RECOVERING 2-D MOTION PARAMETERS IN SCENES CONTAINING MULTIPLE MOVING OBJECTS

Gilad Adiv

COINS Technical Report 83-11

May 1983

Abstract

A method for extracting the motion parameters of several independently moving objects from displacement field information is described. The method is based on a generalized Hough transform technique. Some of the problems of this technique are addressed and appropriate solutions are proposed. A modified multipass Hough transform approach has been implemented, where in each pass windows are located around objects and the transform is applied only to the displacement vectors contained in these windows. The windows are determined by the degree to which the displacement field is locally inconsistent with previously found motion transformations. Thus, the sensitivity of the Hough transform to local events is increased and the motion parameters of small objects can be detected even in a noisy displacement field. We also use a multi-resolution scheme in both the image plane and the parameter space and thus reduce the computational cost of the technique. The method is demonstrated by experiments based on artificial images with four parameters of 2-D motion: rotation, expansion and translation in both axes.

This paper will appear in the proceedings of the DARPA Image Understanding Workshop, to be held in Arlington VA, June 1983.

Table of Contents

1. Introduction

A time-varying scene may contain several independently moving objects with unknown location, shape and 3-D structure. The interpretation of such a scene includes the computation of the motion parameters of the camera and each moving object. This information is useful in areas such as robotics and navigation. It could also be used as an intermediate stage for achieving the tasks of object-surround separation and structure determination.

Our approach for recovering the motion parameters is based on two phases. First, we compute a displacement field, composed of vectors describing the displacement of image elements from one image to the next (see section 2). In this paper we assume a dense displacement field, but the second phase is basically independent of this assumption. Each displacement vector is assigned a weight representing its reliability.

In the second phase the displacement field is interpreted and the motion parameters are recovered. This phase, which is the main concern of the paper, is based on the generalized Hough transform technique [BAL81a]. In this technique the motion parameters are represented by a discrete multi-dimensional parameter space where each dimension corresponds to one of the parameters. Each point in this space uniquely characterizes a motion transformation, defined by the corresponding parameter values. A displacement vector "votes" for a point in the space if the corresponding transformation is consistent with this vector. The points receiving the most votes are likely to represent the motion parameters of different objects.

There are a few techniques described in the literature which use the Hough transform for dealing with scenes containing several moving objects. Fennema and Thompson [FEN79] compute spatial and temporal gradients of the image. A Hough transform technique is used to detect velocities which are consistent with a significant portion of the gradient field. A multipass approach is used: first the most prominent peak in the Hough transform is found and thus the velocity of the largest object is recovered. Then the image points which are consistent with this velocity are removed and a new peak is looked for. The process is repeated until no further objects are found. This system is restricted to translation. It also has problems in recognizing significant peaks [THO81].

Ballard and Kimball [BAL81b] consider the case of general 3-D motion of rigid objects, but assume knowledge of depth information. A Hough transform technique for computing the motion parameters from 3-D optic flow is implemented. The simulation, as described in their report, assumes only one moving object, but it is argued that a multipass approach would handle the case of several moving objects.

Jayaramurthy and Jain [JAY82] describe an implementation of the Hough transform technique for computing motion parameters directly from the intensity information. Several moving objects are allowed, but a stationary background and translational motion are assumed.

One of the well known advantages of the Hough technique is its relative insensitivity to noise and partially incorrect or occluded data. Another advantage is its ability to detect consistency in the image. In our case it can group together displacement vectors which satisfy the same motion parameters and presumably belong to one object.

On the other hand, the Hough technique has a few disadvantages. It is insensitive to spatial relations in the displacement field. Thus, a group of non-adjacent elements, which incidently vote for the same motion transformation, may be considered as representing one object, whereas the motion parameters of a small object may be difficult to detect. The technique also has high computational cost. Fine resolution in the parameter space, which is related to the accuracy of the final results, requires large amounts of memory and computation time.

This paper addresses these problems. A few ideas are examined in a restricted case of 2-D motion with four parameters (rotation, expansion and translation in both axes). An analysis of reliability and efficiency considerations is presented (section 3.2) and new solutions are proposed (section 4). A modified multipass Hough transform approach has been implemented, where in each pass windows are located around objects and the transform is applied only to the displacement vectors contained in these windows. The windows are determined by the degree to which the displacement field is locally inconsistent with previously found motion transformations. Thus, the sensitivity of the Hough transform to local events is increased and the motion parameters of small objects can be detected even in a noisy displacement field. We also use a multi-resolution scheme in both the image plane and the parameter space and thus reduce the computational cost of the technique. These ideas are demonstrated by experiments based on artificial images (section 5).

2. Computing a Displacement Field and a Weight Plane

In the first phase of the algorithm we compute a displacement field from two sampled images. These images contain several objects which are moving independently. The background is considered as one of the objects. The motion of each object is composed of rotation, expansion and translation. It can be represented by the following affine transformation:

$$(2.1) \qquad i' = (1+expan)[\cos(rot)\, i - \sin(rot)\, j] + tr_1$$

$$(2.2) \qquad j' = (1+expan)[\sin(rot)\, i + \cos(rot)\, j] + tr_2$$

where (i,j) is a pixel in the first image, (i′,j′) is the corresponding pixel in the second image and rot, expan, tr_1 and tr_2 are the motion parameter values.

The displacement field can be described by $\{(D_1(i,j), D_2(i,j))\}$ where $(D_1(i,j), D_2(i,j))$ represents the displacement vector at the (i,j) pixel in the first image. We compute it by using the Horn and Schunck technique [HOR80] (however, the second phase of our algorithm is almost independent of this specific choice). In order to use this tehnique we assume a small displacement at each pixel and absence of illumination effects. It starts by calculating, at each pixel, the spatial gradient (E_1, E_2) and the temporal derivative E_t. The assumption that the brightness of a particular point in the scene is constant over time provides the following constraint:

$$(2.3) \qquad E_1D_1+E_2D_2+E_t = 0$$

The assumption of the smoothness of the displacement field provides another constraint.

An error function can represent, for a given displacement field, the degree of departure from these constraints. The technique is based on iteratively minimizing this function. Ideally, the resulting field (D_1, D_2) should satisfy the following equations, derived from equations (2.1) and (2.2):

$$(2.4) \qquad i+D_1(i,j) = (1+expan)[\cos(rot)\, i-\sin(rot)\, j]+tr_1$$

$$(2.5) \qquad j+D_2(i,j) = (1+expan)[\sin(rot)\, i+\cos(rot)\, j]+tr_2$$

where rot, expan, tr_1 and tr_2 are the motion parameter values in the (i,j) pixel.

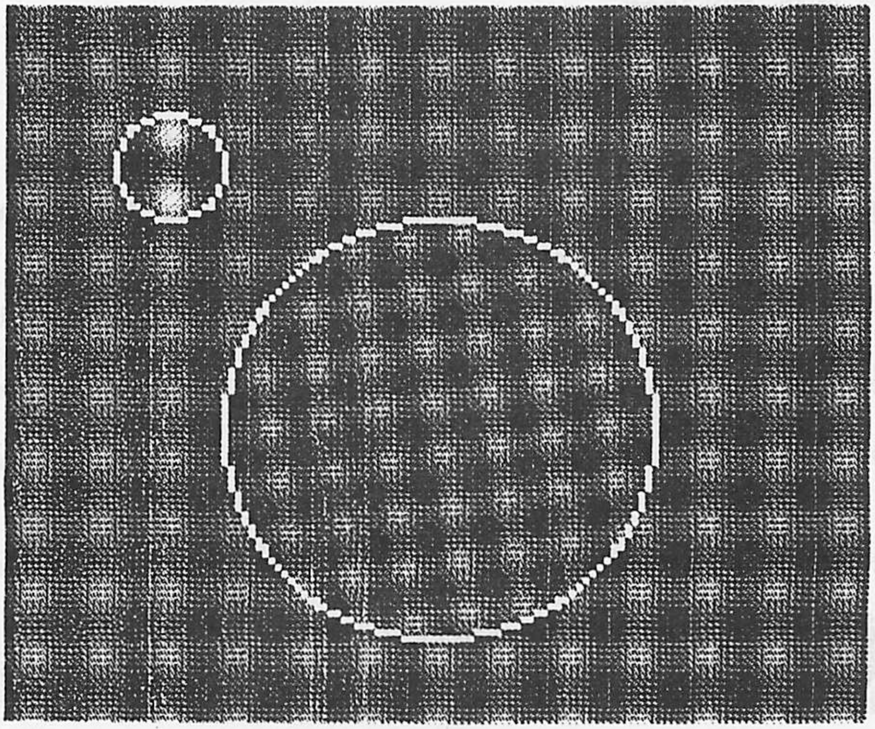
Figure 1 shows two pairs of artificial images which contain several independently moving objects. The motion parameters of each object are specified in tables 5.1 and 5.2. Figure 2 shows the result of applying the Horn and Schunck technique to these images.

The smoothness constraint is violated at the boundaries of independently moving objects. Therefore, the computed displacement values in these areas are incorrect. Fortunately, these areas can be detected by using the error function which represents the departure from the constraints. High values of the error function indicate that the constraints are not satisfied and the computed displacement values are unreliable.

For each displacement vector we compute an associated weight such that high reliability (low value of the error function) is represented by a value close to 1 and low reliability by a value close to 0. An appropriate relation between the error function, erf(i,j), and the weight, W(i,j), can be obtained by the function

$$(2.6) \qquad W(i,j) = e^{-erf(i,j)/k}$$

The parameter k was experimentally determined as 0.07. However, this value need to be decreased with noisier data. Figure 3 shows the weight planes computed for the displacement fields in figure 2. When the Hough transform is computed later the influence ('voting' power) of each displacement vector will be proportional to its associated weight.



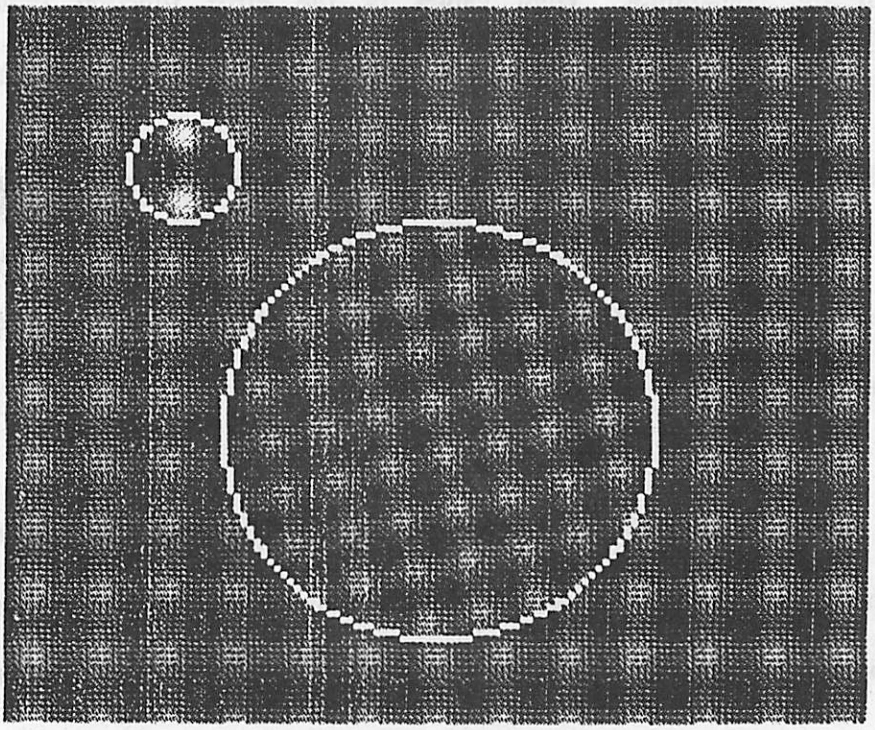
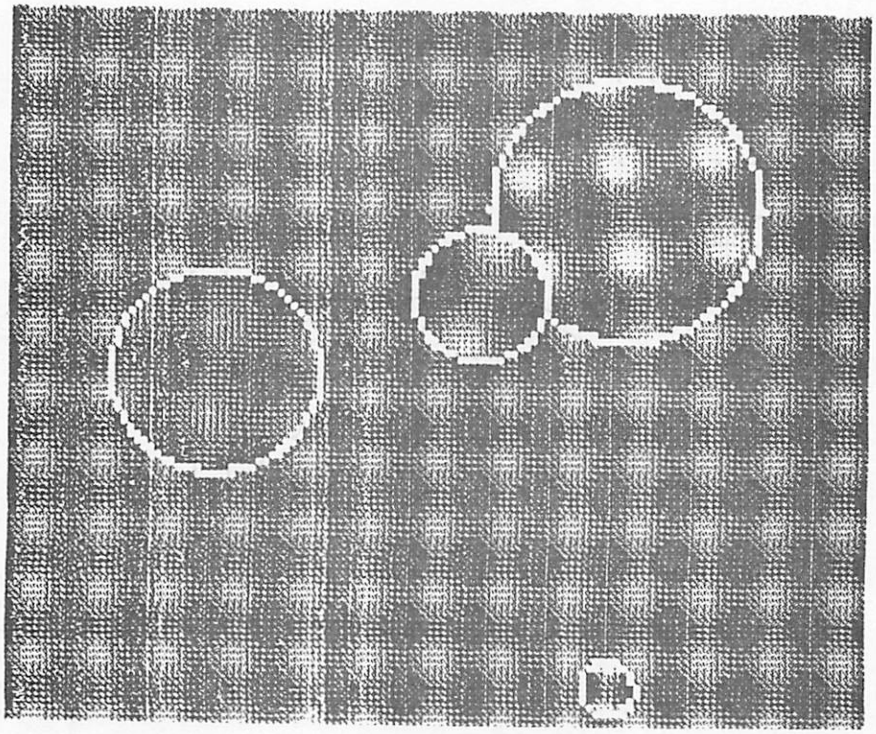


Figure 1.a: Intensity images used in the first experiment (the white lines only emphasize the contours of the objects and are not part of the images);
– object A is the background,
– object B is the large circle in the center of the image,
– object C is the small circle in the upper left corner.



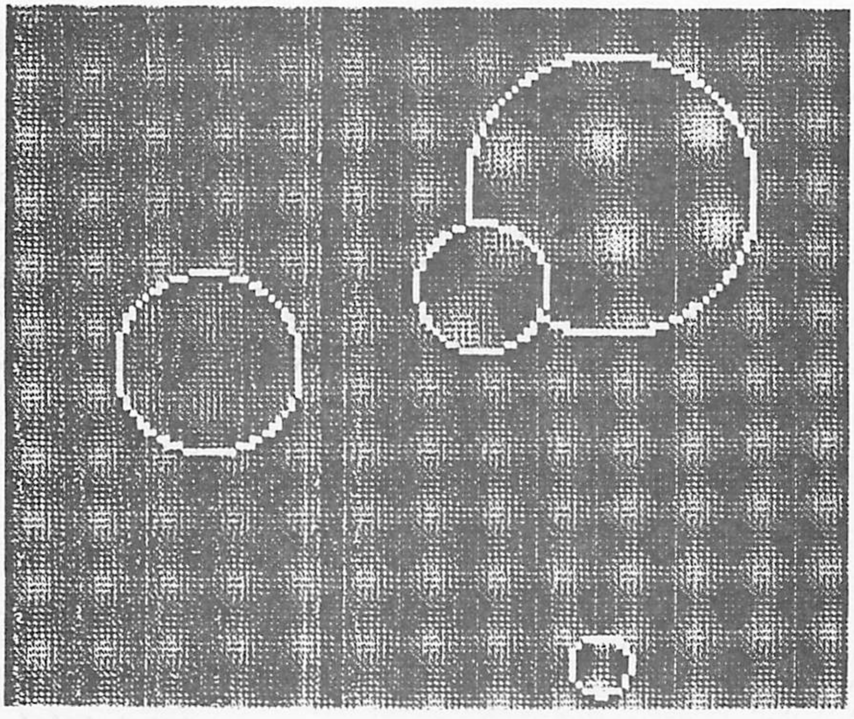
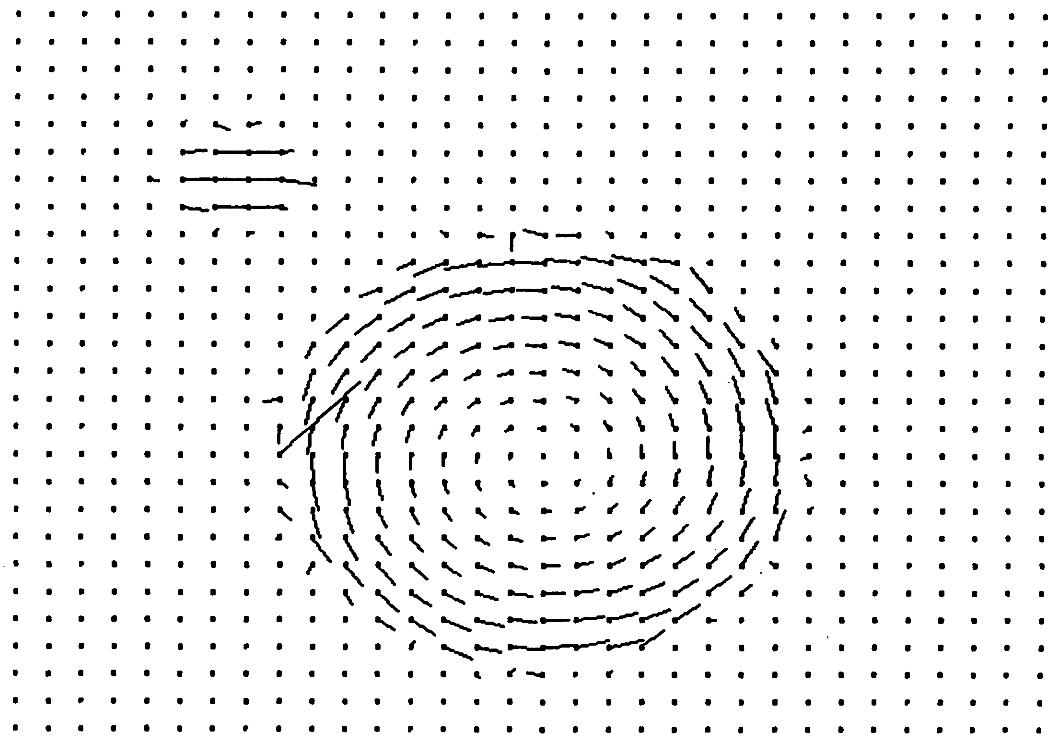


Figure 1.b: Intensity images used in the second experiment;
-- object A is the background,
-- object B is the circle in the upper right corner,
-- object C is the circle which partially occludes object B,
-- object D is the circle in the left part of the image,
-- object E is the small circle in the lower part.

(a)



(b)

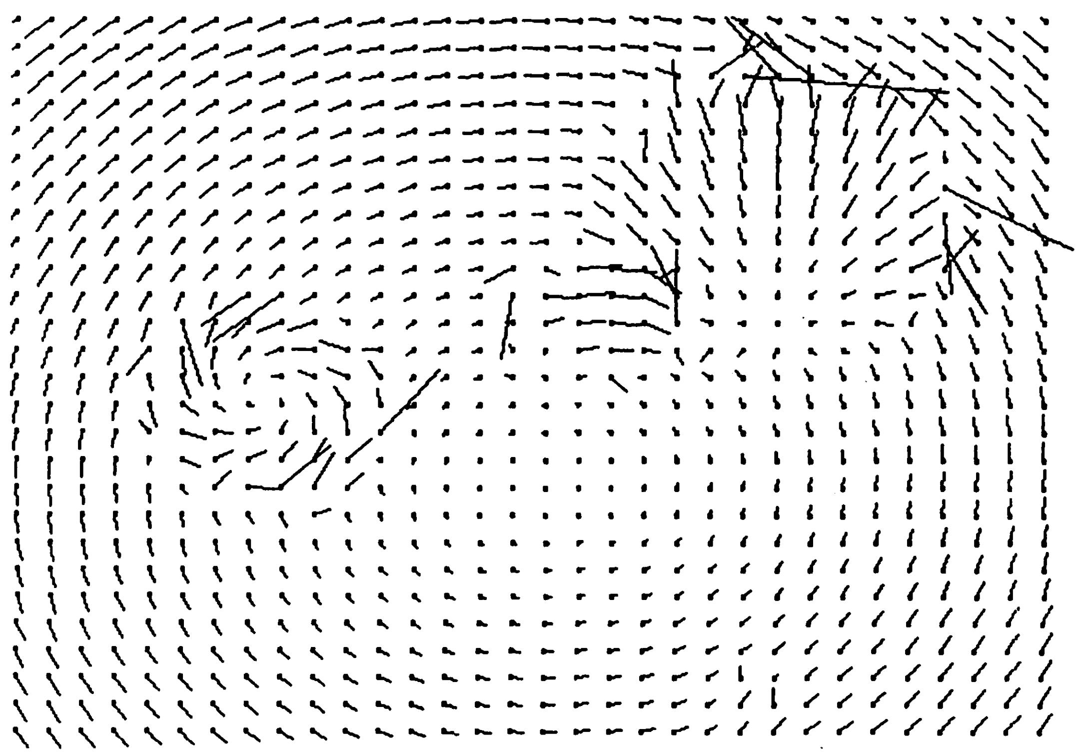


Figure 2: Samples of the displacement fields. (a) First experiment. (b) Second experiment.

(a)

(b)

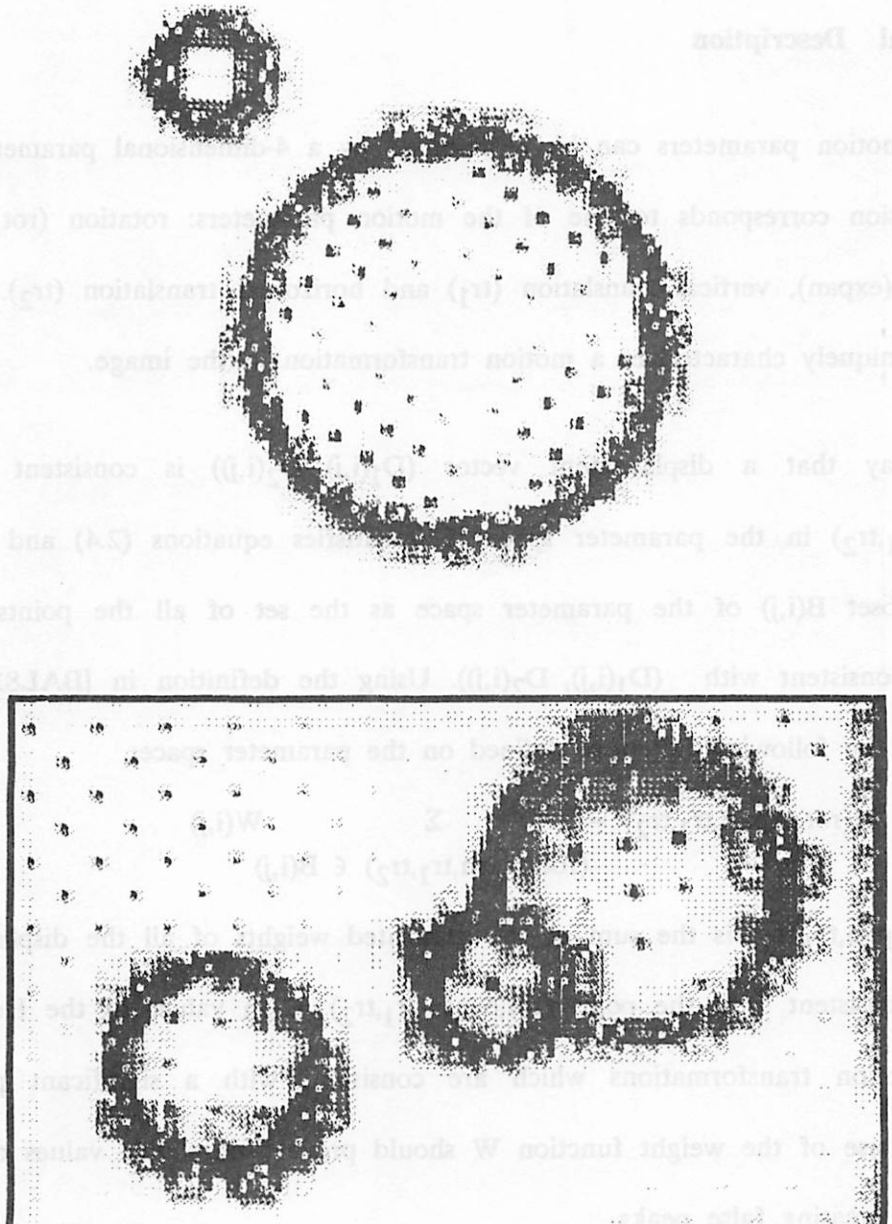


Figure 3: Weight planes. Note the correspondence between low values (represented by darker gray levels) in the weight planes and incorrect values of displacement vectors in the boundaries of the objects. (a) First experiment. (b) Second experiment.

3. The Generalized Hough Transform Technique

3.1 General Description

The motion parameters can be represented by a 4-dimensional parameter space where each dimension corresponds to one of the motion parameters: rotation (rot), expansion or contraction (expan), vertical translation (tr_1) and horizontal translation (tr_2). Each point in this space uniquely characterizes a motion transformation in the image.

We say that a displacement vector ($D_1(i,j)$, $D_2(i,j)$) is consistent with a point ($rot,expan,tr_1,tr_2$) in the parameter space if it satisfies equations (2.4) and (2.5). Let us define a subset B(i,j) of the parameter space as the set of all the points in this space which are consistent with ($D_1(i,j)$, $D_2(i,j)$). Using the definition in [BAL81a], the Hough transform is the following function, defined on the parameter space:

$$(3.1) \qquad H(rot,expan,tr_1,tr_2) = \sum_{(rot,expan,tr_1,tr_2) \in B(i,j)} W(i,j)$$

i.e., $H(rot,expan,tr_1,tr_2)$ is the sum of the associated weights of all the displacement vectors which are consistent with the point ($rot,expan,tr_1,tr_2$). High values of the Hough transform represent motion transformations which are consistent with a significant portion of the vectors. The use of the weight function W should prevent unreliable values of displacement vectors from creating false peaks.

In practice, we assume a limited velocity of objects, i.e. minimal and maximal values for each parameter. The corresponding intervals are quantized and thus each parameter is represented by a discrete set of values. The parameter space is the cartesian product of these sets.

For each displacement vector $(D_1(i,j), D_2(i,j))$ and each pair (rot,expan) of rotation and expansion, there exists exactly one pair of translations (tr_1^*, tr_2^*) which satisfies equations (2.4) and (2.5). If tr_1^* and tr_2^* are within the limits of the respective dimensions of the parameter space, then we can find exactly one pair (tr_1, tr_2) such that tr_1 and tr_2 are sampled values and

$$(3.2) \qquad tr_1 - res/2 \leq tr_1^* < tr_1 + res/2$$

$$(3.3) \qquad tr_2 - res/2 \leq tr_2^* < tr_2 + res/2$$

where res is the resolution of the translation variables in the parameter space. In this case we say that $(D_1(i,j), D_2(i,j))$ 'votes' for $(rot, expan, tr_1, tr_2)$, i.e., it contributes its weight to $H(rot, expan, tr_1, tr_2)$.

Finally, among the points whose Hough transform exceeds a given threshold in the parameter space, local maxima are found. These represent the hypothetical motions of objects in the image.

3.2 Reliability and Efficiency Considerations

The resolution of the parameter space should be determined by a few considerations: the signal to noise ratio (SNR), the required accuracy, the computation time and the required storage space.

For each independently moving object in the scene, the SNR in the parameter space can be defined as:

$$(3.4) \qquad SNR = \frac{\text{no. of votes for the object motion}}{\text{average no. of votes for each parameter value}}$$

(for a different definition see [BRO82]). If the SNR is low, then false peaks, higher than the value corresponding to the object, can be created. Thus the detection of the object's motion, by a straightforward Hough technique, may be difficult or impossible.

Let us assume that the multiplicative parameters of rotation and expansion are quantized into p_1 elements each and that the translation parameters are quantized into p_2 elements each. Then the parameter space includes $p_1^2 p_2^2$ elements. Let n be the number of displacement vectors which are considered in the computation of the Hough transform. According to the voting process described in section (3.1), for each displacement vector and each pair (rot,expan), there exits at most one pair (tr_1,tr_2) of translations such that the displacement vector votes for $(rot,expan,tr_1,tr_2)$. Assuming that tr_1 and tr_2 are likely to be within the limits of the respective dimensions (and that is the case in our experiments) we can estimate the average number of votes for each parameter value as $np_1^2/(p_1^2 p_2^2) = n/p_2^2$. If c represents the fraction of the image covered by an object, then it contains cn displacement vectors, where $0<c\leq 1$. Thus, we can estimate the SNR by:

$$(3.5) \qquad SNR \simeq \frac{cn}{n/p_2^2} = cp_2^2$$

If for reliable detection of the object, the SNR should be larger than some threshold t, then p_2 should satisfy the constraint $p_2 \geq \sqrt{t/c}$. For example, if t=10 and c=0.01 then p_2 should be at least 32. This observation also indicates that for a given p_2, the motion transformation of a small object may be difficult to detect.

The second consideration is the required accuracy. The parameter resolution can be dynamically modified to fit the expected constraints of the task domain. If, for example, the maximal value of rotation is 1/8 radian, the minimal value is −1/8 radian and the required resolution is 1/128 radian, then p_1 should be at least 32.

The third consideration is computation time. Computationally, the most expensive process is the voting process. We saw in section (3.1) that the basic operation in this process is computing tr_1, tr_2 for each displacement vector and each pair (rot,expan). Therefore, the voting process takes approximately snp_1^2 time units, where each basic operation takes s time units.

The fourth consideration is the required memory for the parameter space which includes $p_1^2p_2^2$ elements. If we combine the requirements for high SNR and high accuracy we may have

$$(3.6) \qquad p_1^2p_2^2 \geq 32^4 > 1000000$$

In such a large array, finding local maxima also becomes a computationally expensive task.

Finally, assuming that we want to obtain a given accuracy and we are given a storage space with a fixed size, the optimal values of p_1 and p_2 can be determined. Let us suppose that the image contains m^2 pixels and that the origin of the coordinate system is in the center of the image. Then, using a resolution of ϵ_1 in the multiplicative parameters of rotation and expansion can cause, at the boundary of the image, a displacement error of $m\epsilon_1/2$. Therefore, it is reasonable to quantize the parameter space in such a way that $m\epsilon_1 \simeq \epsilon_2$, where ϵ_2 is the resolution of translation. Consequently, if m is multiplied by 4, for example, then p_1 should be multiplied by 2 and p_2 should be divided by 2.

4. Computing Motion Parameters from Displacement Field Information

4.1 Key Ideas

The proposed method is intended to reduce the problems mentioned in the last section and to test mechanisms for solving such problems for even more difficult tasks, e.g. recovering the motion parameters of 3-D motion with six degrees of freedom.

The key ideas which are used for accomplishing this goal are the following:

1) Given a large displacement field (such as the 128×128 array in the experiments), we can compute the motion parameters of large objects by using a coarse resolution field. Such a field can be obtained by uniformly sampling the initial field. In this way, we can considerably reduce the computation time.

2) We can find the motion parameters of a given small object by locating a window around the object and applying the Hough transform only to the displacement vectors contained in this window. Such a window can be located by using a multipass approach (see next section). By focusing our attention to the window, we increase the proportion of the vectors contained in the object, i.e., we increase c in equation (3.5). We can now decrease p_2 and still find the motion parameters of the object. This technique enables us to detect small objects and save time and storage space.

3) Even with a limited memory size, we can find accurate parameter values by iteratively using the Hough technique. In each iteration we quantize the parameter space around the values estimated in the previous iteration, using a finer resolution. Other methods for reducing the required space in Hough techniques can be found in [ORO81, SLO81].

4.2 Description

4.2.1 General

The algorithm is based on repeatedly executing a basic cycle of operations. The input to each cycle includes:

1) A list L of motion transformations which were computed in previous cycles (initially L is an empty list).

2) A binary mask array A in registration with the displacement field. Each element in A is either 1 or 0: 1 if the corresponding displacement vector is consistent with one of the already computed motion transformations; 0 otherwise. Initially all the entries in this array are set to 0.

Each cycle is composed of the following steps:

1) Locate windows in the image which contain relatively dense clusters of 0-entries in A. Initially there is one window consisting of the whole image.

2) For each window compute the Hough transform and hypothesize (see section 4.2.3) the motion transformations.

3) Test, sequentially, the hypothesized transformations. The test is done by considering the 0-entries in A that are contained in the window, and summing the weights of the associated displacement vectors which are consistent with the hypothesized transformation. If the sum is higher than a given threshold, the transformation is confirmed. In this case it is added to the list L and the array A is updated correspondingly.

4.2.2 Locating Windows

A window can be described as a set $\{(i,j): i_0 \leq i < i_1,\ j_0 \leq j < j_1\}$. For implementation reasons, we consider only windows such that

$$i_0,\ i_1,\ j_0,\ j_1 \in \{0,4,8,\ldots,128\}$$

and

$$i_1 - i_0,\ j_1 - j_0 \in \{8,16,32,64\}$$

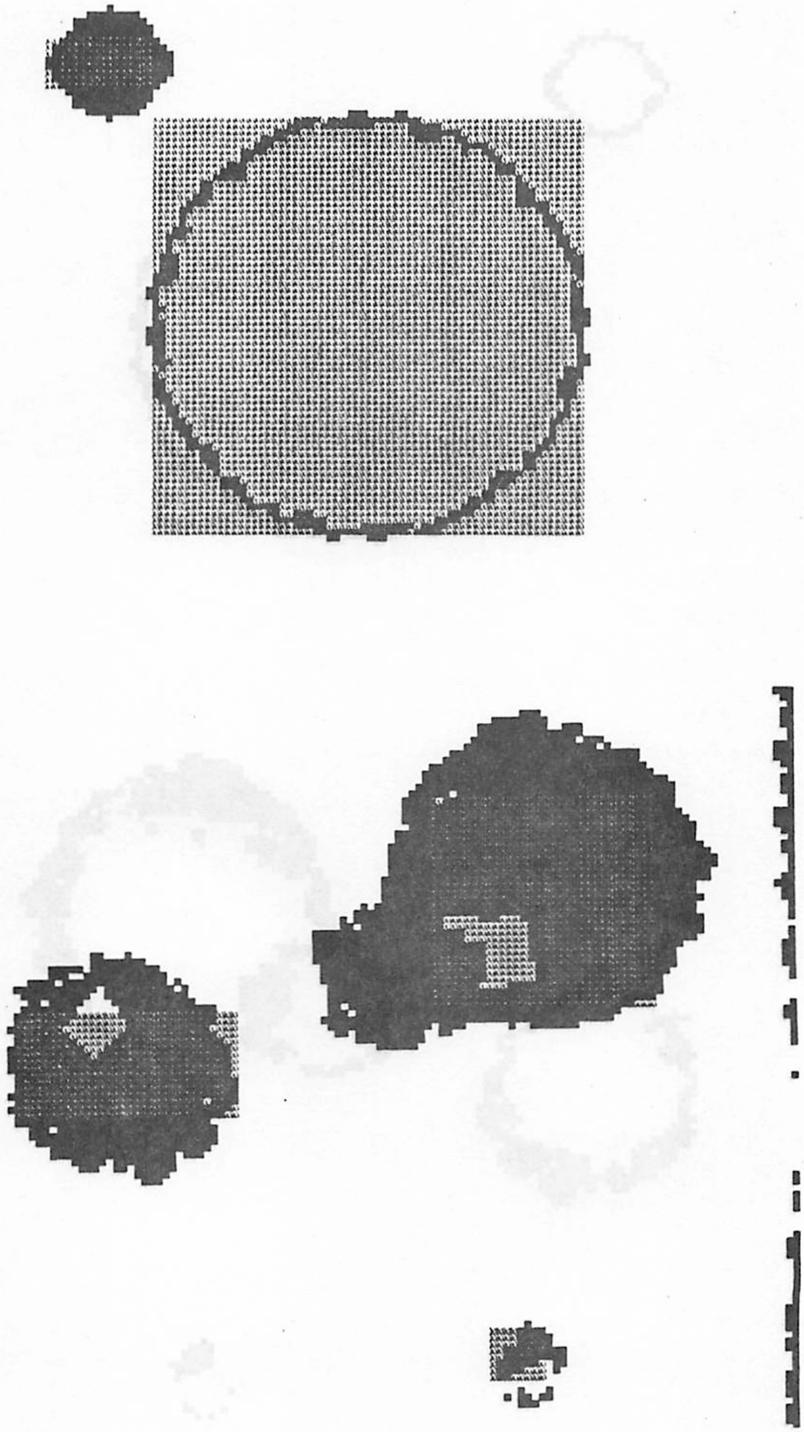
For each window, we define a criterion function CR by:

$$\text{(4.1)} \quad CR = \frac{\text{no. of 0-entries of A in the window}}{\sqrt{\text{area of the window}}}$$

We look for windows with high values of CR. Such windows contain dense clusters of 0-entries in A. The use of square root in the denominator of equation (4.1) means that this density can be lower as the window becomes larger. If we would eliminate the square root in this equation, then too small windows, which contain only 0-entries, would be chosen. If we do not find any appropriate windows, i.e. windows whose criterion function exceeds a given threshold, the process is stopped. The reason is that probably there are no more objects whose motion transformation has not already been found.

Figure 4 shows the windows found in the second cycle when the method is applied to the images in figure 1. Figure 5 shows the A arrays when the processes are stopped. The black areas in figure 5, which represent the 0-entries in the A arrays, correspond to incorrect values of displacement vectors in the boundaries of the objects.

(a)



(b)

Figure 4: Optimal windows found in the A arrays during the second cycle of the experiments. (a) First experiment. (b) Second experiment.

(a)

(b)

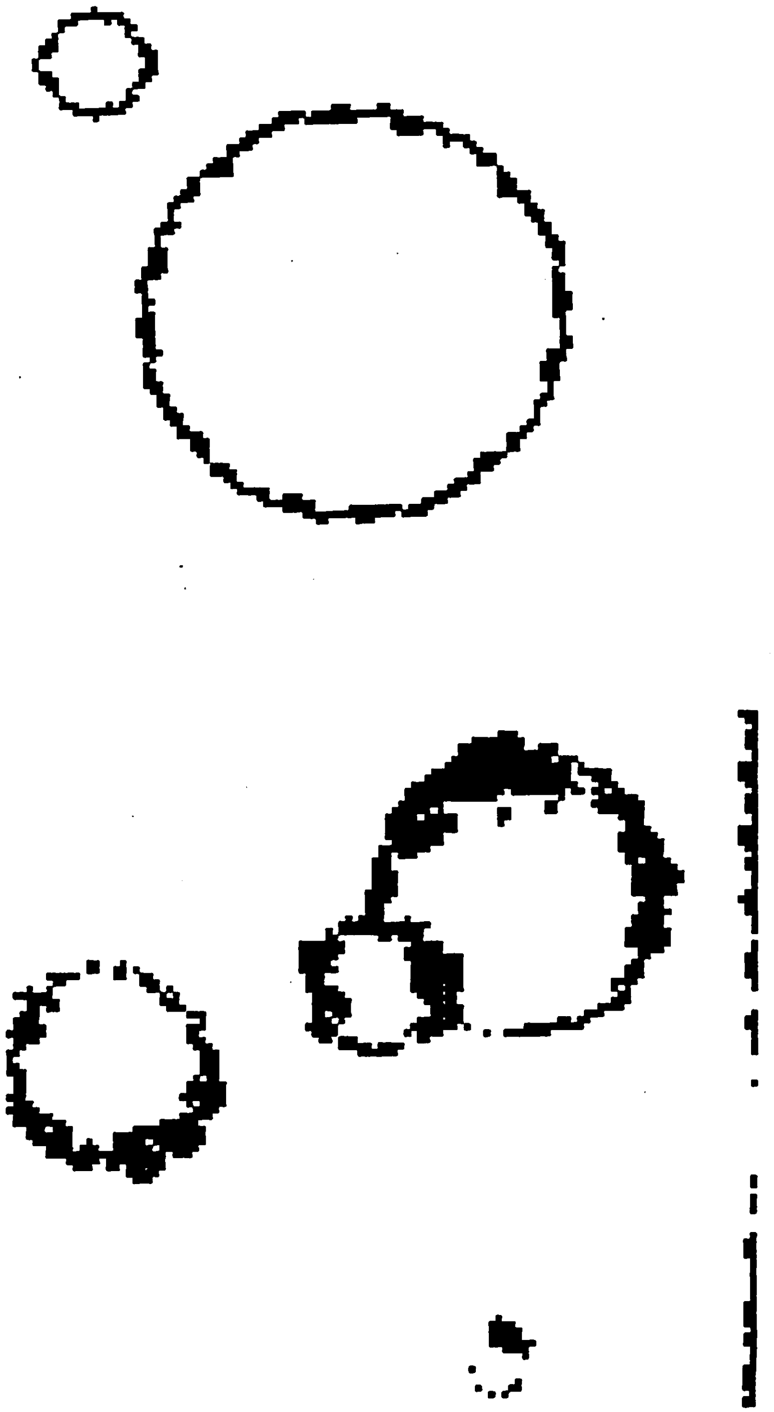


Figure 5: Final A arrays. (a) First experiment. (b) Second Experiment.

4.2.3 The Hypothesizing Phase

Before we start the voting process of the displacement vectors in a given window, we have to decide which vectors take part in this process and how the parameter space is defined. If the window contains no more than 1024 elements, then all of them take part in the voting process. Otherwise, for efficiency considerations (see section 4.1), we will utilize a uniformly sampled subset of 1024 elements. For example, if the window is 32×64 elements, we define a sub-array of 32×32 elements by choosing all the elements (i,j) such that j is even.

The parameter space is an adjustable 4-D array ('adjustable' means that the number of elements in each dimension is not fixed) which contains 17^4 (≈90000) elements. We assume that the rotation is limited to 0.125 radians in each direction, the expansion (or contraction) is limited to 0.125 and the translation is limited to 8 pixels in each direction. The axes which correspond to rotation and expansion contain p_1 elements each and the axes which correspond to the translations contain p_2 elements each. If the length of the window is at least 64 elements then, according to the argument described at the end of section 3.2 for equalizing the effective parameter resolutions, we choose $p_1=p_2=17$; otherwise p_1 is decreased and p_2 is increased. So, for example, if the window is 16×16, we choose $p_1=9$ and $p_2=31$.

After the voting process is finished, local maxima in the Hough transform are determined. From these candidates, the ones that exceed a given threshold are selected. The threshold is proportional to the number of all the voting displacement vectors. If the resolution of the translation axes is more than 1/2 pixel, we define a new parameter space, around each maxima point, with finer resolution. We then recompute the Hough transform. The process is repeated until we achieve a resolution of 1/2 pixel at most. At the end of this process we have a set of hypothesized transformations.

4.2.4 The Testing Phase

In this phase we sequentially test the hypothesized transformations in the order of their Hough transform values in the parameter space. We test only the transformations which are still not included in the list L of confirmed transformations. When we test a given transformation, we check all the displacement vectors with associated 0-entry in the corresponding window. We sum the weights of such vectors which are consistent with the hypothesized transformation. We also compute a threshold proportional to the number of 0-entries in A, contained in the window. If the sum is higher than the threshold, we accept the transformation, add it to the list L and update the array A. In the current implementation, the process is stopped if we do not accept any transformation in any of the windows.

5. Experiments

We performed two experiments based on two pairs of 128×128 artificial images (figure 1). In the experiments the objects were transformed according to the upper values in each entry in tables 5.1 and 5.2. The lower numbers in these entries are the computed parameters.

	rotation (radians)	expansion	vertical translation (pixels)	horizontal translation (pixels)
object A				
actual	0.	0.	0.	0.
computed	0.	0.	0.	0.
object B				
actual	0.07	0.	0.	0.
computed	0.0781	0.	0.	0.
object C				
actual	0.	0.	0.	2.
computed	0.	0.	0.	2.

table 5.1 - first experiment

In the first stage – the displacement field was computed (figure 2). Note the errors at the boundaries of the objects which correspond to low values in the weight planes (figure 3).

In experiment 1, during the first cycle of the algorithm for computing the motion parameters, the motion transformations of objects A and B were detected (see the results in table 5.1). In the second cycle, two windows were located around areas in the mask array A with relatively dense clusters of 0-entries (figure 4), but only the window around object

C gave a positive result – the motion transformation of object C. In the final cycle no appropriate windows were found in the array A (figure 5).

Corresponding results from experiment 2 are also shown in figure 4 and figure 5. The computed transformations are shown in table 5.2.

	rotation (radians)	expansion	vertical translation (pixels)	horizontal translation (pixels)
object A				
actual	0.025	0.	0.	0.
computed	0.0234	0.	0.	0.
object B				
actual	0.	0.1	–1.5	0.
computed	0.	0.09375	–1.2	0.
object C				
actual	–0.1	0.	0.	2.2
computed	–0.09375	0.	–0.2	2.1
object D				
actual	0.12	–0.1	0.	0.
computed	0.125	–0.0937	–0.2	0.
object E				
actual	0.	0.125	–1.1	0.7
computed	0.	0.0625 (*)	–1.05	0.6

table 5.2 - second experiment

(*) The large error indicated in this entry is due to the small size of object E (radius = 8 pixels) which reduces the possible resolution in the measurements of rotation and expansion.

6. Conclusions and Extensions

This work demonstrates an efficient and robust algorithm, based on the Hough technique, for recovering motion parameters in scenes containing several independently moving objects. An hierarchical approach, combined with a windowing scheme, is implemented in order to deal with objects of different size. The storage space and computation time can be limited, while still computing the motion parameters very accurately and distinguishing between real objects and noise effects.

We hope to extend this work for sequences of images and for recovering the 3-D motion parameters of rigid objects. However, the latter task is much more difficult than recovering 2-D motion parameters. In the 2-D case each vector contributes two constraints (equations 2.4 and 2.5) whereas in the 3-D case, assuming that depth information is unknown, each vector contributes only one constraint. Therefore, the signal to noise ratio in the parameter space (section 3.2) is much lower. In addition, we expect to have problems of ambiguity in the interpretation of noisy displacement fields, where a group of motion transformations can be equally consistent with the data. In such cases a probabilistic approach might be more suitable. We also plan to implement less restricted methods for computing a displacement field or other equivalent information, to use the motion information for object-surround separation, and to test the method in real scenes.

Acknowledgements

I would like to acknowledge the support of Ed Riseman, Al Hanson and the VISIONS group at UMASS. I am especially grateful to Daryl Lawton and P. Anandan for their generous help and many fruitful discussions. Ed, Al and P. Anandan made useful suggestions for improving this paper. This research was supported by DARPA Grant N00014-82-K-0464 and ONR Grant N00014-75-C-0459.

References

[BAL81a] Ballard, D.H., "Parameter Networks: Towards a Theory of Low-Level Vision", Proc. of the 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., U. Rochester, 1981.

[BRO82] Brown, C.M., "Bias and Noise in the Hough Transform I: Theory", TR105, Computer Science Dept., U. Rochester, 1982.

[FEN79] Fennema, C.L. and Thompson, W.B., "Velocity Determination in Scenes Containing Several Moving Objects", Computer Graphics and Image Processing, Vol. 9, 1979.

[HOR80] Horn, B.K.P. and Schunck, B.G., "Determining Optical Flow", MIT A.I. Memo 572, 1980.

[JAY82] Jayaramamurthy, S.N. and Jain, R., "Segmentation of Textured Scenes Using Motion Information", Proc. of the Workshop on Computer Vision, Rindge, N.H., 1982.

[ORO81] O'Rourke, J., "Motion Detection Using Hough Techniques", Proc. of Pattern Recognition and Image Processing, Dalas, Texas, 1981.

[SLO81] Sloan, K.R., "Dynamicallly Quantized Pyramids", Proc. of the 7th IJCAI, Vancouver, Canada, 1981.

[THO81] Thompson, W.B. and Barnard, S.T., "Lower-Level Estimation and Interpretation of Visual Motion", IEEE Computer, August 1981.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER COINS TR 83-11	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RECOVERING 2-D MOTION PARAMETERS IN SCENES CONTAINING MULTIPLE MOVING OBJECTS		5. TYPE OF REPORT & PERIOD COVERED INTERIM 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gilad Adiv		8. CONTRACT OR GRANT NUMBER(s) DARPA N00014-82-K-0464 ONR N00014-75-C-0459
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer and Information Science Department University of Massachusetts Amherst, Massachusetts 01003		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE May 1983 13. NUMBER OF PAGES 28
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Multiple moving objects
Hough transform
Recovering motion parameters
Displacement field

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A method for extracting the motion parameters of several independently moving objects from displacement field information is described. The method is based on a generalized Hough transform technique. Some of the problems of this technique are addressed and appropriate solutions are proposed. A modified multipass Hough transform approach has been implemented, where in each pass windows are located around objects and the transform is applied

only to the displacement vectors contained in these windows. The windows are determined by the degree to which the displacement field is locally inconsistent with previously found motion transformations. Thus, the sensitivity of the Hough transform to local events is increased and the motion parameters of small objects can be detected even in a noisy displacement field. We also use a multi-resolution scheme in both the image plane and the parameter space and thus reduce the computational cost of the technique. The method is demonstrated by experiments based on artificial images with four parameters of 2-D motion: rotation, expansion and translation in both axes.