Estimation of Image Motion Using Wavefront Region Growing

Bir Bhanu and **Wilhelm Burger**

Department of Computer Science
University of Utah
Salt Lake City, UT 84112

Abstract- This paper presents a novel approach for the computation of displacement fields along contours of moving regions, where distinct point features are difficult to maintain. The proposed algorithm is simple and lends itself to parallel implementation. Individual frames of the image sequence are treated one at a time, and the segmentation and motion analysis are performed simultaneously. For the first frame an original segmentation of the image into disjoint regions is assumed to be given in the form of pixel markings and the properties of these regions. The analysis of each new frame consists of *(a)* finding the new segmentation and *(b)* a set of displacement vectors that link corresponding points on the original and the new contour. The new region is assumed to overlap with the original region, such that their intersection is not empty. After finding this intersection, wavefront region-growing is applied to obtain the new region. A relaxation-type algorithm is then used to determine the shortest links between subsequent boundaries to approximate the displacement field. Experiments on synthetic images show, that results compare favorably with work done by Hildreth[5], yet following a simpler and more realistic approach.

Index Terms- Displacement Fields, Motion Analysis, Region Growing, Segmentation, Tracking.

INTRODUCTION

Motion Analysis is concerned with the reconstruction of an object's 3-D motion parameters from a dynamic scene, given a series of two-dimensional projections.[1,8] From the *apparent motion* of a sufficient number of points on each moving object its actual three-dimensional rotation and translation components can be determined, assuming that the objects involved are rigid. Here we address the problem of how to obtain the apparent motion from s sequence of two-dimensional images. Two main approaches have been used to compute the Optical Flow Field or Displacement Field from a given motion sequence of grey-level images.

The *Gradient Method* [6] uses spatial and temporal grey-level variations to estimate the *instantaneous velocity* at each pixel in the image. It relies on sufficient object texture, continuous motion and small displacements between subsequent frames. Since the magnitude of flow can only be determined in the direction of the spatial gradient (perpendicular to the tangent of the boundary), the flow vectors cannot be computed locally. Global smoothing of the flow field has been proposed, which gives rise to problems at flow discontinuities, such as object boundaries. Paradoxically, motion estimates should be obtained most easily at exactly those locations.

The *Displacement Method* uses the parts of the image, where discontinuities in brightness or motion occur (which give trouble in the gradient method). Significant features such as line segments or distinguished ("interesting") points in two consecutive frames are selected and matched, rendering a field of *displacement vectors* for the selected features. Two problems arise during this process: one is the selection and location of significant image features, especially when the images are noisy; the other problem is finding an optimal match between them. This is commonly referred to as the Correspondence Problem.[3,9]

We propose a solution which lies between these two methods, one that combines the *implicit matching* process of the gradient method and the *locality* of the displacement method. One such approach has been suggested by Hildreth [5] , where an approximation of the actual displacement field of moving closed contours is sought in two steps: first the displacement vectors perpendicular to the original contour are determined, and second the resulting flow fields along the contour are smoothed. Experiments indicate, that the approximations are close to the motion *perceived* by the human visual system. The problems arising from this approach are twofold.

First, since the perpendicular components of the flow vectors are obtained using the Gradient Method mentioned above, the resultant vectors will be susceptible to errors in direction as well as in length. Estimating the perpendicular *direction* is done by determining the direction of the edge encountered. This is difficult due to noise and limited spatial resolution. Estimation of vector *magnitude* from the brightness gradients is inherently unreliable in cases of missing or too fine texture and/or displacement that exceeds the range of approximately linear gradient.

Second, the final result of this approach gives at best an approximation to the motion that *humans* would perceive, including some forms of illusionary motion. This is an important aspect for understanding human vision, but it is not the prime goal of quantitative motion analysis as pursued here.

The method proposed here is region-based and makes use of the fact, that (assuming sufficient sampling in time) corresponding regions in two subsequent frames will overlap, thus giving the initial cue for correspondence. In contrast to finding corresponding boundary segments, the direction of search is implicitly given by the assumption of overlapping regions. The intersection of the old and the new region serves as the *seed* [7] for a *region-growing* process, which renders a segmentation for the new image frame. Region-growing is done layer-by-layer very much like a *wavefront* to keep the overall region consistent. By propagating *shortest-distance* information an approximate mapping between the two boundaries is obtained. This mapping between boundary points is smoothed and subsequently *rotated* until a region-correspondence with *minimum deformation* is found. Experiments show that the final results approximate the actual displacement fields closely, even for extreme displacements.

This approach goes beyond pure motion analysis as it provides a *dynamic segmentation* scheme as well. Each frame is segmented by region-growing, using the previous segmentation as a starting point, while displacement data are computed simultaneously. This is an important benefit in Dynamic Scene Analysis and Understanding, where both segmentation *and* motion estimates are essential.

APPROACH

The input assumed to be given is a sequence of digitized images, representing a time-varying scene. The time interval between consecutive images is furthermore assumed to be sufficiently small, such that the condition of overlapping regions is met. This of course depends upon the granularity of the segmentation used and thus also upon the region pro-

CH2465-3/87/0000/0428$01.00 © 1987 IEEE

perties that govern the segmentation process. An initial segmentation is supposed to be given, which is constantly updated while frames are processed successively as part of this *dynamic segmentation* scheme. At this point no attention is paid to the problem of how this initial segmentation is obtained. This could either be accomplished using well developed segmentation techniques, or it could become an integral part of the proposed algorithm.

During dynamic segmentation it might well occur, that certain regions vanish due to occlusion or when they move out of sight. Similarly new regions are created when objects move into the field of view. The case of occlusion does not pose a problem, since we can assume the shape of a region will not change dramatically between two frames, and the vanishing of a region is easily detected. Newly created regions can be handled by the same process that provides the initial segmentation. Here we concentrate on the problem, how an established segmentation is *carried over* from one frame to the next while extracting motion data at the same time.

The suggested approach of approximating displacement vector fields consist of the the following steps:

(1) Update the given segmentation by *growing* each individual region onto its corresponding region in the next frame.

(2) For every region in the scene compute a set of *displacement vectors* which links the boundary points of the original region to corresponding points on the boundary of this region in the next frame. This again is done in two steps:

(a) Get an initial estimate for the displacement vectors by establishing a *tentative correspondence* between the two contours. Here the *closest neighbors* on the opposing boundaries are selected.

(b) Improve the initial match by looking for an *optimal* correspondence that implies *minimal deformation* of the region between the two instances of time.

Segmentation and finding the closest neighbors are accomplished simultaneously in one computational step, using a purely local technique. The result is the new segmentation and a relation in the form of pairs of coupled boundary points. The algorithm lends itself naturally to *pipelined* and *VLSI* implementations, making real-time operation feasible. The optimal correspondence is found by *rotating* one of the boundaries until the *minimum deformation* is observed. The sum of differences between corresponding *diameters* is used as the measure for deformation. Details are given in the remainder of this section.

Wavefront Region Growing

This first step of the algorithm operates on a given segmentation for the current frame and finds the new segmentation for the following frame. In addition to that, a *tentative correspondence* between the contour points of the two related regions is determined. Both tasks are accomplished simultaneously in a *relaxation*-type fashion, where the image is scanned iteratively and one new *layer* is added to each region during each iteration. The growing region as well as the displacement data propagate similar to a *wavefront* during this process.

Seed. First the *intersection* between the old and the new region is determined, which is non-empty since we assume that regions overlap in successive frames:

```
find_intersection:
    given:
        S1 ... a segmentation into disjoint regions at time t
        I2 ... the input frame at time t+1
    for all image points (x,y) do
        if I2(x,y) is consistent with Region[S1(x,y)] then
            mark S1(x,y) as new
        end_if
    end_for
```

Region Growing. The intersection serves as the *seed* to grow onto the new region by acquiring points that are consistent with the properties of the region.

Starting with the intersection, one new layer of consistent image points is added to the current region during each iteration. The region-growing process stops when no points could be added during an iteration. At this point the new segment covers the corresponding image region in the new frame completely. During each iteration the following update is performed:

```
add_layer:
    for all image points Xdo
        if X is marked new then
            for all 8 neighbors N do
                if N is consistent with Region[S1(X)] then
                    mark Region[S1(N)] := Region[S1(X)]
                end_if
            end_for
        end_if
    end_for
```

Displacement Propagation. During the region-growing process, displacement data are propagated into the newly created parts of the region, such that each point in this region holds information about the location of the closest point in the original intersection. When *multiple* points grow into the same image location, the closest-neighbor data from all the sourcing points will determine the closest neighbor of this new point (Figure 1).

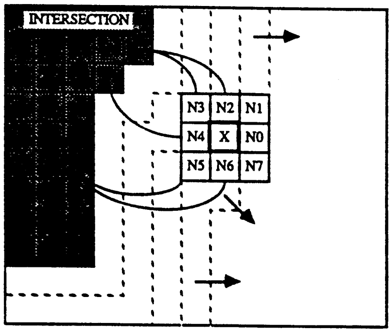


Figure 1. Propagation of displacement data. The shortest distance to the original intersection is determined for every point, from the displacements of its 8 neighbors. Assume that the shortest displacement for point X is to be found. For its neighbors N2..N6 the shortest displacements are already known, while N0, N1, N7 have no displacement values assigned yet. The shortest path from the intersection to X will thus go through N2, N3,.. or N6.

In areas where the intersection can *grow* onto the new region, region growing and displacement propagation are done simultaneously. In areas of the original region, which are *not* part of the intersection, displacement data are propagated *backwards*, until the original boundary is reached (Figure 2).

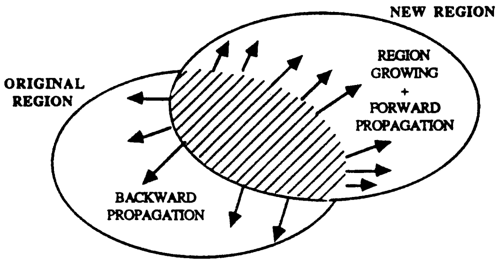


Figure 2. Propagation of displacement data. From the intersection of the original region and the new region, displacement data are propagated *forward* where the region can grow, and *backward* onto the original region.

The entire image can be viewed as a *connected graph*, where each *node* corresponds to a pixel which is connected to all the neighboring points that are members of the same region. Each node in the graph holds information about the closest point on the original intersection. Those nodes lying on the intersection are initialized as referring to themselves, their displacement from the intersection is zero. The problem can thus be stated as finding the shortest distance from one node (on the intersection) to all other nodes of the graph. This is well known in graph theory as the *Shortest Path Problem* (Dijkstra algorithm[2]), which can be applied here immediately. The only difference from this classical problem is that, due to the region-growing process, new neighborhoods (and thus links in the graph) are established successively. The graph becomes stable when no further changes in the nodes can be made. As a consequence the entire image must be scanned and displacement data propagated until all displacements have settled *(relaxed)* to stable values. Although this might appear computationally expensive on a conventional (serial) computer, this technique is well suited for *pipelining* and *VLSI* implementation, where high regularity of computation is an important requirement.

```
propagate_displacement:
  for all image points X do
    for all 8 neighbors N of X do
      if disp (N) + dist (N,X) < disp (X) then
        disp (X) := disp (N) + dist (N,X)
        end_if
      end_for
    end_for
```

where
disp (X) ... displacement of X from the closest point on the intersection
dist (N,X) ... distance between points N and X.

Result. After the process of region-growing and displacement propagation has terminated, the boundary points of the *union* of the old and the new region carry pointers to the closest points on the intersection. From this information a *correspondence relation* C is computed, consisting of pairs of boundary points:

Correspondence Relation C:

$$C = \{ (X,Y) \mid X=(x_1,y_1) \in B1 \text{ , } Y=(x_2,y_2) \in B2 \}$$

where
B1 ... original region boundary B2 ... new region boundary.

This relation represents a *mapping* of the original boundary onto the new boundary of the region. Notice that one point in the original boundary may have several corresponding points on the new boundary and vice versa, while some points on either boundary are not linked to any other points at all. The finite spatial resolution exaggerates this fact, since corner points on a jagged boundary are likely to be closer to other objects than their neighbors. *Smoothing* (see below) is applied to greatly reduce this effect and obtain a more uniform distribution of linked boundary points. Still this *Shortest-Distance Approximation* is "well-behaved", in the sense that the displacement vectors do not cross over, a fact that will be useful in the following section.

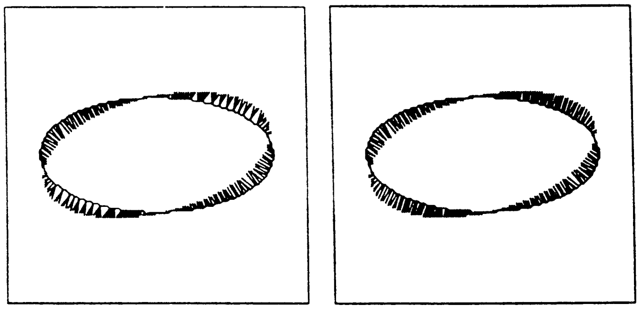
Optimal Correspondence

Given the tentative correspondences between the old and the new region boundaries as described in the previous section, we try to obtain a more realistic set of point-to-point relations. In general, the initial approximation by selecting the *nearest neighbor* on the opposite contour is not a good estimate for the actual displacement vectors. For instance, displacement caused by translation in the direction of the boundary is not detected, because the estimated displacement vectors are zero at these points. This has been termed the *Aperture Problem* in motion analysis.

A better correspondence relation is obtained by *smoothing* to remove the effects of finite spatial resolution and to distribute matched points more uniformly along the boundaries. The smoothed correspondence is then modified by *rotating* the new boundary in order to find an optimal correspondence. An *error function* is used, which indicates the amount of *deformation* applied to the region under the given boundary mapping.

Smoothing. The *Closest Neighbor Approximation* is computed, following the steps covered in the previous section. As mentioned earlier, individual points in this relation may have several corresponding points on the other boundary, while other points in the neighborhood may not be matched at all. The first step tries to remove these clusters by spreading out the points uniformly along the boundary. The approach is similar to techniques used for histogram equalization.[4]

A *local* smoothing technique is applied, which iteratively traverses the correspondence relation and dissolves clusters of boundary points on both sides of the relation. The smoothed relation is available after a few iterations, in fact only *two* iterations were applied in the actual experiments. The following outline of the smoothing algorithm provides separate steps for smoothing each side of the correspondence relation for the sake of clarity. The two steps could easily be done in parallel. The results of the smoothing algorithm applied to an initial approximation is shown in Figure 3.



(a) *(b)*

Figure 3. Smoothing of the Correspondence Relation in two steps: After equalizing the density of endpoints on the new boundary *(a)*. After equalizing the density of endpoints on both boundaries *(b)*.

The displacement field between the old (outlined) and the new (dashed) boundary is shown for an ellipse undergoing rotation. First the relation is smoothed by spreading points on the new boundary (Figure 3.a), then the original boundary is smoothed (Figure 3.b).

Error Function. The error function should be an indicator on how good the present set of point-to-point correspondences represents the actual set of displacement vectors. Since the actual displacement vectors are *unknown*, we cannot expect to find an objective error function task without applying additional restrictions. Previous approaches [5] have used the constraint of "*smoothness of flow*" either as a global restriction or along boundaries. As mentioned earlier, global smoothness of flow cannot cope with motion discontinuities which occur at object boundaries. Also a smooth vector field along a region boundary is not a good approximation in general, such as in the case of pure rotation (Figure 5).

The error function that we apply is supposed to quantify the *amount of deformation* that the region would undergo with a given set of point-to-point correspondences. The correspondence that results in the *least deformation* of the region is chosen as the closest approximation to the real situation. The measure of deformation is based on the differences of *diameters* across the region. If no deformation has occurred, then the distance between one point-pair on the old region should be the same as the distance between the corresponding point-pair on the new region. Pairs are selected such that their points lie approximately *opposite* to each other on the regions' boundaries. For all point-pairs that correspond on the old and the new boundary, the resulting diameters are compared. The sum of the squares of their differences is taken as the error measure:

Given:
B1,B2 ... two ordered sets of boundary points with approximately equal number of elements
C ... a mapping B1 → B2: $C = \{ (P,P) \mid P \in B1, Q \in B2 \}$

the error function is defined as

$$E_d(C) = \sum_{\substack{(P,Q),(P',Q')\in C \\ \text{opposite}(P,P')}} [\, d(P,P') - d(Q,Q')\,]^2$$

The diameter *d* is defined as

$$d(P,P') = [\, (x_{\bar{p}} - x_p)^2 - (y_{\bar{p}} - y_p)^2 \,]^{½}$$

To obtain a quantitative estimate of the goodness of fit for the selected correspondence, this error measure is normalized as shown in Table I. Minimal deformation as an indicator for the optimal correspondence will produce satisfactory results as long as there are no dramatic changes in the shape of a region between two frames.

Modification Rules. After defining the criterion to guide the search for an optimal correspondence mapping we define rules to select candidate mappings out of the many different mappings possible. It turns out that the search space of suitable mappings can be reduced considerably by making use of the implicit order of the set of boundary points. The initial *shortest-distance* approximation has the property, that displacement vectors do *not* crossover, which means that the order of pairs of points on one boundary (for instance clockwise) will be the same for the corresponding points on the other boundary. We term this property of a mapping between two closed boundaries as *radial*:

Given:
B1,B2 ... two ordered sets of boundary points representing two closed boundaries.

A mapping C: B1 → B2 is called *radial*, iff

for all $(P_i,Q_i),(P_j,Q_j),(P_k,Q_k) \in C$: *ordered* $(P_i,P_j,P_k) \rightarrow$ *ordered* (Q_i,Q_j,Q_k) ,

where *ordered* (P_i,P_j,P_k) means, that points P_i, P_j, p_k lie on boundary B in (clockwise) order.

This condition must hold for the optimal mapping as well, so we never need to investigate *permutations* of the initial mapping. Among all the other remaining possible mappings we select those that can be found by a *cyclic shift*. This means that the maximum number of mappings to be considered equals the number of boundary points.

The optimization problem can thus be stated as:

Given:
B1,B2 ... two ordered sets of boundary points with approximately equal number of elements
C_0 ... a *radial* mapping B1 → B2

Find a *radial* mapping C_{opt}: B1 → B2, such that

$E_d(C_{opt})$ is a minimum for all *radial* mappings C': B1 → B2

This means that we only have to *rotate* the correspondence relation until the mapping of minimal deformation is found. In practice it is sufficient to restrict the amount of cyclic shifting to a limited neighborhood of the original estimate. A shift of ±1/4 the length of the boundary (as used in the experiments) will include the optimal solution in most practical cases. For each cyclic displacement the deformation error of the corresponding mapping is evaluated. From all the inspected mappings the one that results in the minimal deformation is selected. The associated set of displacement vectors is taken as an estimate for the actual displacement field. Results of this algorithm obtained from simple (elliptical) moving regions are given in the following section.

RESULTS

The selection of the type of objects and motion used was influenced by the work of other researchers in the Motion Analysis community. Thus all experiments were conducted with ellipses undergoing translation and rotation in 2-D space. This allows some comparison with results obtained by Hildreth.[5]

Experiments

Using elliptical regions, four different types of image motion were investigated: translation only, rotation only, translation and rotation, and extreme rotation.

Table I lists the number of rotation steps and the resulting (normalized) deformation error for the various cases of motion. As expected, the number of region-growing iterations (*waves*) depends directly upon the amount of displacement. Thus the (2-D) *velocity* of image features determines the necessary amount of computation. For our experiments *(1)* and *(2)* the corresponding results from Hildreth[5] are included in Figures 4.d and 5.d . Quantitative results are given in Table II. The definitions for the error measures used in the tables and the graphic results for experiments *(3)* and *(4)* can be found elsewhere[10].

Experiment 1 (Figure 4): In this case of *pure translation*, the shortest-displacement approximation for an ellipse is symmetric, in contrast to the field of perpendicular displacement components. The smoothed approximation is the correspondence of minimum deformation, no rotation of the boundary is necessary. This is due to the symmetric shape of the regions. Compared to the corresponding result from Hildreth[5] (p.50) our approximation shown in Figure 4.c is inferior. One reason is that in the case of pure translation, the smoothest displacement field obtained by Hildreth is *always* the correct solution. The type of motion is of course not known a priori. The second reason is that the target points on the new boundary are not distributed uniformly. An improved smoothing-step could remedy this problem.

Experiment 2 (Figure 5): In this case of *pure rotation* the initial approximation differs significantly from the actual displacement field. The final *rotated* result, however, represents a good estimate of the real situation. The quality of this approximation can be seen from the low error values for pure rotation in Table II. Extreme smoothing of the vector field around the boundary will *not* render the optimal result and thus the corresponding approximation by Hildreth[5] (p.52) fails to come close to the real situation.

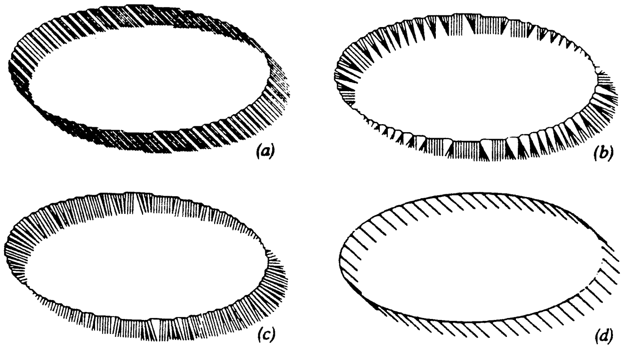


Figure 4. Pure Translation: *(a)* Actual displacement vectors. *(b)* Shortest-distance approximation. *(c)* Final approximation. *(d)* The result obtained by Hildreth [5].

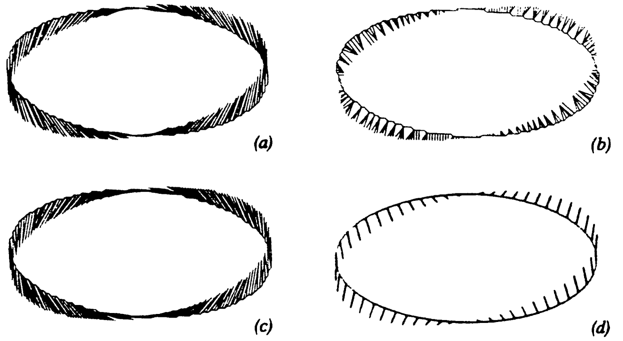


Figure 5. Pure Rotation: *(a)* Actual displacement vectors. *(b)* Shortest-distance approximation. *(c)* Final approximation. *(d)* The result obtained by Hildreth [5].

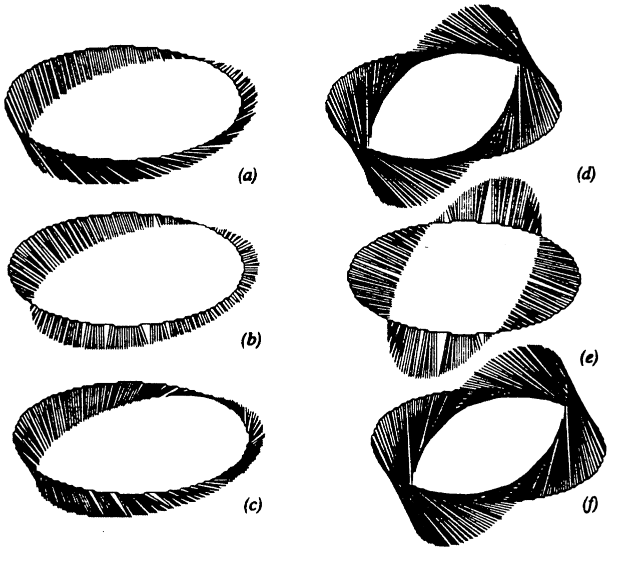


Figure 6. Translation and Rotation *(a-c)* , Extreme Rotation (60°) *(d-f)* : *(a,d)* Actual displacement vectors. *(b,e)* Shortest-distance approximation. *(c,f)* Final approximation.

Experiment 3 (Figure 6.a-c): Here *translation and rotation* were both applied. The initial approximation is already very good at points of *small* displacement, whereas areas of large displacement are not estimated well. This means that some areas along the boundary require rotation, while other areas do not. Since rotation is applied uniformly to the entire boundary, no perfect fit can be expected. Again an improved smoothing process will help to get a better estimate.

Experiment 4 (Figure 6.d-f): This setup was originally chosen to demonstrate the limits of the approach in the presence of extremely wide rotation (60°). Of course in this case it applies even more - the *smoothest* displacement field[5] would not yield a valid approximation. From the error values listed in Table II it can be seen that our technique yields close estimates for the displacement fields over a wide range of rotations.

CONCLUSION

A novel, region-oriented approach for estimating the displacement fields of moving objects has been devised and implemented, performing segmentation and two-dimensional motion analysis simultaneously. Corresponding regions in successive image frames are supposed to overlap, such that connectivity information is carried over from one frame to the next. *Region-growing and wavefront propagation* of displacement data is used to obtain a new segmentation *and* a motion estimate simultaneously. Both processes can be realized in a "relaxation-type" fashion, where each iteration is done in parallel, lending itself to *pipelined* and *VLSI* implementations. From an initial *shortest distance* approximation, the solution is improved by *rotating* the mapping between contours until the correspondence of *least deformation* is found. Experiments conducted on elliptical regions show, that for the case of pure rotation almost perfect estimates are obtained. In the presence of translatory motion, the results depend crucially on how uniformly the points on the new boundary can be distributed. More work must be done to investigate the effects of object-shape, especially non-convex objects, and arbitrary 3-D motion on the quality of the results.

The results can be used in a reconstructive approach to obtain actual 3-D motion parameters. Apart from its application in motion analysis *wavefront region growing* could be useful for image segmentation in its own right.

TABLE 1: Amount of Computation and Deformation Error

Fig.	Trans. in X	Trans. in Y	Rot.	No. of Waves[a]	Boundary Rotation[b]	Deformation Error[c]
4	4.0	0.0	0°	4	0	0.014
	0.0	4.0	0°	4	0	0.010
	8.0	0.0	0°	8	0	0.025
	8.0	-8.0	0°	10	0	0.030
	16.0	0.0	0°	16	0	0.049
5	0.0	0.0	5°	2	2	0.005
	0.0	0.0	10°	4	5	0.006
	0.0	0.0	15°	6	7	0.007
	0.0	0.0	30°	12	16	0.011
6	0.0	0.0	60°	21	40	0.017
6	8.0	-8.0	15°	11	8	0.031

TABLE 2: Errors of the Estimated Displacement Fields

Trans.	Trans.		Orient. Error		Magnit. Error		Total
in X	in Y	Rot.	rms	avg	rms	avg	Error
4.0	0.0	0°	0.111	52.0°	0.661	43.6%	0.896
0.0	4.0	0°	0.646	25.7°	0.296	8.8%	0.526
8.0	0.0	0°	1.143	57.3°	0.651	43.4%	0.850
8.0	-8.0	0°	0.727	37.1°	0.416	17.3%	0.645
16.0	0.0	0°	1.058	53.1°	0.643	41.4%	0.809
0.0	0.0	5°	0.225	7.7°	0.224	5.0%	0.302
0.0	0.0	10°	0.105	4.1°	0.203	4.1%	0.226
0.0	0.0	15°	0.114	4.8°	0.195	3.8%	0.219
0.0	0.0	30°	0.105	4.6°	0.172	3.0%	0.196
0.0	0.0	60°	0.124	5.5°	0.143	2.1%	0.189
8.0	-8.0	15°	0.864	33.0°	0.633	40.0%	1.220

Acknowledgement

This work was supported in part by a grant from Ford Aerospace and Communications Corporation, Newport Beach, California. The authors are grateful to Rick Holben for his continued support.

References

1. G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE PAMI* PAMI-7(4) pp. 384-401 (1985).
2. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley (1974).
3. S. T. Barnard and W. B. Thompson, "Disparity Analysis of Images," *IEEE PAMI* PAMI-2(4) pp. 333-340 (July 1980).
4. R. C. Gonzalez and P. Wintz, *Digital Image Processing,* Addison-Wesley (1977).
5. E. C. Hildreth, *The Measurement of Visual Motion,* MIT Press, Cambridge, Mass. (1984).
6. B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence* 17 pp. 185-203 (August 1981).
7. T. Pavlidis, *Algorithms for Graphics and Image Processing,* Springer-Verlag, Berlin-Heidelberg (1982).
8. J. W. Roach and J. K. Aggarwal, "Determining the Movements of Objects from a Sequence of Images," *IEEE PAMI* PAMI-2(6) pp. 554-562 (1980).
9. S. Ullman, *The Interpretation of Visual Motion,* MIT Press, Cambridge, Mass. (1979).
10. B. Bhanu and W. Burger, "Approximation of Displacement Fields Using Wavefront Region Growing," University of Utah, Dept. of Computer Science UUCS 86-111 (June 1986).

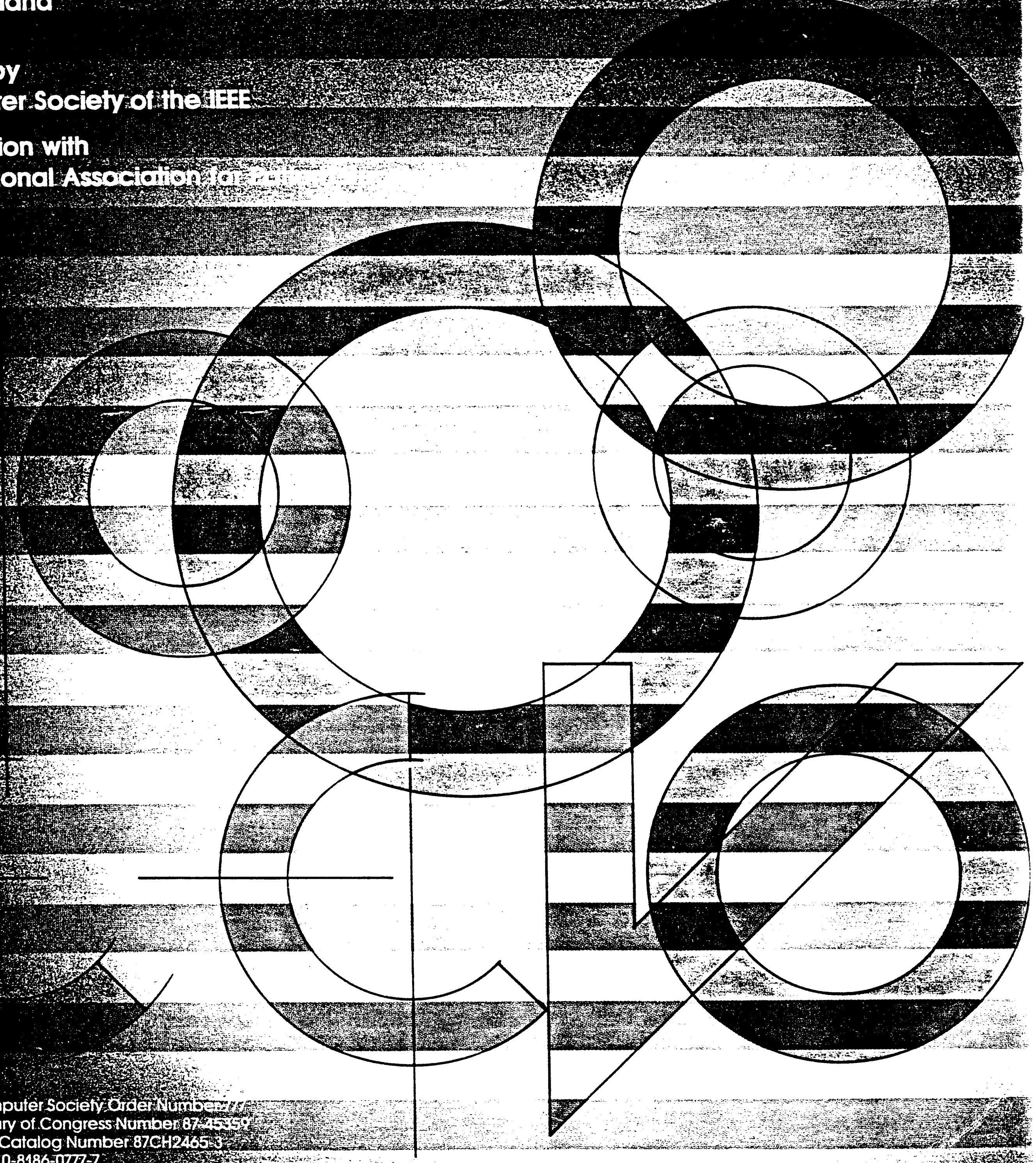
Proceedings

FIRST INTERNATIONAL CONFERENCE ON COMPUTER VISION

June 8-11, 1987
Royal National Hotel
London, England

Sponsored by
The Computer Society of the IEEE

In cooperation with
The International Association for [illegible]



Computer Society Order Number 777
Library of Congress Number 87-45359
IEEE Catalog Number 87CH2465-3
ISBN 0-8186-0777-7
SAN 264-620X

THE COMPUTER SOCIETY
OF THE IEEE

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE
COMPUTER
SOCIETY
PRESS

Bir Bhanu

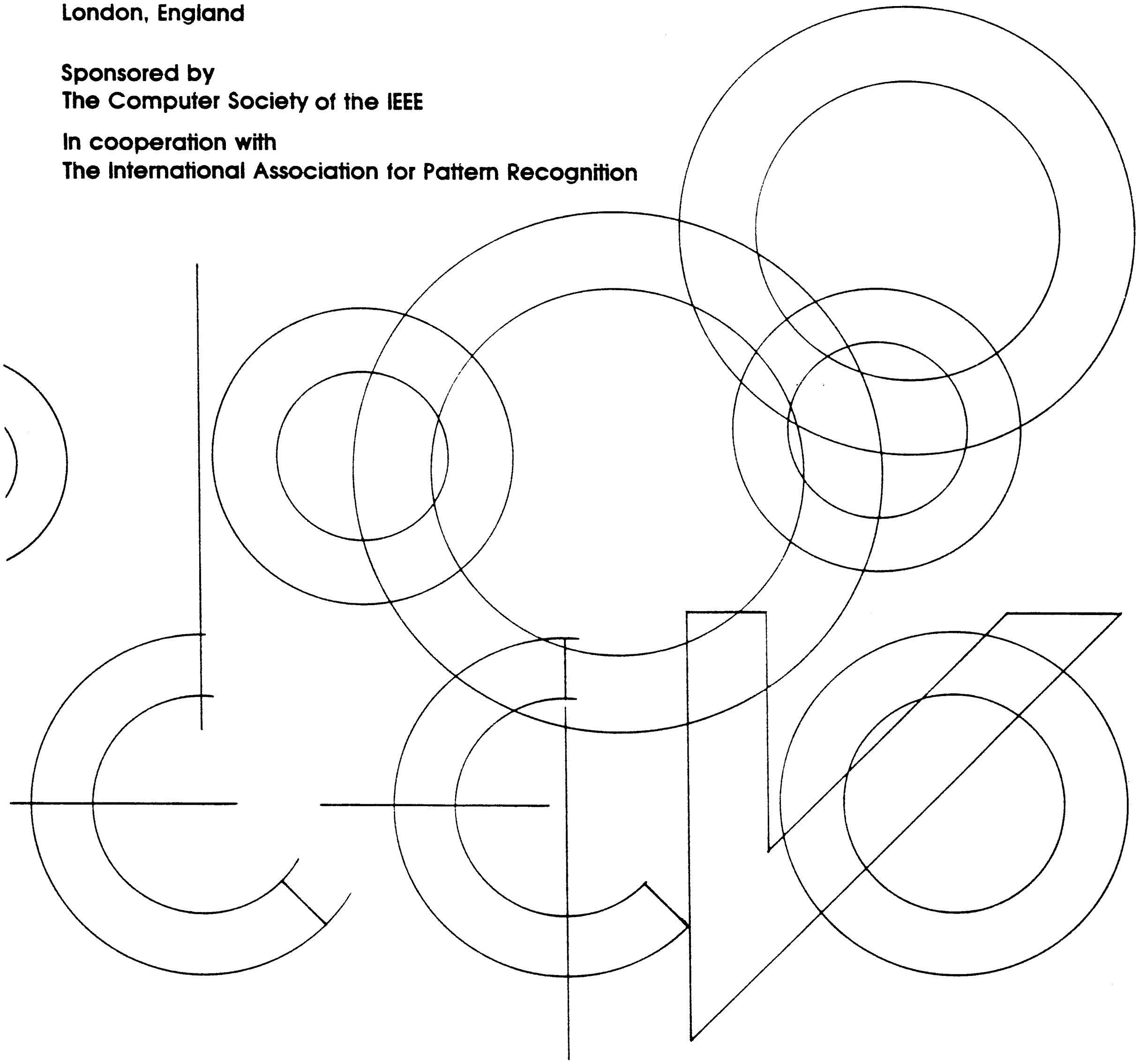
Proceedings

FIRST INTERNATIONAL CONFERENCE ON COMPUTER VISION

June 8-11, 1987
Royal National Hotel
London, England

Sponsored by
The Computer Society of the IEEE

In cooperation with
The International Association for Pattern Recognition



Computer Society Order Number 777
Library of Congress Number 87-45359
IEEE Catalog Number 87CH2465-3
ISBN 0-8186-0777-7
SAN 264-620X

THE COMPUTER SOCIETY
OF THE IEEE

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE
COMPUTER
SOCIETY
PRESS