

A PARALLEL ALGORITHM FOR DETERMINING TWO-DIMENSIONAL OBJECT POSITIONS USING INCOMPLETE INFORMATION ABOUT THEIR BOUNDARIES

AMIR A. AMINI, TERRY E. WEYMOUTH* and DAVID J. ANDERSON

Department of Electrical Engineering and Computer Science, University of Michigan,
Ann Arbor, MI 48109-2212, U.S.A.

(Received 23 October 1987)

Abstract—Extraction of two-dimensional object locations using current techniques is a computationally intensive process. In this paper a parallel algorithm is presented that can specify the location of objects from edge streaks produced by an edge operator. Best-first searches are carried out in a number of non-interacting and localized edge streak spaces. The outcome of each search is a hypothesis. Each edge streak votes for a single hypothesis; it may also take part in the formation of other hypotheses. A poll of the votes determined the stronger hypotheses. The algorithm can be used as a front end to a visual pattern recognition system where features are extracted from the hypothesized object boundary or from the area localized by the hypothesized boundary.

Experimental results from a biomedical domain are presented.

Computer vision	Two-dimensional object recognition	Search	Parallel algorithm
Automated histopathology	Edge-linking	Model-based recognition	Feature-learning

1. INTRODUCTION

The boundaries of objects often provide important features in recognition and classification of visual patterns. Examples include industrial and biomedical applications. In industrial applications, parts are generally assumed to be flat, that is, one of the dimensions is small compared to the other two. The methodology is to match features of the object boundary model against the features of the object boundaries in the image. The closest match in the sense of a metric space is then chosen as having the desired shape (partial occlusion is thus dealt with in some cases). The algorithm developed by Perkins,⁽¹⁾ for example, is based on cross-correlating the tangent angle or the curvature as functions of the curve length between the scene description and the database of models. More recently, the algorithms proposed by Turney⁽²⁾ and Gottschalk⁽³⁾ improve on the time complexity of the previous results. Turney uses salient features of object boundaries for matching. Gottschalk *et al.* reduce the dimension of the feature vectors by projecting them onto a subspace by the Karhunen-Loève transform before the actual matching. For a thorough review of relevant work relating to two-dimensional object recognition the interested reader is referred to Turney⁽²⁾ and Knoll.⁽⁴⁾

In biomedical applications, a system designer has limited control over the environment and thus the problem of extracting object positions and contours still remains to be solved before pattern recognition and classification algorithms may be employed. Examples from biomedical applications include locating tumors in chest radiographs, finding boundaries of white blood cells in images, extraction of left ventricular outlines from serial angiocardiograms, extraction of lung outlines from digitized chest X-rays, recognition of muscle cells and the recognition of inner-ear hair cells in grey-level images. The latter application has been the primary motivation underlying the work reported here. It is expected that the results, however, are of much wider applicability.

Application of various operators to raw image data yields primitive edge elements. However, in non-controlled environments in many instances, disconnected or spurious edge elements result due to noise in the data or inaccuracies of the operators. The resulting edge elements are relatively featureless; hence, additional processing must be done for grouping edge elements (or edge streaks) into structures better suited to the process of interpretation.

Several methods exist for extraction of contours from images. The Hough transform has been used to detect boundaries of shapes which can be specified using a small number of parameters. Kimme *et al.*⁽⁵⁾ have used the Hough technique for locating tumors which have circular shapes in chest radiographs. In the

* To whom correspondence should be addressed.

simplest form, the Hough technique involves a single transformation which is applied to all the edge points, mapping them into a parameter space. Each edge element then votes for a number of possible parameters; the parameters with the most number of votes are chosen as descriptions of object boundaries. The transformation itself is found by inverting the parametric characterization of the curve sought, finding the parameters as a function of the edge point locations.

In the *dynamic programming formulation*,⁽⁶⁾ one must define a figure of merit which embodies the properties of the desired boundary. The procedure then is to optimize this figure of merit in stages, successively. In a variation of the dynamic programming formulation, instead of taking successive points on the grid for optimizing the figure of merit, points far apart are considered.⁽⁷⁾

The goal of contour-tracing algorithms is to find the boundaries of all distinct regions in the image. The strategy is to keep extending the boundary by adding points in the contour direction. A more general and robust method is the heuristic graph searching technique. In this method an image of edge elements is represented as a graph. The problem then becomes one of determining an optimal path in the graph. Martelli⁽⁸⁾ for example, applies the A* algorithm to minimize a cost function determined by the heuristics of the problem at hand. A* algorithm has also been used by Lester⁽⁹⁾ for boundary finding in white blood cell images. In a related representational form, Ashkar and Modestino⁽¹⁰⁾ formulate the contour extraction problem as a minimum-cost tree searching one. Branch costs or metrics are defined and a heuristic tree-searching algorithm (the ZJ stack algorithm) is used for carrying out the search. Ashkar and Modestino apply their algorithm to images of the left ventricular and the lungs.

In the algorithm that we describe, features of the graph search and Hough transform algorithms are combined in a parallel algorithm for producing hypotheses about objects using edge streaks.

2. DESCRIPTION OF THE ALGORITHM

Consider the general problem of contour extraction formulated as graph searching. First, let us define a few terms. A graph consists of a set $\{n_i\}$ of elements called nodes and a set $\{e_{ij}\}$ of directed line segments called arcs. If e_{pq} is an element of the set $\{e_{ij}\}$, then there exists an arc from n_p to n_q , and n_q is called the successor of n_p . Costs can be associated with arcs. The cost of arc e_{ij} is represented by c_{ij} . An arc from n_i to n_j does not imply the existence of an arc from n_j to n_i . If both arcs exist, in general $c_{ij} \neq c_{ji}$. A tree is a directed graph with the following properties. There is one node, the initial or root node which has only outgoing arcs and no incoming arcs, whereas all other nodes have exactly one incoming arc and any number of outgoing arcs.

Given a point on the contour, it is clear that in the general formulation of graphsearching, the contour can be extended in eight possible directions, one for each of the eight nearest neighbors of the point. One can clearly see that optimization of a path metric can be computationally very expensive if the number of nodes in the graph is large. The problem is much more severe if a large number of objects are present in the image as would be in domains where one deals with the counting of objects. An alternate approach is to formulate the search in the space of edge streaks. Instead of considering the points on the sampling grid as the nodes in the graph consider the edge streaks in the plane (strongly eight-connected edge elements) as nodes of the search graph. The problem then becomes one of connecting the edge streaks into object boundaries by a search process. Formulating the problem in this fashion has clear computational advantages over the classical graph search formulation. In addition, in our formulation, no starting point needs to be specified for the search. Dependency of the classical graph search formulations on where the starting point is, can be a significant shortcoming when *a priori* the starting point cannot be specified.

In the algorithm described here, object hypotheses are created by grouping edge segments in a polygonal model with the vertices of polygons being the centroid of each edge segment. The search for polygonal structures is controlled by a set of heuristic rules which picks the best next segment at every step in the search.

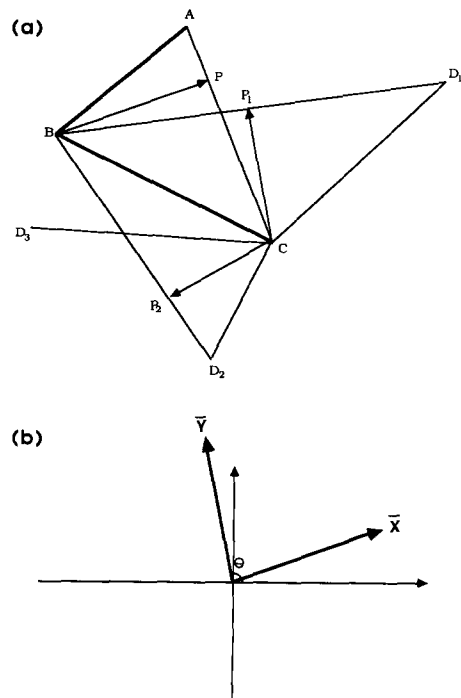


Fig. 1. Criterion for the next plausible segment. Given any two consecutive vectors \vec{x} and \vec{y} (BP and CP₁ in the example), the angle θ between them cannot be greater than $\frac{\pi}{2}$.

In order to make use of possible redundancies and in eliminating the effect of segments produced by noise, a search is started for every edge segment in the image. This is done because there is no *a priori* evidence for knowing whether an edge segment is produced by noise or is caused by presence of an object in the scene. These searches proceed independently and thus can be considered as parallel.

In picking the best next segment at every step in a search, the set of possible candidates are ranked according to a cost function. The cost function captures the problem domain dependent features of the search. Starting from the least costly segment, a test is made to see if the candidate agrees locally with a convex polygon formation process. Figure 1 illustrates how a segment may be chosen from a set of candidate segments. Assume that the point marked with A is the centroid of a search starting segment and that the segments B and C are correctly chosen. Further assume that the set of possible next segment candidates include D_1, D_2 and D_3 . Clearly the segment that should be chosen is D_1 . The way that the algorithm reaches this conclusion follows. In the initial step, the three segments are ranked according to the cost function and the best segment is chosen. Following this step, a test is made to see if the centroids of the segments agree

with a polygon formation process. Consider the vectors $\mathbf{BP}, \mathbf{CP}_1$ and \mathbf{CP}_2 which are drawn in such a way that they are perpendicular to lines $\mathbf{AC}, \mathbf{BD}_1$ and \mathbf{BD}_2 respectively. In order for a convex polygon to form an object contour, given any two such consecutive vectors \mathbf{X} and \mathbf{Y} , the angle θ between them cannot be greater than $\frac{\pi}{2}$ (Fig. 1). In fact, it is easy to prove that a polygon is convex if and only if this is true for all such consecutive vectors.

If one of D_2 or D_3 is incorrectly chosen, the algorithm backtracks and chooses the best next segment. In this fashion the algorithm is capable of exhausting the whole search space.

A method has also been devised which has a more global nature. The idea is to only consider the segments that are consistent with the current search direction (clockwise or counter clockwise).

The criterion for stopping a search started from a particular segment is that the algorithm reach the same segment that it started the search from. The algorithm will backtrack however if the sum of the external angles formed exceed or equal 2π without the starting segment having been reached (Fig. 2).

Figure 3 illustrates the nature of the searches and the voting mechanism. The outcome of each search is an object contour hypothesis; a set of grouped edge streaks satisfying both the local and the global relational constraints described before. In addition, if one starts from the starting segment of an object contour hypothesis and moves in the direction of the search along the contour, he will eventually reach the search starting segment. After the search in all the search spaces comes to a halt a poll of the "voted" object contour hypotheses is taken and "verified" object contour hypotheses are associated with the hypotheses that have the most number of votes and adhere to global figures of merit.

In the Hough transform method each edge element votes for a number of parameter values (in the case of the generalized Hough transform, parameter values are the shape's reference point coordinates). The parameter values with the most number of votes are chosen as they give the best description of the curve in terms of the class of shapes described by the parametric

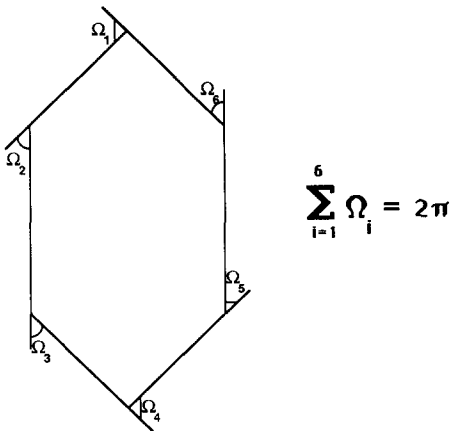


Fig. 2. Sum of the external angles of a convex polygon is 2π . The simple geometric fact is used in the search (refer to text).

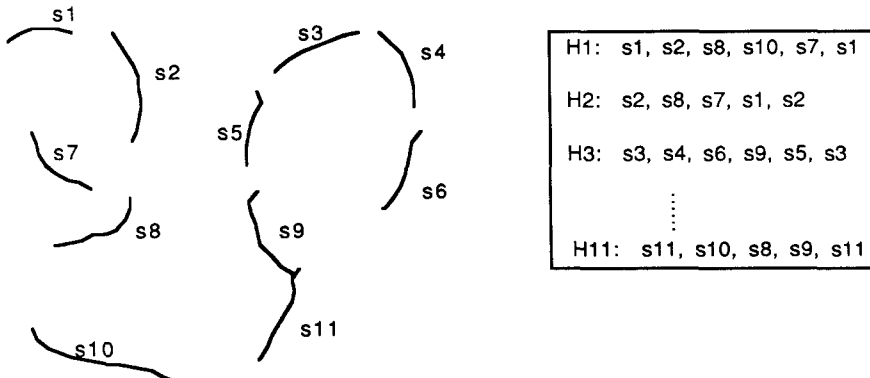


Fig. 3. For every edge segment in the image, a separate, independent search for an object contour is started.

equation (or the so-called R-table in the case of the generalized Hough transform). The algorithm presented here does not need to have a parametric model of shapes as do algorithms such as the Hough transform.

If there are k segments in a given search space with at most $m \leq k$ segments forming a closed contour, then a lower bound to the computational speed results when at each search level there is only a need to sort offsprings of one node, a total of $\sum_{i=0}^m (k-i)$, hence a computational speed of $O(k \log k)$ [assuming a sorting algorithm of $O(k \log k)$ time]. The worst case results when, due to backtracking, every nodes' offspring need be sorted, a total of $\sum_{j=0}^{k-1} \left(\prod_{i=0}^j (k-i) \right)$. This results in a computational speed of $O(k!)$. One should bear in mind, however, that given an effective cost function the probability of occurrence of a worst case is rather small. In the algorithm described here, use is made of edge streaks rather than edge points, this amounts to more efficient computations. The computational speed of the Hough transform is $O(ed^n)$ where e is the number of edge elements, d is the number of distinct parameters, and n is the number of possible values for the parameters.⁽¹¹⁾ It is expected that when there are a large number of edge elements and parameter values the Hough algorithm can be computationally intensive. In addition, the Hough algorithm is sensitive to variations in the instances of a modelled shape. Thus when there are a large number of similar but different shapes present in an image the performance of the Hough technique deteriorates.

The algorithm proposed here can be used as a first step in a larger shape matching system where in the first step locations are specified. Boundary matching can then be carried out on descriptions with the most number of votes.

3. EXPERIMENTAL RESULTS

Figure 4 shows an image of the inner-ear hair cells. Hair cells are responsible for transforming the mechanical energy of acoustic vibrations into electrochemical energy suitable for propagation on the auditory nerve. The objective here is to produce a count of the normal hair cells in assessing the damage done to the hearing of laboratory animals.

Application of the Canny edge operator⁽¹²⁾ does not yield perfect cell contours. As is evident from Fig. 5 edge segments often belong to more than a single cell and those that do belong to a single cell contour are normally disconnected.

In dealing with the first problem curves are partitioned at the zero-crossings of their k - curvature function.^(13,14) Figure 5 illustrates the zero-crossings of the edge segments in this problem domain (these are points that the gray value changes.) Following partitioning, edge segments are grouped into cell contour hypotheses and are tested for fitness.

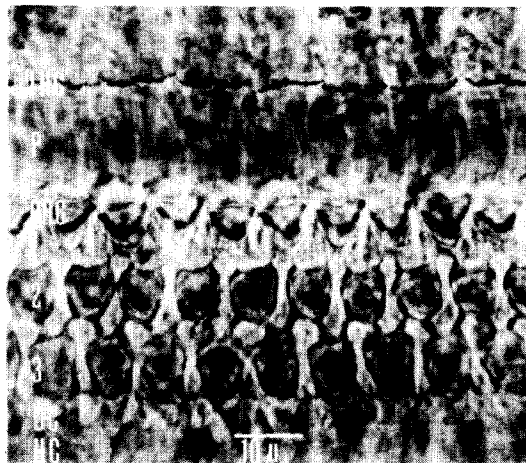


Fig. 4. Image of a cross section of the hair cells. Notice that there is one row of inner hair cells and three rows of outer hair cells.

As in any searching scheme the question of plausible strategies for limiting the search space arises; i.e. a way to reduce the branching in the search tree. As a reasonable strategy for example, it should be possible to search in the space of edge segments that are spatially proximate to a search starting segment. The only way to take advantage of such an idea is to rely either on *a priori* information about object sizes provided as input or have some way of measuring it based on problem domain constraints. The constraint that we use is the size of the spacing between the inner hair cells and the first row of outer hair cells. A cell area measure may be directly based on this. In order to find a quantitative measure of this spacing, an edge frequency histogram as a function of rows in the image is made. Since the number of edge pixels in this spacing is substantially less than the number of edge pixels in the areas where cell contours exist, we need only detect the start and end of the deepest valley in this histogram. It is worth noting that it may also be possible to locate different cell rows by detecting other valleys in the histogram (that is, the top and bottom of cells).

3.1. The cost function

In ranking the edge fragments we use: length, total curvature and distance. The intuitive idea here is that segments that are longer and closer to the current segment should be less costly. In addition, knowledge about concavity of the current segment and its position forces a constraint on the possible shape of the next segment.

The cost between fragment i , and fragment j , a possible successor is computed to be:

$$c_{ij} = -\alpha_1 L_j + \alpha_2 \gamma^{\theta_{ij}, P} K_T^j + \alpha_3 d_{ij}^2.$$

L_j is the length of the j th segment, θ_{ij} is the search orientation computed by finding the orientation of the line connecting the centroids of the i th and j th

fragments, P is the position which can be one of top or bottom. This may be found offline by determining whether the centroid of the segment is near the top or bottom of the cell. Computation of the γ parameter is as follows:

- (1) if $P = \text{top}$ and $-\pi/2 \leq \theta_{ij} \leq \pi/2$ then $\gamma = 1$;
- (2) if $P = \text{top}$ and $\pi/2 \leq \theta_{ij} \leq \pi$ or $-\pi \leq \theta_{ij} \leq -\pi/2$ then $\gamma = -1$;
- (3) if $P = \text{bottom}$ and $-\pi \leq \theta_{ij} \leq 0$ then $\gamma = -1$;
- (4) if $P = \text{bottom}$ and $0 \leq \theta_{ij} \leq \pi$ then $\gamma = 1$.

Choosing γ in this way reduces the cost for segments whose curvature has the correct sign.

$K_T^j = \sum_{m=1}^{L_j} k_m^j$ with k_m^j being the curvature of the j th segment at the m th point and d_{ij} is the smallest Euclidean distance between start or end of i th segment and start or end of j th segment.

The value of the coefficients $\alpha_1 > 0$, $\alpha_2 > 0$ and $\alpha_3 > 0$ are empirically set. During the search, the user is allowed to update these coefficients after each segment is chosen in order to reflect the importance of each feature in the search. In the future we intend to have a "feature-learning mechanism" which would compare the features of the wrongly chosen segment which lead to backtracking and the segment which should have been chosen in the first place. The coefficients can then be updated based on this measured difference. Such updating is expected to decrease the search time by reducing the need for backtracking.

Figure 6 shows some of the typical cell hypotheses generated by the algorithm when the algorithm is run on the picture in Fig. 5. In each case the smaller square

corresponds to the centroid of the search starting segment, the bigger square corresponds to the centroid of the next segment found. In Fig. 7 we illustrate the best first search strategy used in hypothesizing the object contour corresponding to "bucket 0" of Fig. 6. Part (a) of the figure illustrates the first step in the search where segments within a specified radius are marked with their associated costs. In part (b) the result of the best-first search strategy is seen where the segment marked with a cost of 1.000 in (a) is picked. The cost for all other segments are now measured relative to this newly found segment. Searches for hypotheses continue recursively in this fashion.

In one experiment, 224 search starting segments were present in the image. From this total, 21 correct cell contour hypotheses were associated with cell 1 and 31 with cell 2. Eighteen incorrect hypotheses were generated for cell 1 and 25 incorrect hypotheses were generated for cell 2. The remainder of the hypotheses could not be associated with either cell. Although meaningful contours were formed by linking some segments together, the number of votes that such objects received were not significant.

4. CONCLUSIONS

The algorithm presented here can be employed for locating objects in the plane. Once this is done more elaborate analysis of the region enclosed by the grouped edge fragments can be done for recognition or classification of the region. Shape matching and other higher level image interpretation algorithms may be



Fig. 5. Application of the Canny edge detector followed by curve partitioning.

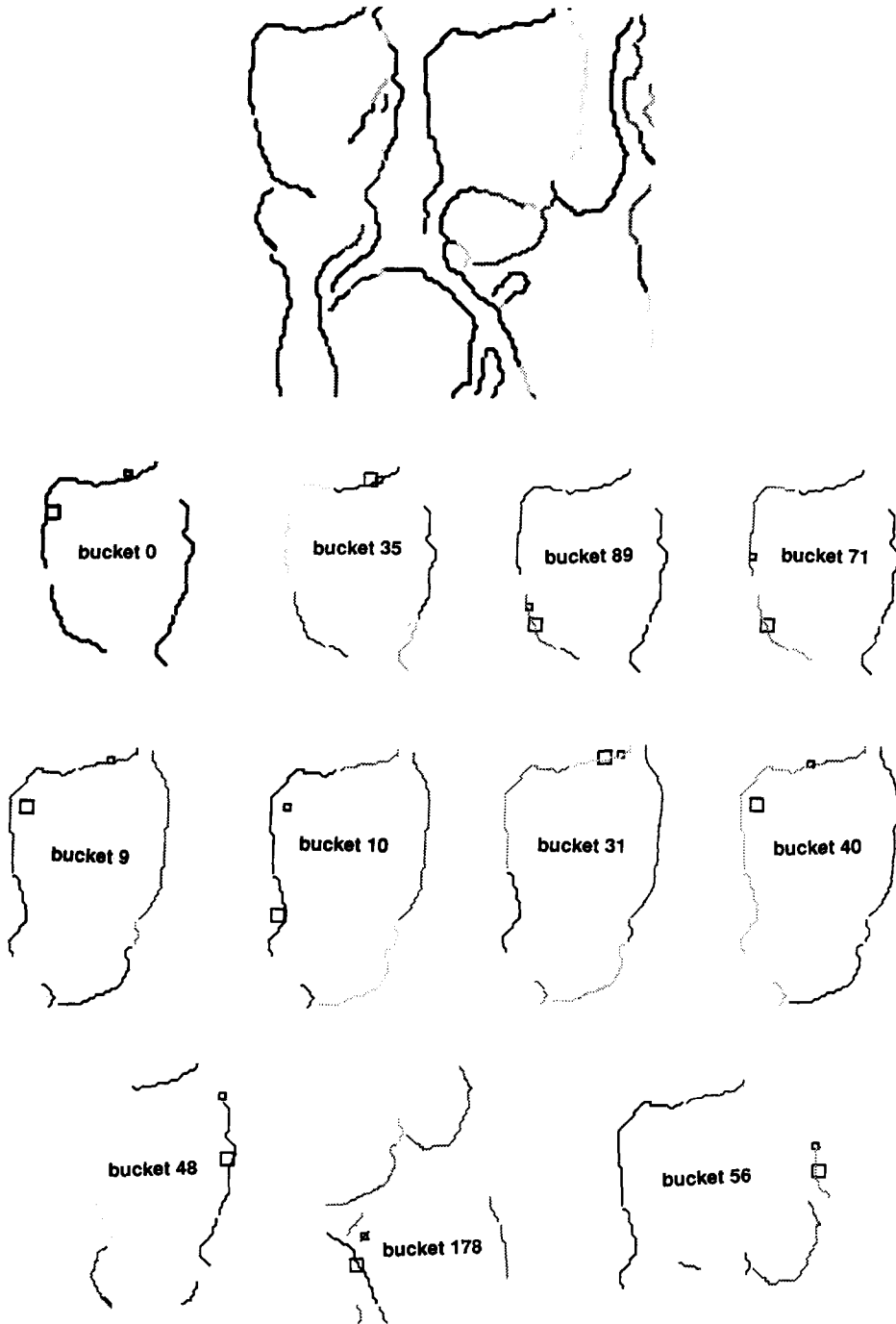


Fig. 6. A few of the generated cell hypotheses. Smaller square corresponds to the centroid of the starting segment. The bigger square is the second segment.

used for further abstracting the image data into symbolic descriptions using the connected edge fragments.⁽¹⁵⁾

The algorithm has interesting features.

(1) As long as we know that there is any number of closed, well-behaved contours (any kind of closed object which might be non-analytic) in the image

which are non-overlapping, the algorithm can produce their positions in the image.

(2) No starting point needs to be specified for the algorithm since there is a search for every segment.

(3) All searches can proceed in parallel (Fig. 3).

(4) Each edge segment votes for a single object contour hypothesis (and not a number of hypotheses)

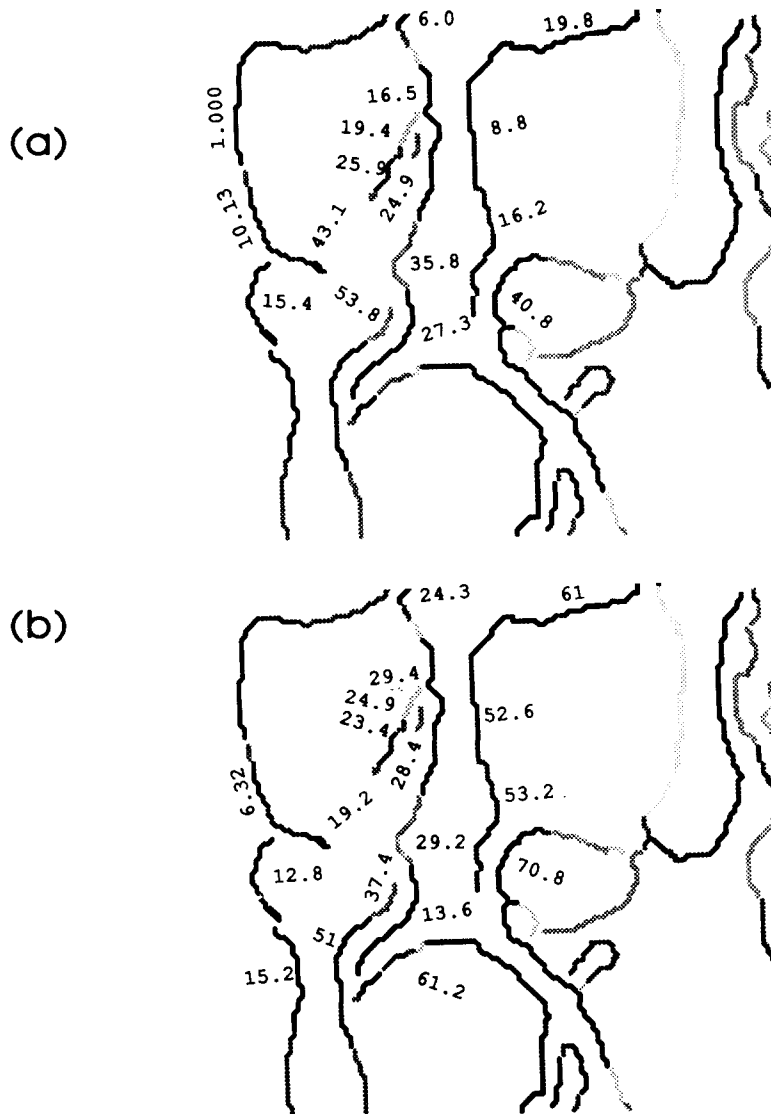


Fig. 7. The calculated costs for segments within a specified radius of the starting segment in the first two steps of the search for the 0th hypothesis. (a) The search starting segment is marked with a small square; (b) the segment marked with the bigger square is the best segment that was found in (a).

but can contribute to any number of other hypotheses (Fig. 3).

(5) Once votes are gathered, global figures of merit may be invoked in order to remove "bad votes."

Experimental results in the domain of counting inner-ear hair cells are promising.

SUMMARY

Due to noise and variations in illumination, edge detection algorithms applied to images of real objects often produce incomplete object boundaries. We present a parallel algorithm that can specify the location of objects in the image plane based on edge streaks produced by an edge operator. Contour hypotheses are created by grouping edge segments in a polygonal model with the vertices of polygons being

the centroids of edge segments. For every edge segment in the image a local search is carried out to find the most plausible boundary. The edge grouping process is a best-first search for convex, polygonal structures. Every edge segment in the image thus "votes" for a contour hypothesis; it may also be part of another contour hypothesis started from a different segment. In determining the correct hypotheses global figures of merit are applied to contours with the most votes, producing the final output of the algorithm.

Features of the algorithm are compared and contrasted mainly to graph search formulations and the Hough transform; both methods having been proposed in the literature for extraction of image features. In addition, an analysis of the time complexity of the algorithm is made and is compared to the Hough transform.

Experimental results are presented from a biomedical domain where the algorithm is applied to edge images as part of a two step process. In dealing with the problem of connected edge segments belonging to more than a single cell, edge segments are partitioned at the zero-crossings of their curvature function. In the second step the algorithm is applied to the resulting edge data in order to produce cell contour hypotheses. A cost function based on shape, distance and length of each edge segment is described which is utilized in picking the best next segment at each step of the search. Finally, conclusions are made about the efficacy of the approach.

REFERENCES

1. W. Perkins, Simplified model-based part locator, *Proceedings of the 5th International Conference on Pattern Recognition*, pp. 260–263 (1980).
2. J. Turney, T. Mudge and R. Volz, Recognizing partially occluded parts, *IEEE Trans. Pattern Anal. Mach. Intell.* **7**, 410–421 (1985).
3. P. Gottshalk, J. Turney and T. Mudge, Two-dimensional partially visible object recognition using efficient multidimensional range queries, *Proceedings of the International Conference on Robotics*, 1582–1589 (1987).
4. T. Knoll and R. Jain, Recognizing partially visible objects using feature indexed hypotheses, *IEEE. Robotics Autom. RA-2*, 3–13 (1986).
5. C. Kimme, D. Ballard and J. Sklansky, Finding circles by an array of accumulators, *Commun. ACM* **18**, 120–122 (1975).
6. U. Montanari, On the optimal detection of curves in noisy pictures, *Commun. ACM* **14**, 335–345 (1971).
7. M. Fischler and R. Elschlager, The representation and matching of pictorial patterns, *IEEE Trans. Comput.* **22**, 67–92 (1973).
8. A. Martelli, An application of heuristic search methods to edge and contour detection, *Commun. ACM* **19**, 73–83 (1976).
9. J. Lester *et al.*, Two graph searching techniques for boundary finding in white blood cell images, *Comput. Biol. Med.* **8**, 293–308 (1978).
10. G. Ashkar and J. Modestino, The contour extraction problem with biomedical applications, *Comput. Graphics Image Process.* **7**, 331–355 (1978).
11. D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition* **13**, 111–122 (1981).
12. J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 679–698 (1986).
13. A. Rosenfeld and E. Johnston, Angle detection on digital curves, *IEEE Trans. Comput.* **C-22**, 875–878 (1973).
14. B. Shahraray, Measurement of boundary curvature of quantized shapes: a method based on smoothing spline approximation, Ph.D. Thesis, The University of Michigan, Ann Arbor, MI (1985).
15. A. Amini and T. Weymouth, Representation and organization of domain knowledge in a blackboard architecture: a case study from computer vision, *Proceedings of the Conference on System, Man, and Cybernetics*, Fairfax, VA, pp. 884–889 (1987).

About the Author—AMIR A. AMINI was born in Tehran, Iran, in 1965. He received his B.Sc. in Electrical Engineering with honors from the University of Massachusetts at Amherst in 1983, and the M.S.E. in Electrical Engineering from the University of Michigan, Ann Arbor in 1984. He is currently a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. His research interests are in computer vision and artificial intelligence. Mr. Amini is a member of IEEE, Eta Kappa Nu (Electrical Engineering Honors Society) and Phi Kappa Phi.

About the Author—TERRY E. WEYMOUTH received the B.Sc. and the M.Sc. degrees from the University of Nebraska in 1973 and 1976, respectively. He was awarded the Ph.D. degree in Computer Science in 1985 for his work on VISIONS, a system for interpreting natural scenes, developed at the University of Massachusetts at Amherst. From 1976 to 1979 he was employed by the Bell Laboratories, Naperville, IL. Since fall of 1985 he has been on the faculty of Electrical Engineering and Computer Science at the University of Michigan where he does research on computer vision and artificial intelligence, and teaches courses in computer vision and graphics. Dr Weymouth is a member of IEEE, American Association for Artificial Intelligence, Association for Computing Machinery, and Upsilon Pi Epsilon (Computer Science Honors Society).

About the Author—DAVID J. ANDERSON received the B.S.E.E. degree from Rensselaer Polytechnic Institute, Troy, NY, and the M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Wisconsin, Madison.

After completing a postdoctoral traineeship with the laboratory of Neurophysiology, University of Wisconsin Medical School, he joined the University of Michigan, Ann Arbor, where he is now a Professor of Electrical Engineering and Computer Science and Otolaryngology. His duties at the University of Michigan include teaching and research in digital signal processing and bioengineering.