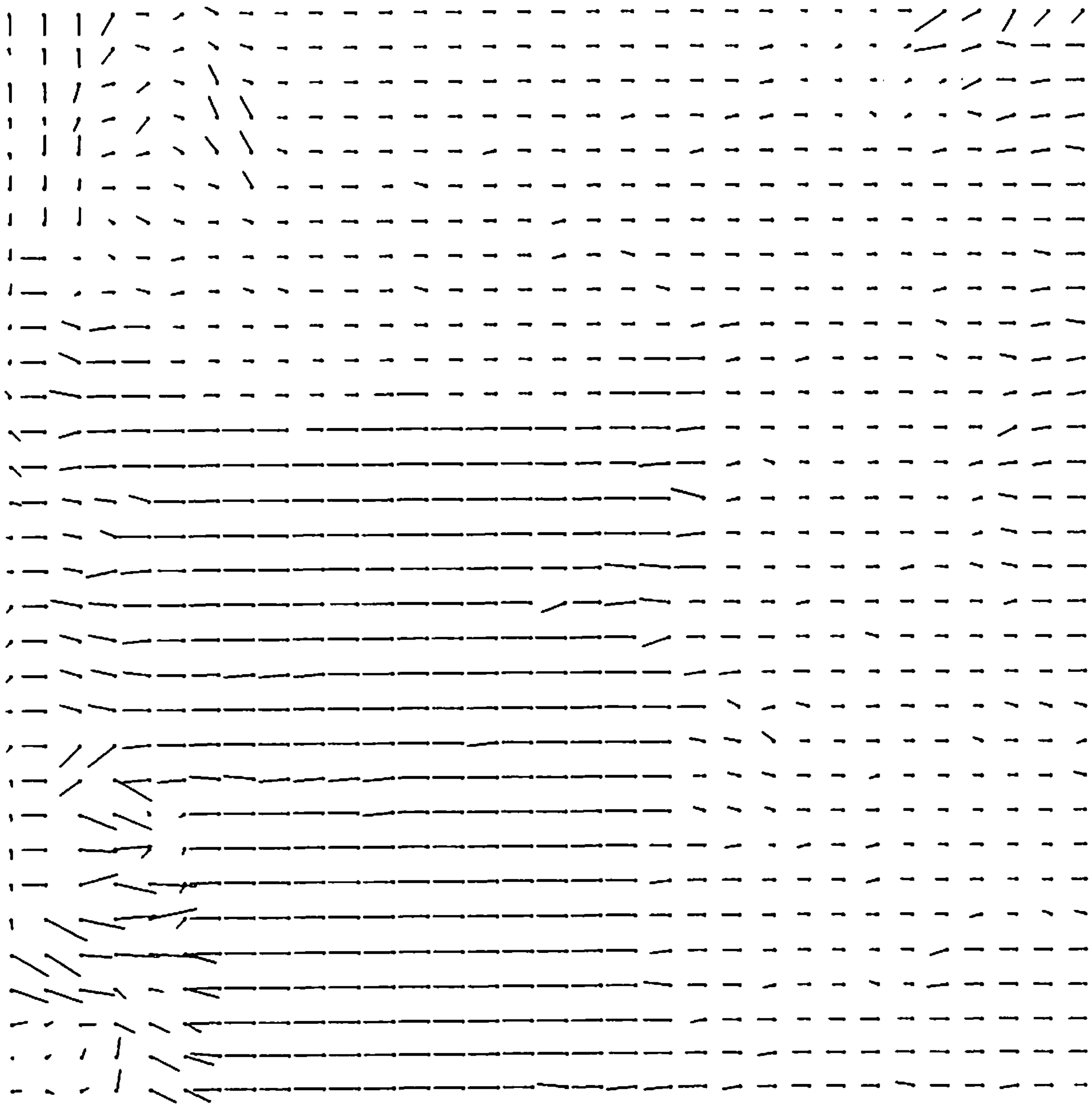

Figure 6.4c: A 32 × 32 sample of the computed flow field.

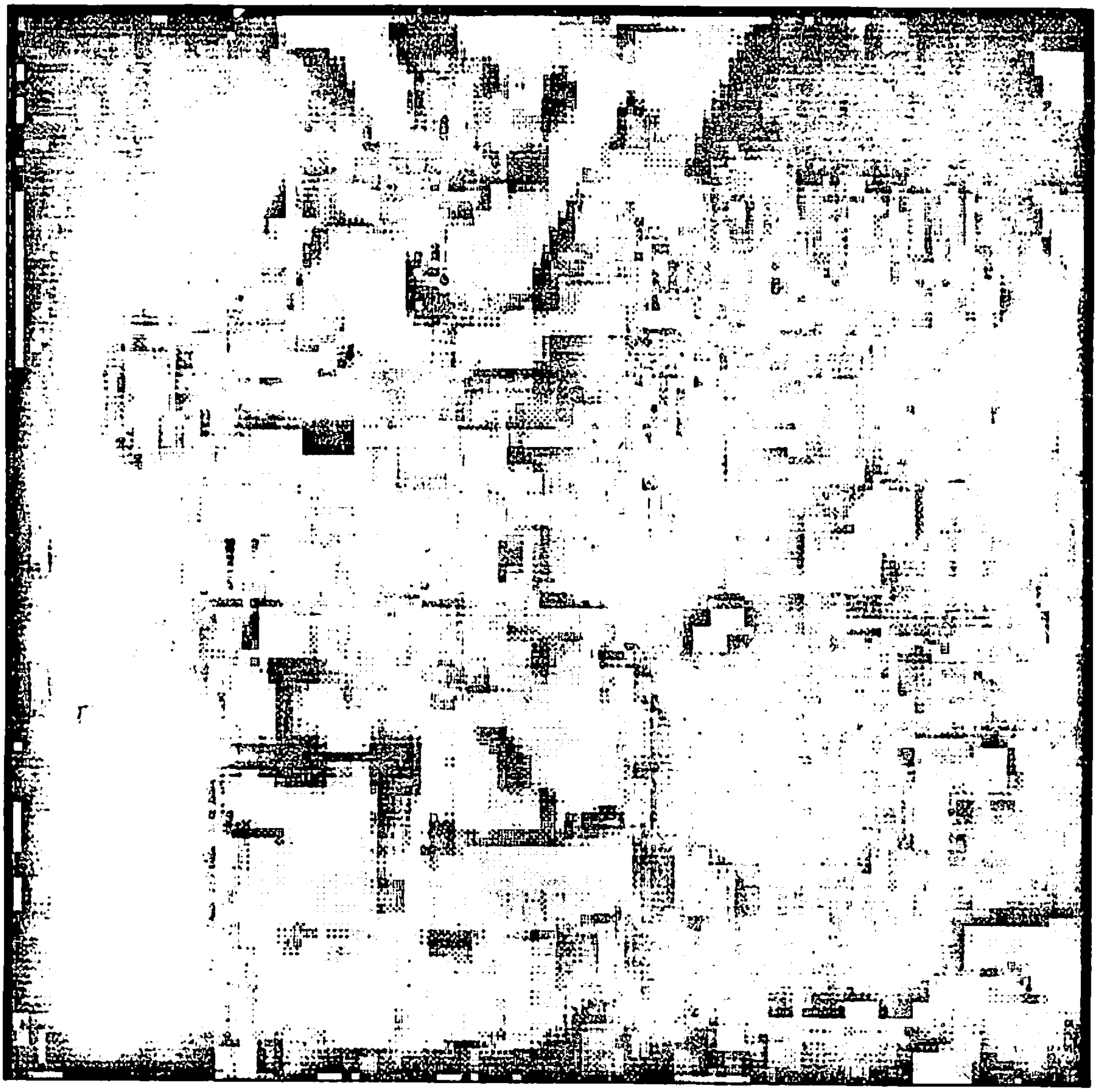

Figure 6.4d: The weight plane. High values are represented by bright gray levels.

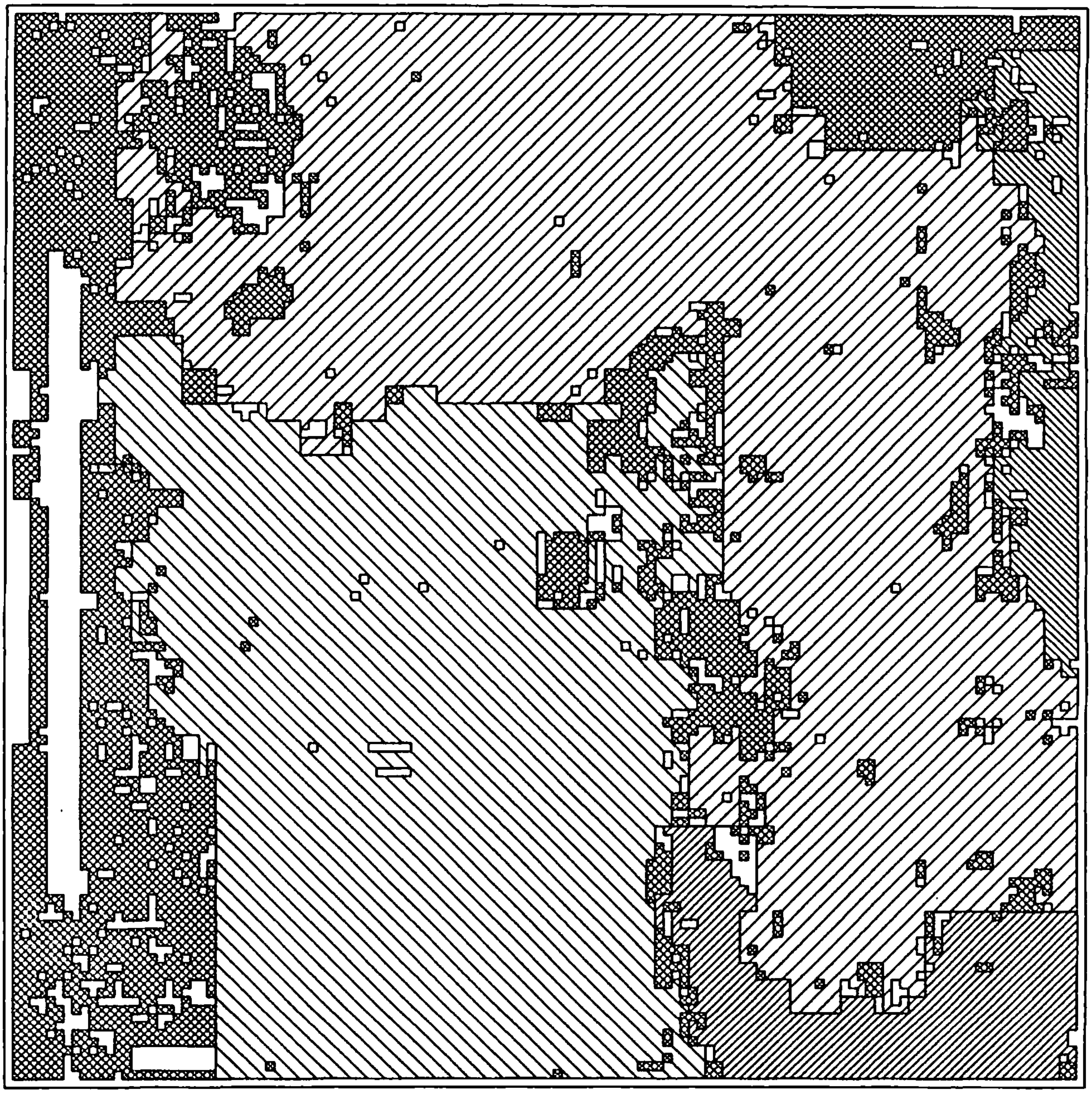

Figure 6.4e: Final segmentation. The white areas correspond to flow vectors assigned weight 0. The areas with the densest pattern correspond to unsegmented vectors.

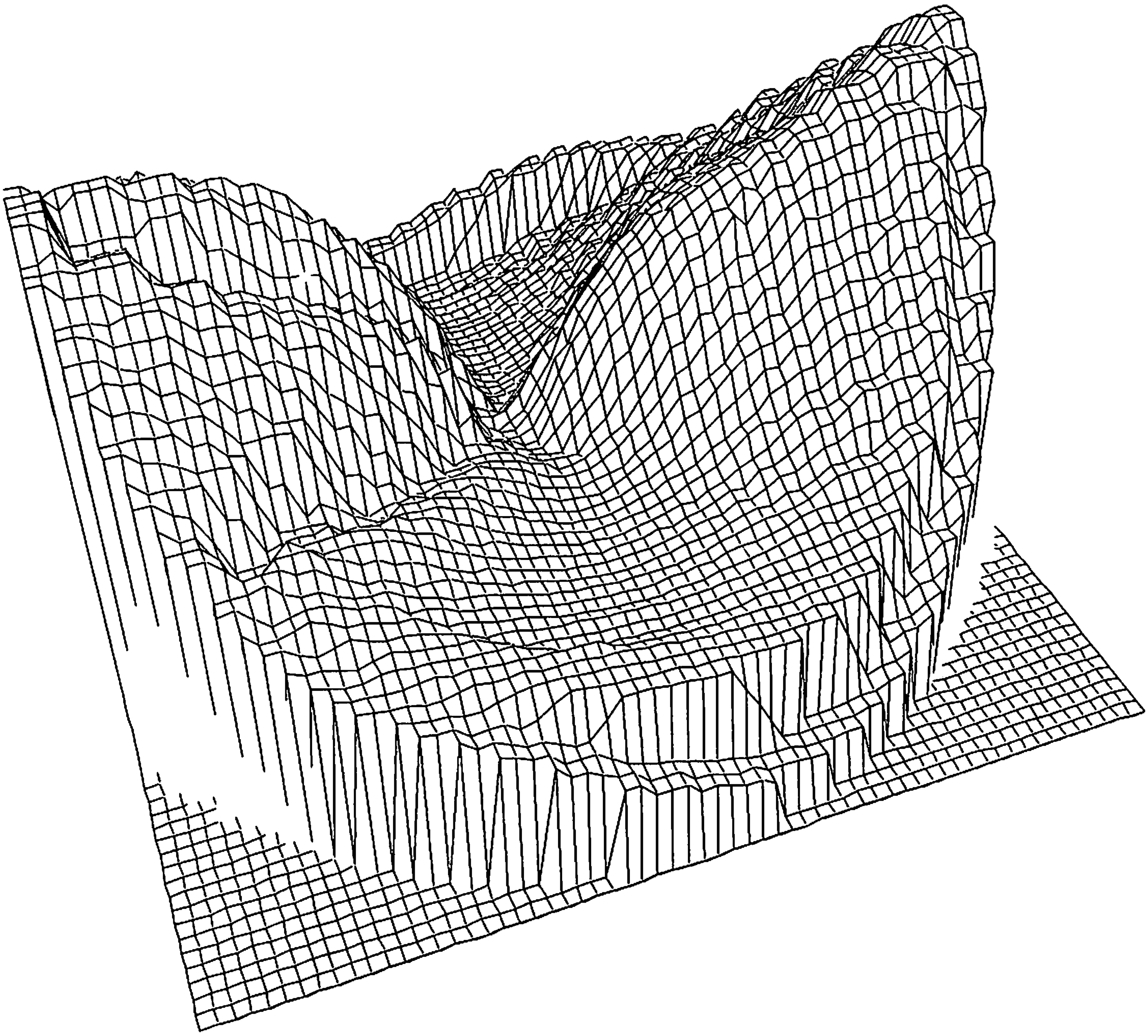

Figure 6.4f: The error function $\hat{\sigma}_1$ shown upside-down. Note the two peaks which actually correspond to the same translation, because $\hat{\sigma}_1$ is invariant to sign change in the translation vector.

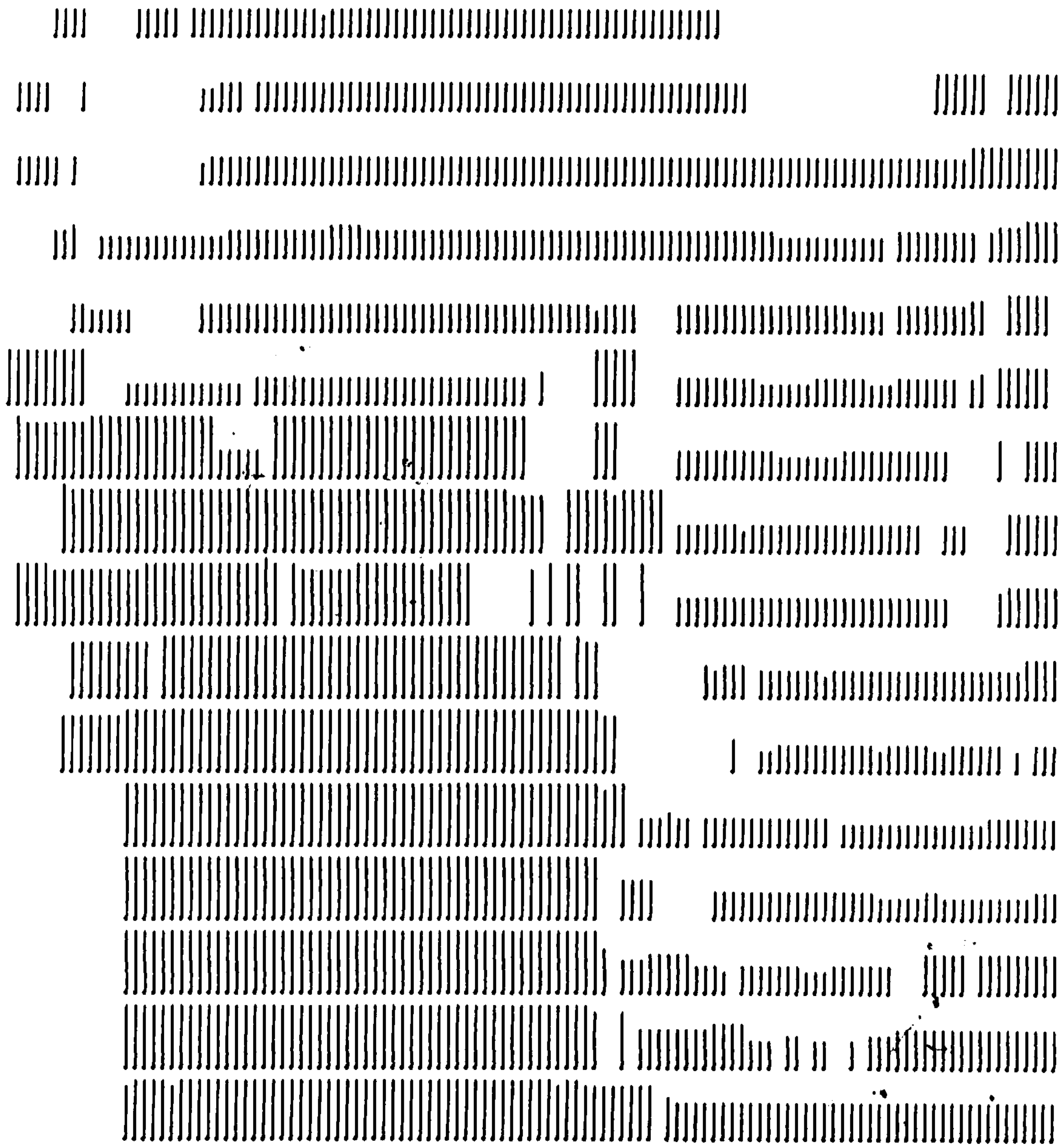

Figure 6.4g: The estimated depth function r/Z.