

DETERMINING OBJECT SHAPE FROM LOCAL VELOCITY MEASUREMENTS

M. LINDENBAUM and A. M. BRUCKSTEIN

Department of Electrical Engineering, Technion, IIT, Haifa, Israel

(Received 17 February 1987; in revised form 11 November 1987)

Abstract—In this paper we discuss the following question: what can be said on the movement and shape of a planar rigid object from sparse samplings of edge crossing measurements. It is shown that if the object moves in translation over an array of detectors, providing local edge crossing information, one can determine the translation velocity and use a backprojection process to obtain an equivalent sampling of the object along a set of parallel lines. Furthermore, if information from several such passes (with different translation velocities) over detector arrays is available one can combine the corresponding backprojected data-sets to determine classes of objects that are consistent with all the data. Such an array of detectors is shown to be a useful alternative way of acquiring information on dynamic scenes when the tasks are limited to positioning, velocity measurement and shape recognition of rigid two-dimensional objects moving on a contrasting background.

1. INTRODUCTION

The task of detecting moving objects in a dynamic scene and extracting information about their movement and shape can be performed by sampling the time varying image at short intervals and analysing sequences of static images.⁽¹⁾ This approach gives the possibility to develop sophisticated algorithms to evaluate motion in space, to determine object shapes and to deal with scenes involving multiple moving objects.⁽²⁾ However, if high resolution images are to be processed, this method is prohibitively expensive computationally. Furthermore, this approach is based on the notion that the basic element in the interpretation of a dynamic scene are static images. Motion analysis therefore relies here on detecting differences or similarities between successive static images, assuming these were already analysed. This "static is basic" approach is most probably not the one implemented in biological systems as can be learned, for example, from the human visual system, which can sometimes distinguish an object from its background by motion alone, and may even sense movement without any ability to point out where, precisely in the field of sight it happened.^(2,3)

A popular alternative approach postulates that the input to a system that analyses dynamic imagery is the so-called optical flow.^(1,8) This approach, which assumes that there exists a wired in processor capable of providing the optical flow directly, is yet equally expensive to implement since no efficient way to get the optical flow has been discovered.

However, movement and shape analysis could also be based on the analysis of results of some local

spatio-temporal processing. Such local processing may yield any function of the space or time derivative of gray levels in some neighborhood, as is for example the locally measured velocity field. Since only neighborhood and parallel preprocessing is involved, this approach is biologically feasible and may be computationally efficient. A considerable simplification is possible when making some assumptions about objects in the image and their motion. For example, Ref. (9) describes a method which assumes a moving rigid body, looks for time varying corners in the image (where velocity can be uniquely detected) and proceeds using this information as a constraint for determining velocities along adjacent edges. If some restricted tasks are to be performed, it may be possible to extract enough information on motion from local spatio-temporal processing done only at a sparse set of points. It may even be possible to extract shape characteristics of the moving object from this data.

One biological example is the visual system of insects in which, at higher levels of processing, there exists a group of relatively few neurons responding proportionally to movement in specific directions occurring in specific areas of the visual field.⁽⁴⁾ Other examples are simple tactile and visual systems used to sense objects on a conveyor belt. By placing above the belt an array of sensors that can merely detect the presence of an object passing under them, it is sometimes possible to identify the shape and orientation of objects on it. (Here the type of motion is of course known in advance, as is the class of object shapes expected to appear.)

In this paper we discuss a method for extracting motion and structure information from the outputs of

sparse arrays performing local spatio-temporal processing on a dynamic imagery. As pointed out earlier, physiologists have determined that the fly's visual system has giant neurons whose response measures motion in specific areas and directions. However, very little is known about the higher level processing of this information. The system we deal with is based on a simplified model of the measurements carried by the giant neurons, where the simplification is the assumption that the measurement is done in very small regions of the input image. Furthermore, we also assume that these measurements are all the information available for higher level processing. As this method derives information about the shape of an object it can be included in the so called "shape from motion" class of problems. However it is very different from the classical approaches as it postulates for input neither a series of static images nor a dense optical flow field, but a very sparse spatial sampling of the locally measured velocity field. We shall see that, under the assumption that the dynamic scene is a rigid two-dimensional shape moving in translation, these relatively few measurements are enough to estimate position and velocity, to recognize a shape from a predetermined set of shapes and also to find an approximation of an arbitrary object.

This paper is organized as follows. First the problem formulation is given, followed by the description of a method to determine motion parameters and gather some basic information about the shape of the object. Then we present a way to improve the quality of shape evaluation by integrating information from several sets of independent measurements. We conclude with a discussion which shows the close relation between the proposed scheme and a known method for shape recognition and also point out directions to relax some of the limitations imposed on the image and measurement model.

2. THE PROBLEM AND SOME PRELIMINARY DISCUSSIONS

Let us assume that a binary dynamic image contains a rigid "black" object moving in translation at constant speed and direction upon a "white" background. Suppose we place local velocity detectors at M fixed places in the image field, each of them characterized by the following operation: at the moment an edge crosses the detector, the detector measures the slope of the edge and the component of the velocity normal to it (Fig. 1). Therefore passage of the object boundary on a certain detector yields a quadruple (i, T, ϕ, V_n) , where we denoted by:

- i the label of the particular detector;
- T the time the passage of the edge happened;
- ϕ the slope of the edge at the time and place of passage;
- V_n the velocity component normal to the edge;

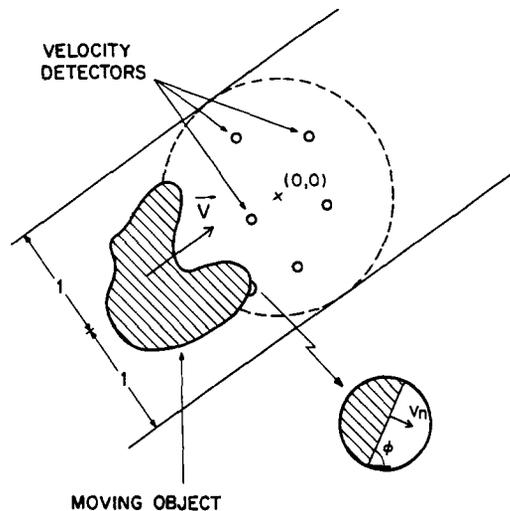


Fig. 1. An object moving in translation over a detector array.

and of course $1 \leq i \leq M$, $0 \leq \phi < 2\pi$, $V_n > 0$. (A brief discussion of the properties and limitations of motion measurement with local spatio-temporal processing of dynamic images is found in the Appendix).

Suppose that after the object has swept through the image all the information we have is the even number of quadruples (since each white to black change implies a corresponding black to white one except for singularities), as described above, together with the positions of the detectors. Given this information, we ask what can be inferred about the shape of the object and its velocity.

Another related and more interesting question that arises is then the following one: suppose the same object (in the same orientation) moves in translation, with different speed and direction and passes elsewhere on another set of detectors (or alternatively it moves over the same detector array again). Given the above described edge crossing information for several passages, generated with different velocity directions, what can now be inferred on the shape of the object? We shall therefore ask whether and how such further information reduces the class of objects that could have produced all the measurements. (In the discussion we provide suggestions on how to relax the limitations on the object and its motion model and we also show the usefulness of such detector array for shape recognition tasks.)

Let us first discuss certain limitations on the size and position of the moving object, under which we shall consider the above presented shape evaluation problems. The input is assumed to be a continuous binary image with theoretically infinite spatial resolution. (Therefore, no discretization will be implied in our solution too; however, the particular implementation of the shape reconstruction procedure will of course be limited in resolution due to the finite word length of the computer.) We clearly have to insure that the whole object passes within a certain monitored

frame for some time, otherwise only partial information becomes available. For normalization purposes, and with no loss of generality, we shall assume that every point of the object passes at some time through a circle of radius 1 and a fixed center-point designated as (0,0) in the image field. This means that, the entire object is at all times, included in an infinite strip extending in a direction parallel to the motion velocity, whose axis passes through the point (0,0) and whose width is 2 (Fig. 1). By assumption, the M motion detectors are placed within this "monitored" frame and as the object sweeps through, we get the quadruples, (i, T, ϕ, V_n) as defined above. The number of quadruples, even with a small M , is generally unlimited. Some artificial objects might even give rise to an infinity of edge crossings. However, for most "nonpathological" objects, given the number of detectors M , the number of data quadruples will be a small multiple of it. In convex objects, for example, the maximum number is $2M$ but for complex shape like the one in Fig. 2 the number of data quadruples with an array of three detectors, is 16.

Finally, in all our discussion on shape evaluation from such partial information, we shall never question the consistency of the data, i.e. it will always be assumed that a particular connected object did indeed elicit the detector readings provided to us.

3. VELOCITY AND SHAPE EVALUATION PROCEDURES

3.1. Determination of velocity

The determination of the true object velocity (magnitude and direction) is straightforward. Each data quadruple (i, T, ϕ, V_n) constrains the velocity to a line in the (V_x, V_y) velocity plane. By taking two such nonparallel lines in the velocity plane and finding their intersection we have the velocity of the rigid object (Fig. 3). The minimum number of detectors required to uniquely fix the velocity can be as small as one (which of course has to meet at least two edges with different slopes), but it is also possible that the velocity cannot be determined from any number of detectors. An example could be the case of a parallelogram-shaped object moving in a direction parallel to one of its sides. This later case is obviously pathological and we will assume that it happens very rarely. The above discussion also assumes that no noise or quantization affect the measurements. If this is the case, the velocity can be determined from any two "non parallel" measurements quadruples, i.e. having different ϕ 's, the information from the other quadruples being redun-

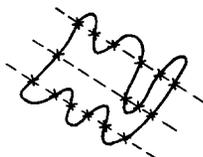


Fig. 2. Example of boundary intersection for a complicated shape.

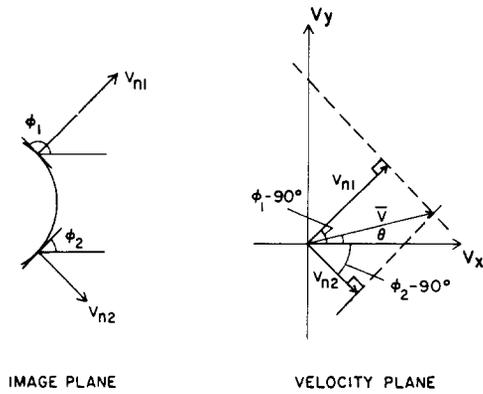


Fig. 3. Velocity determination from two edge crossing data quadruples.

dant. If noise or quantization do affect the data we can obtain a velocity estimate by using the relation:

$$V_n = V \cdot \cos\left(\phi - \frac{\pi}{2} - \theta\right)$$

which has to hold for all data points, (V and ϕ being the polar coordinate of the object velocity, as in Fig. 3). Fitting the best cosine curve to all the available (V_n, ϕ) pairs will yield the estimate of V and θ .

3.2. Shape information from one pass

Once the object velocity is determined we may proceed to extract information on the shape of the moving object. The following argument can be made: the meaning of a measurement (i, T, ϕ, V_n) is that a point on the edge of the object was at the specific time T in the place of detector i , (which is the detector that "created" the quadruple). Therefore at time $t = 0$ this point was in the position $\bar{r}_i - \bar{V}T$, where the vector \bar{r}_i denoted the position of detector i .

This "backprojection" applied to all data quadruples will yield a set of points known to be on the boundary of the object at $t = 0$. Thus for the backprojected points we know the precise relative position and the corresponding slope (Fig. 4). We further know that some segments between edge points "created" by the same detector consist only of interior points of the shape. Indeed, if we order the quadruples generated by a certain detector with increasing T , the interior segments are between the first and second, third and fourth, etc. backprojected points. (These interior segments will be called "black segments".) The other segments on the same line, including two half infinite segments (or one infinite segment in the case there was no interception of the detectors) are known to belong to the exterior of the shape (and will be denoted as "white segments"), as shown in Fig. 4. Apart from these points and segments our basic assumptions impose further restrictions that require the connected shape to be completely included in a strip which is in the direction of motion and has a width of 2 (Fig. 1). Detectors that create no quadruple will of course limit the range even more. For example,

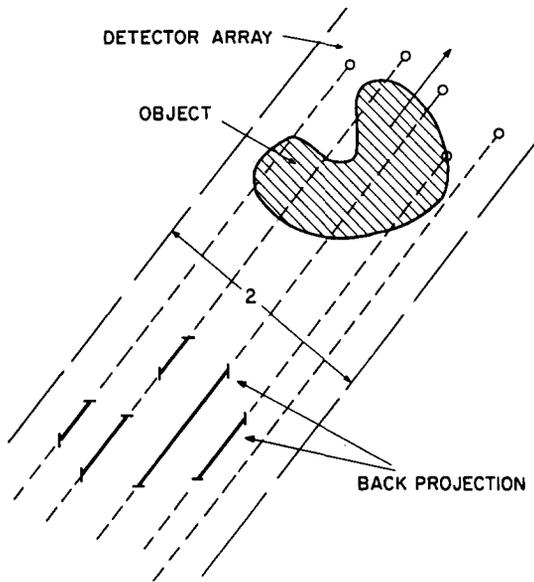


Fig. 4. Back-projected edge points.

let us assume that detector i was not intercepted, then line $\bar{r}_i + \bar{V}t$ (t - parameter) is an infinite "white" segment, whose points belong to the exterior of the object and the connected object will lie on one of its sides.

In case the object is known to be convex, or, more generally, if no detector yields more than two quadruples, we can proceed by trying to approximate the shape invoking additional assumptions. For example, we can model the object to be a polygon, or try to approximate its boundary with polynomial curves (using the measured slopes at the edge points). These approximations are by no means guaranteed to give a reconstructed object which resembles the original one. For example, Fig. 5a shows two different shapes which both satisfy the same measurements. If, however, the above condition is not satisfied, we cannot do even this much, since we do not know how to determine the ordering of the points along the object boundary. We have in this case several possible orderings corresponding to radically different solutions (as for example in Fig. 5b) and we sometimes may be able to pick one of them based on some prior information about the shape.

Let us now address two particular cases. The first one is the case in which the object is not intercepted by any of the detectors, but is known to have passed through the image field. In this case the width (the maximal dimension of the object in the direction perpendicular to the motion direction) is limited to be smaller than $L = \max_{i,j,\phi} \{d(i, \phi) - d(j, \phi)\}$ where $d(i, \phi)$ is defined as the projection of the placement of detector i on an axis whose slope is ϕ and i, j are neighbors in this projection.

The second case is when the object is *a priori* known to have a convex shape. Then, not only do we know points on its edge, and their order, but also the

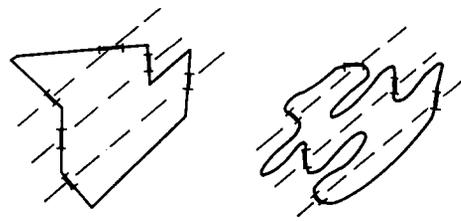


Fig. 5a. Two different shapes satisfying the same measurements and the same ordering of edge points.

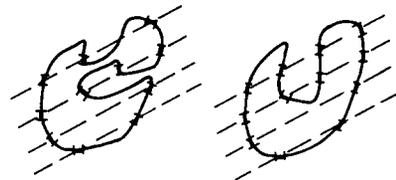


Fig. 5b. Two different shapes satisfying the same measurements with different ordering of edge points.

tangents in these points create a convex polygon in which the object has to be included. Furthermore, this bounding polygon can be created even without determining the backprojected points on the object boundary! This is so since the data in each quadruple, although it does not independently define the tangency point on the boundary, is enough to define the position of the tangent line at $t = 0$, (ϕ defines its slope and the component of the object velocity, perpendicular to the tangent, can be used, together with T to back project the entire line, see Fig. 6).

3.3. Integrating information from two passes

Consider the situation when an object moves over a large detector system composed of several arrays (which may have some degree of overlapping in space), in nonconstant translation, or that the object passes the same array several times and in different directions. The situation is also similar to the one in which a static object is actively scanned with a detector array in

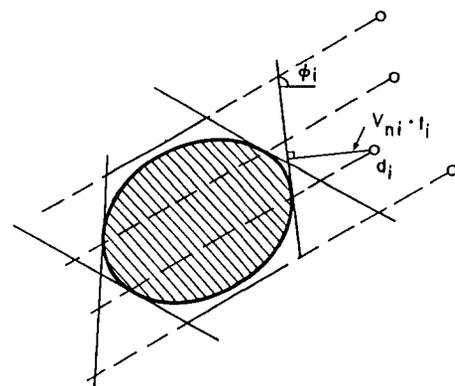


Fig. 6. Generation of convex enclosing polygon.

various directions. Assume that the time of every passage over each array is short enough so that the velocity can be considered to be constant for all the edge crossing times by detectors in the array. The passes may be in different directions and velocities but recall that the object is assumed to be in the same orientation in all of them. The question that arises is how to integrate different sets of edge data (which follow from different passes), to find an approximation of the object shape that is better than the one obtained from a single pass. We start by considering the case of two sets of edge points and the results will serve as the basis for the treatment of the general situation when data from many passes is available.

In fact, what we have to do is to determine the relative positions of the sets of backprojected edge points generated by the two independent passes so that all of them will belong to the true object boundary. Making this placement is not straightforward since from each pass we only know the relative position of a set of boundary points, and we do not know anything about their situation relative to the points recovered from the other pass. Assignment of one coordinate system to points of pass i , and another to the points of pass j , such that the points of every pass have fixed places in their own "master" coordinate system, reduces the problem to the following one: find the place for the origin of coordinate system j , in coordinate system i , such that all points fall on the true edge of the object.

Since no data is available about the position of points created in one pass relative to the ones created in another, this problem is unsolvable in general. We will therefore address the following easier problem: find the locus of places for the origin of coordinate system j , in the coordinate system i , such that there exists a connected object that could have generated the correspondingly merged data. We expect the merging process to be in many cases sufficiently constrained to effectively "reduce" the range of the objects that could have generated the combined data, so that each of them can serve as a practical approximation to the solution of the original and unsolvable question.

The criteria for finding a legal relative placement are the following.

(a) Extent criterion: the union of points from the two passes can be on the object boundary only if they belong to the interior of a parallelogram resulting from the intersection of two infinite strips of width 2, with directions corresponding to the velocities of the two passes.

(b) Intersection criterion: since "black" segments contain only interior points and "white" segments only exterior points, there can be no intersection between "black" and "white" segments in a legal placement. (If there is one, the intersection point would be both an exterior and interior point to the object.)

Addressing the problem in a more formal way, let us denote by $\{S_k\} = \{(X_b, Y_l)_k | l = 1, \dots, N_k\}$ the set of N_k edge points of pass k , where k is either i or j , measured

in its own "master" coordinate system. Further define by $\{S_k + r_k\}$ the set $\{(X_b, Y_l)_k + r_k | l = 1, 2, \dots, N_k\}$ where r_k denotes the placement of the set of points S_k in an arbitrarily chosen, world coordinate system. It is required to determine the two vectors r_i and r_j so that the resulting set of points $\{S_i + r_i\} \cup \{S_j + r_j\}$ may belong to the boundary of a connected two-dimensional shape. As the world coordinate system is arbitrary, it can be chosen to coincide with the system of pass i , i.e. $r_i = 0$. Hence the problem reduces to finding the vector r_j only.

We shall call the set of all "legal" r_j 's, which is the range of placements for the origin of coordinate system of pass j relative to the coordinate system of pass i , while satisfying the two criterions, the basic placement range of j w.r.t. i or $R_b[j|i]$. Clearly $R_b[j|i] = -R_b[i|j]$.

To understand the implications of the placement criteria described above let us examine a simple example: each set of edge points consist of only two points (one "black" segment, Fig. 7a). Using only the intersection criterion, nine "topologically" distinct regions for placing j 's origin of coordinates on i are created. Five of them are "legal" and four are not. Note that each region is characterized by the fact that for all r_j in it, the same segments intersect. Hence the points of these regions are equivalent with regard to intersection (see Figs 7b and c for a map of the regions and for representative examples). If the origin of pass j is placed at a point in region No. 1, there is an intersection between two black segments and the intersection criterion is satisfied. If the origin of j is placed at a point in one of regions denoted Nos 2, 3, 4 or 5 there is an intersection between two white segments and the intersection criterion is again

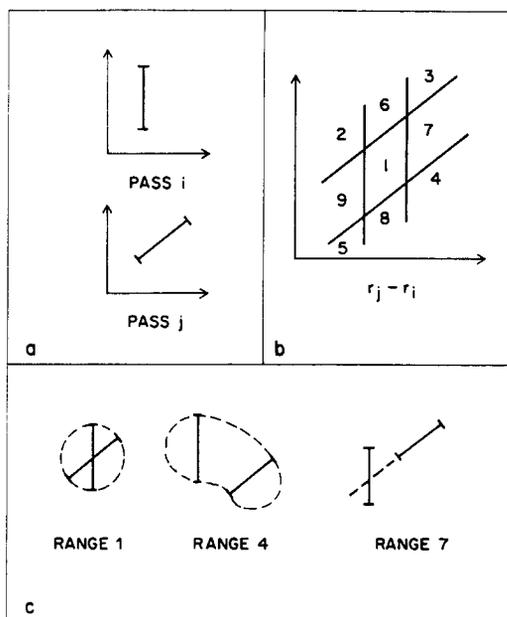


Fig. 7. Ranges and placements for a simple example.

satisfied. If, however, the origin of j is placed in a point in one of regions No. 6 or 8 there is an intersection between a white segment of i and a black segment of j , and the intersection condition is violated. Also if the origin of j is placed in a point in one of regions No. 7 or 9 there will be an intersection between white segment of j and black segment of i , and the intersection condition is again violated.

As can be seen the legal placement regions are parallelograms or infinite "open parallelograms". If we impose the extent criterion too, the resulting legal regions would be the intersection of the five legal regions with the legal placement area resulting from only the first criterion. In this process all placement regions will necessarily become of finite extent and some of them may disappear.

In the more general case where every group of edge points consists more than two points, we determine legal placement regions with similar rules. The results are usually much smaller areas due to the multitude of conditions (between every black and every white segment) that must be fulfilled. Note that the basic placement range will always be a collection of parallelogram shaped regions. This is so since a multitude of intersection conditions imply that the legal placement area is the intersection of many "elementary" parallelogram-shaped regions whose edges are in the same directions.

Once the entire legal range has been determined, any point in it will provide a "correct" relative placement of the data from the two passes, and for any such placement it is possible to find an object such that the merged edge points from both passes lie on its boundary. We may then want to choose from the infinity of such correct placements the ones that provide boundary reconstructions which meet further criteria (like smoothness of the boundary curve).

3.4. *Shape information from more than two passes*

Let us next address the more complicated problem of integrating information from more than two independently collected edge crossing measurements.

Suppose we are given P groups of edge points resulting from P different passes. Let us denote by $\{S_k\} = \{(X_l, Y_l) | l = 1, \dots, N_k\}$ the set of N_k points of pass k , measured in their own "master" coordinate system. Further define by $\{S_k + r_k\}$ the set $\{(X_l, Y_l)_k + r_k | l = 1, \dots, N_k\}$ where r_k denotes the placement of the set S_k in an arbitrarily chosen world coordinate system. It is required to determine a set of placement vectors $\{r_k | k = 1, \dots, P\}$ so that the resulting set of points $\bigcup_{k=1}^P \{S_k + r_k\}$ may belong to the boundary of a connected two dimensional object, under the two criteria stated in the previous section. We can choose the world coordinate system to coincide with the "master" coordinate system of, say, the first pass ($k = 1$), since only the relative displacement of the sets of points matters.

Our problem is then stated as follows: given the

basic relative placement range $R_b[j|i] (= -R_b[i|j])$ for all pairs of passes i, j , determine a placement of P points (origins of the master coordinate systems of the P data sets) in the plane via the set of vectors $\{r_k | k = 1, \dots, P\}$ with $r_1 = 0$, so that $r_j - r_i \in R_b[j|i]$, for all i, j .

To get insight on the interaction between the constraints, we first examine the case of three passes. We show that information from a third pass further restricts the relative placement of points resulting from the first two passes. Assume that the origin of pass 3 was placed at a particular point r_3 and the origin of pass 2 was placed at another point r_2 , in a world coordinate system which coincides with the "master" coordinate system of pass 1 ($r_1 = 0$). The placement of edge points of the three passes satisfy the extent and intersection criteria provided the following relations hold:

$$\begin{aligned} r_3 &\in R_b[3|1] \\ r_2 &\in R_b[2|1] \\ r_2 - r_3 &\in R_b[2|3] \end{aligned}$$

where recall that $R_b[j|i]$ is the basic legal range of placing points from pass j in the coordinate system of pass i . These relations mean that r_2 , besides the requirement of belonging to its basic range $R_b(2|1)$ is further restricted to be in:

$$r_2 \in \{r | r = r' + r''; r' \in R_b[3|1] r'' \in R_b[2|3]\} = \{R_b[3|1]\} * \{R_b[2|3]\}$$

where $*$ denotes the set convolution operation. (Therefore $r_2 \in \{R_b[2|1] \cap \{R_b[2|3]\} * R_b[3|1]\}$), which shows that, the range for placing the origin of pass 2 will be smaller than the one implied by the interaction between pass No. 1 and pass No. 2 alone. See Fig. 8 for an illustration of the above argument.)

A method for finding a placement that fulfills the placement constraints was found. It is based on a linear programming algorithm whose variables are the coordinates of all master coordinate systems origins, (which make a total of $2P$ variables).

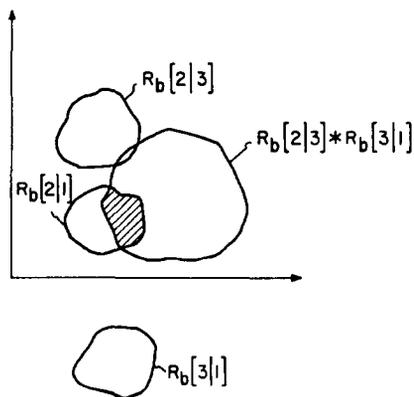


Fig. 8. Restriction of a placement range $R_b[2|1]$ due to information from a third pass.

Suppose that each $R_b[j|i]$ consists only of one convex region. (Later we shall discuss the case in which the range is a union of more convex regions.) Since the boundary of $R_b[j|i]$ is a convex polygon (in our particular case a parallelogram), the region is the intersection of (four) half planes. The inclusion of a point in a half plane corresponds to a linear inequality constraint on its coordinates x and y . Therefore the region inside a parallelogram is described by a set of four inequalities. In consequence the constraint $r_j - r_i \in R_b[j|i]$ may be rewritten as a set of four inequalities in the vector variables r_j and r_i . So the $2P$ variables of the problem will be constrained by $2P(P - 1)$ inequalities.

Linear programming procedures accept inequalities as constraints, and introduce an additional variable for each of them.⁽¹²⁾ Recall that besides the above constraints, we also have two equality constraints

anchoring the origin of pass 1 to some reference point. The total number of variables therefore becomes $2P + 2P(P - 1)$ and the total number of constraints is $2P(P - 1) + 2$ (besides the usual requirement to have all variables positive).

Since all information on relative displacement resides in the constraints, no further restriction is imposed on the cost function. Every linear cost function will produce an acceptable solution, which lies on the boundary of the feasible solution space (a convex region). In fact linear programming is used here only as a method to determine a feasible solution for a given system of linear inequalities. Therefore we can determine more feasible placements via convex combinations of different feasible solutions, corresponding to various choices of the cost function. An object model, if available, could in principle direct the choice of the cost function, although it is not obvious

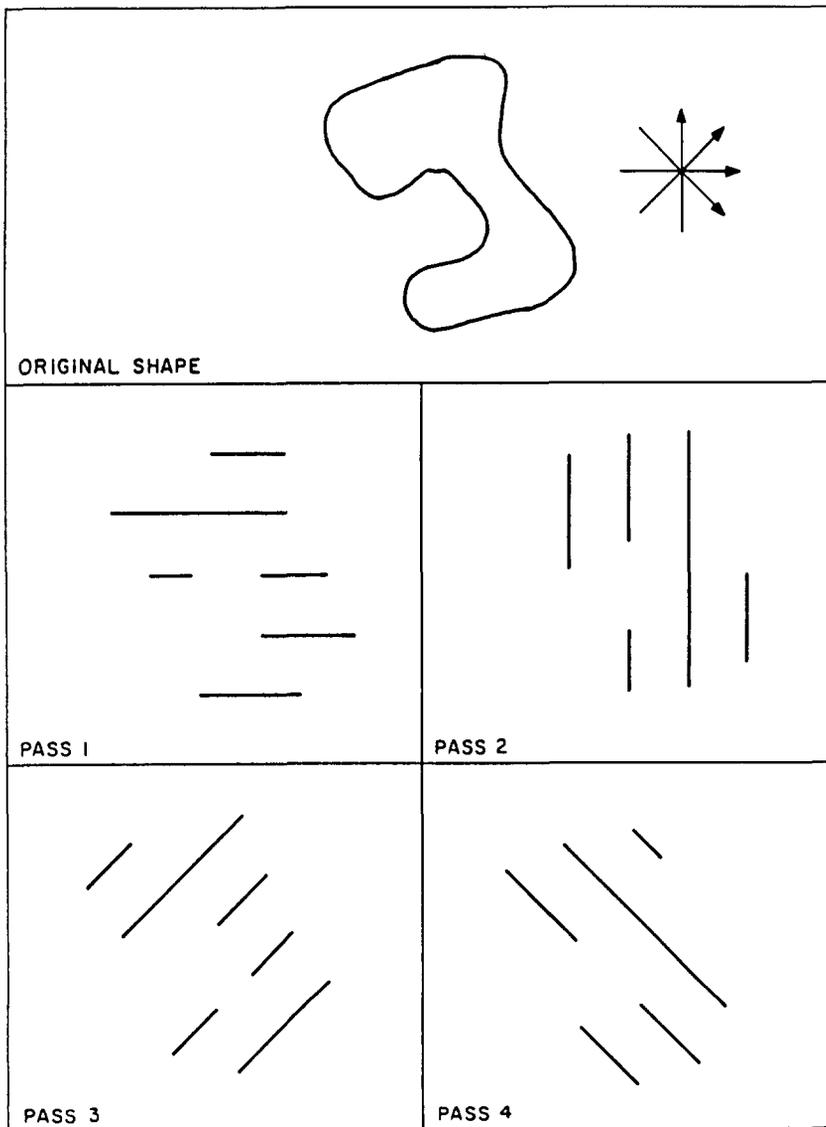


Fig. 9a. Simple shape and backprojected inner segments derived from it with four passes.

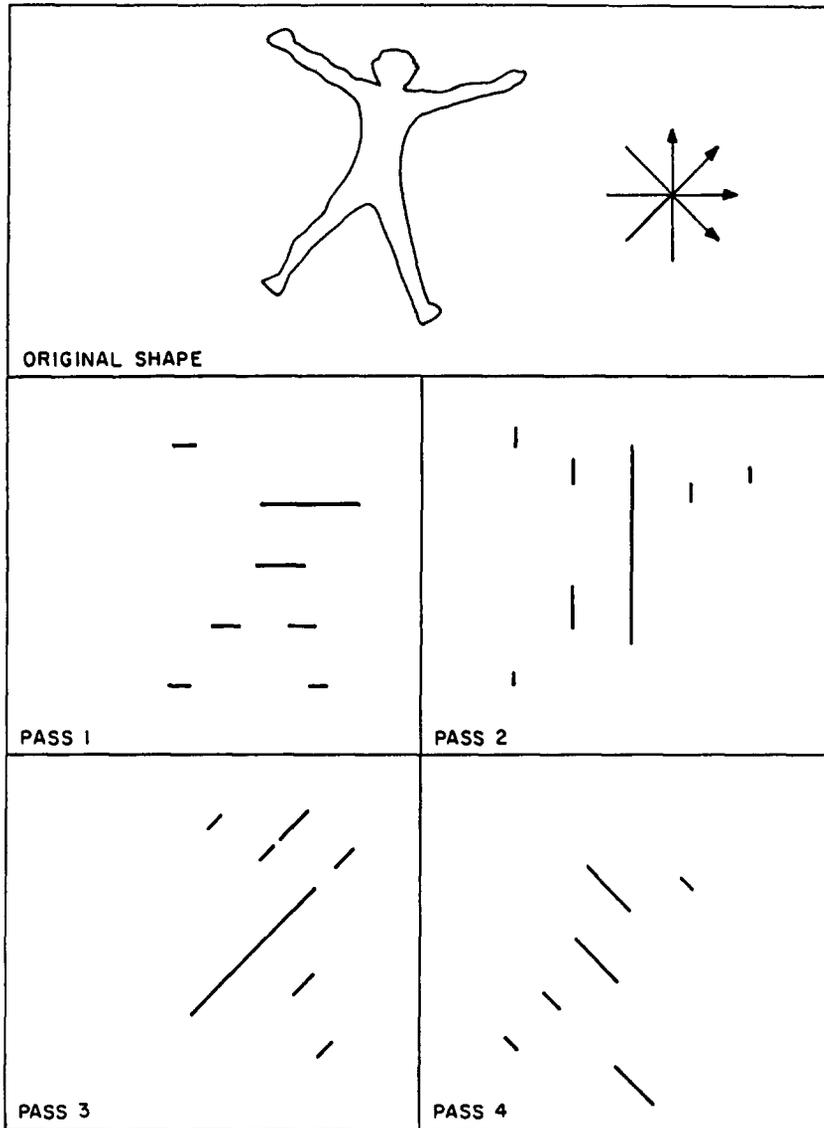


Fig. 9b. More complicated shape and backprojected inner segments derived from it with four passes.

how to incorporate such information.

The complete algorithm determining a feasible solution is the following.

Step 1. Between every pair i, j of different passes find the basic range $R_b[j|i]$. (Each basic range consists of one or more convex regions.)

Step 2. From each range choose one convex region (which is bounded by a parallelogram).

Step 3. Using these regions, determine constraints for linear programming problem and try to solve it.

Step 4. If a feasible solution was found this will necessarily be a solution of the original problem. If not, it means that no solution exists for the particular combination of convex regions chosen in Step 2. Then proceed to Step 2 and choose a different combination of sub-regions.

In at least one of the chosen combinations a solution will be found, otherwise there would be no solution to

the original problem, a case we excluded by assumption. After fixing all places of origins relative to pass one (which serves as the world coordinate system), all edge points are fixed. It is then possible to find an object having all the given points on its boundary.

4. SIMULATION RESULTS

To demonstrate the above discussed shape reconstruction procedure we have chosen to run simulations on objects with arbitrary shape. It was assumed that the objects pass in translation over a detector arrays having five elements. We assumed that for each pass the detectors are arranged with equal spacing in a linear array whose direction is perpendicular to the velocity. Figure 9 shows the two objects on which the simulations were performed, the four translation directions assumed for four independent passes and

the backprojected data provided by the edge detectors arrays.

Using this data, the basic placement ranges $R_b[j|i]$ for $i, j = 1, 2, 3, 4 \ i < j$ were determined. This was done by invoking a program that computes the permitted ranges for the placement of every black and white segment of the projection resulting from pass j on the corresponding segments of pass i , and intersects them. The six basic ranges are shown in Fig. 10.

Arbitrary placements taken in these basic ranges will yield placement of the data from two passes which satisfy both intersection and range criteria. Choosing, for example, the point LP indicated in $R_b[2|1]$ in the case of the two objects will yield the placements of Fig. 11a which are legal. If however, the points are chosen outside the basic ranges like the point IP indicated in $R_b[2|1]$ it will result in an illegal placement (Fig. 11b).

Note that some of the basic ranges consist of several convex subranges bounded by parallelograms. Such split ranges are typical and from a computational point of view this means that determining a combined legal placement of the data from all passes will necessitate analyzing all combinations of such subranges, in the worst case. This is so since application of the linear programming procedure requires all basic ranges to be convex.

In our examples all combinations of basic subranges were checked. In those combinations in which a legal placement was found the linear programming algorithm was applied with a sequence of different cost functions, requiring maximization of the additional variables one after the other (which forced departure from the corresponding boundary). Note that all the solutions obtained in this way are topologically equivalent in the sense discussed in the previous

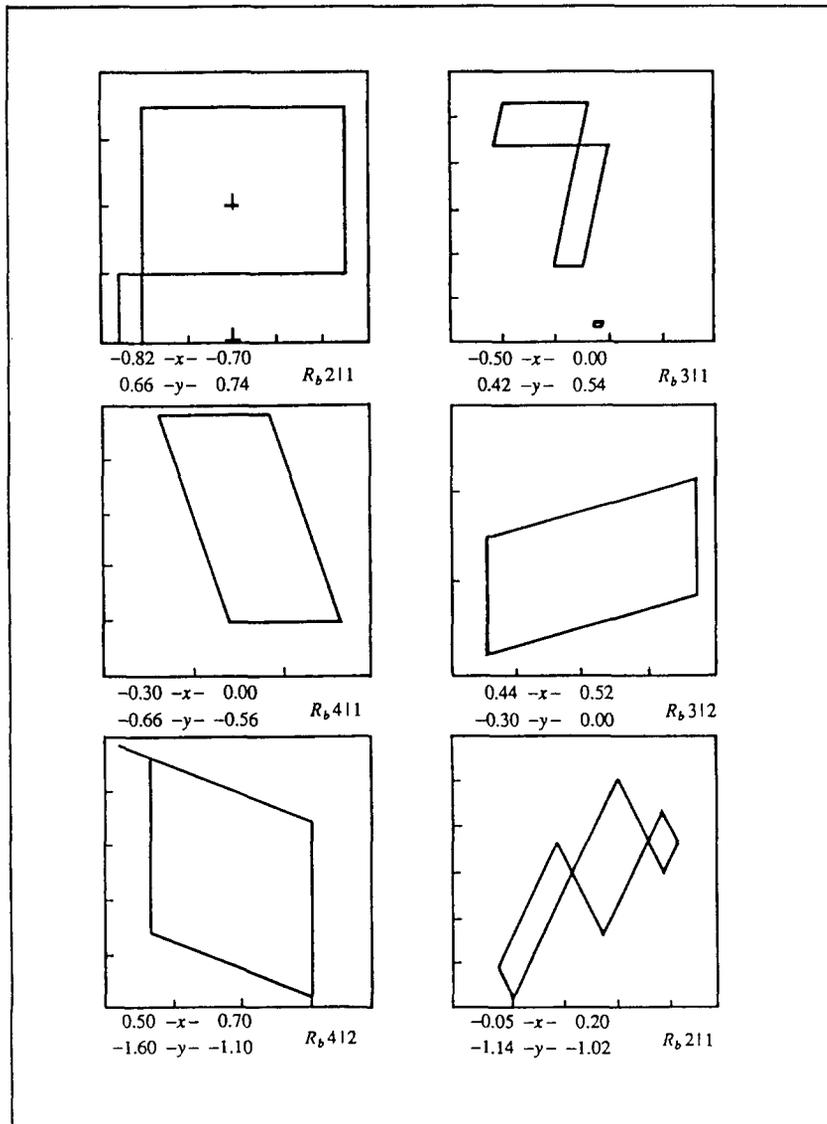


Fig. 10a. Basic ranges obtained for the first example.

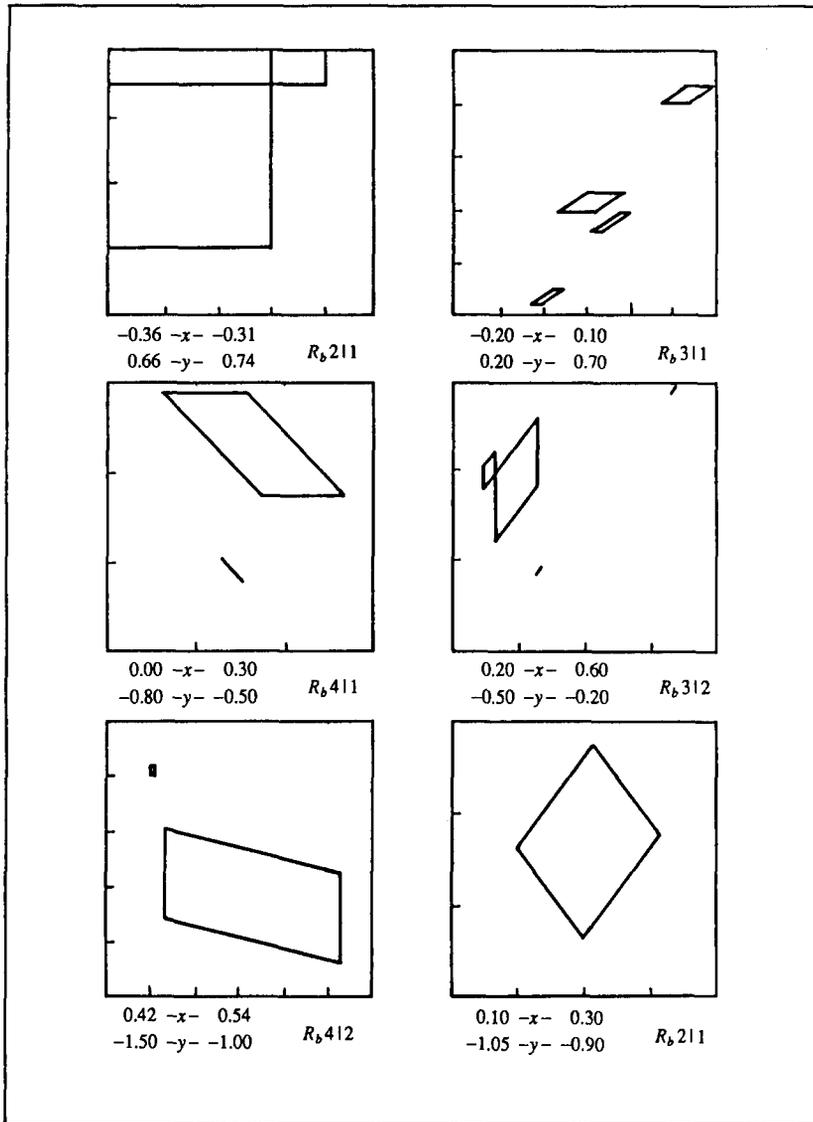


Fig. 10b. Basic ranges obtained for the second example.

section, i.e. the same black and white segments intersect. Figure 12 shows superimposed all the solutions obtained for a specific combination of subranges (the biggest area subranges for each $R_b[j|i]$) in order to give an idea of the variability of solutions within a topologically equivalent set. For the first object analysed, the simpler one, there were 36 possible subrange combinations and feasible solutions were found for 10 of them. For the more complicated object only two subrange combinations out of 128 possible combinations yielded feasible solutions. Figure 13 shows that, for the two objects analysed, the placement obtained by averaging topologically equivalent solutions for several combinations of subranges which yielded feasible solutions.

To enable the evaluation of the scatter of topologically nonequivalent solutions, Fig. 14 shows such topologically different solutions superimposed,

one for each subrange combination. We see here that, in the case of the more complicated example the scatter in placement is smaller. This is somewhat expected, since a more complicated object will impose more restrictions on the relative placement.

From these simulations we may conclude that the variability of solutions is not very big, and good eye evaluations of the shape may be obtained from any solutions picked at random. In fact, for some simple shapes a good evaluation can be obtained from only two passes (or with less detectors), as can be seen in Fig. 11.

5. DISCUSSION

In this paper we discussed a method for learning about the shape of moving objects from data supplied by sparse arrays of motion detectors. It was shown

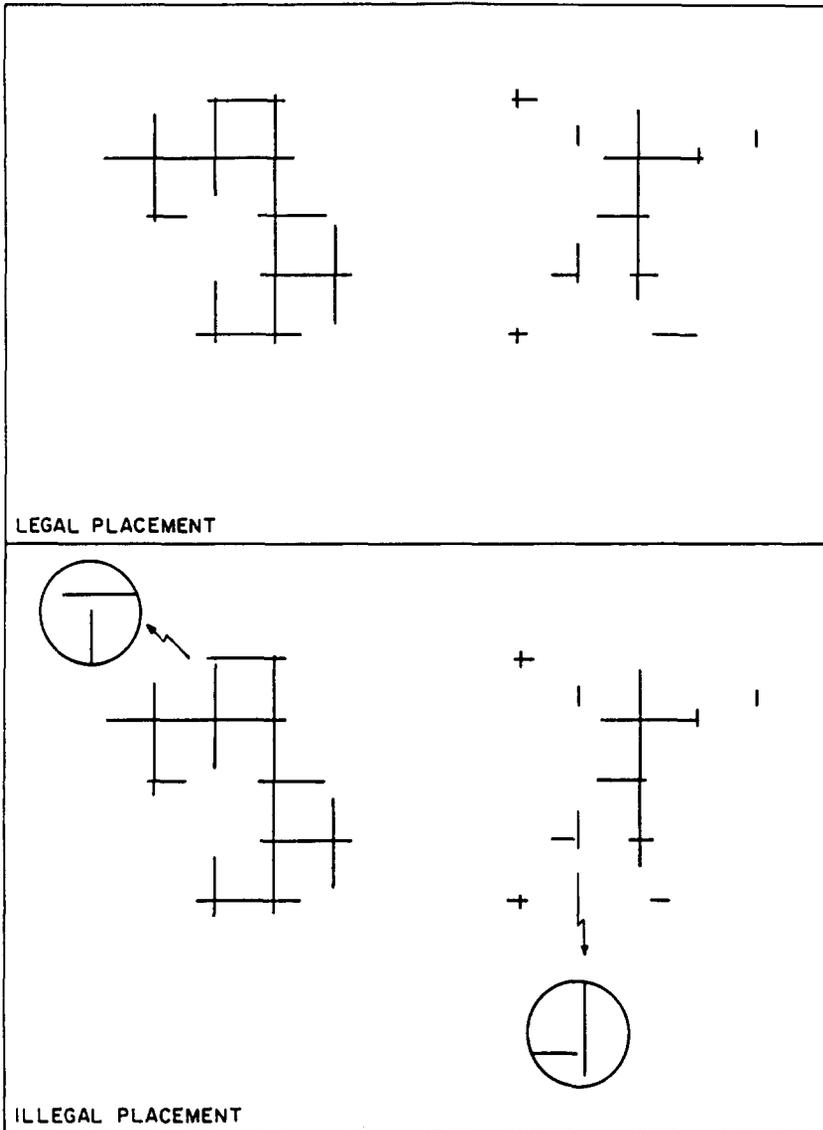


Fig. 11. Legal and illegal placements of inner segments from two passes.

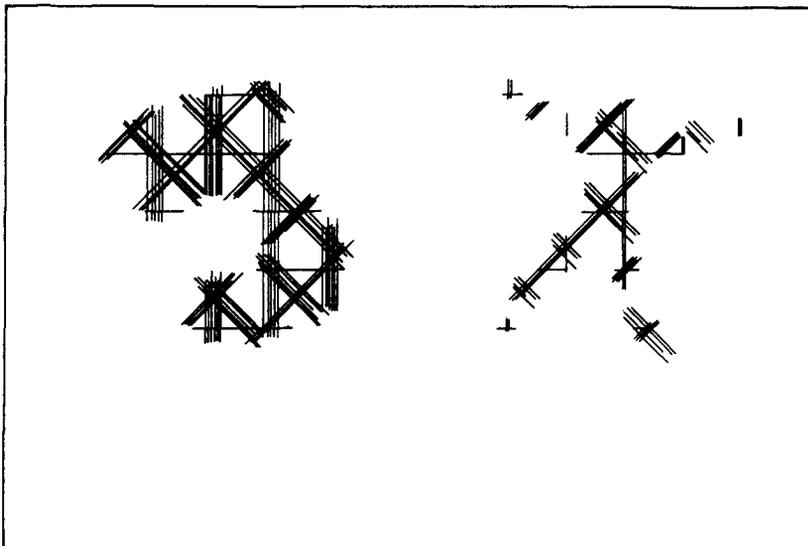


Fig. 12. Different topologically equivalent solutions superimposed.

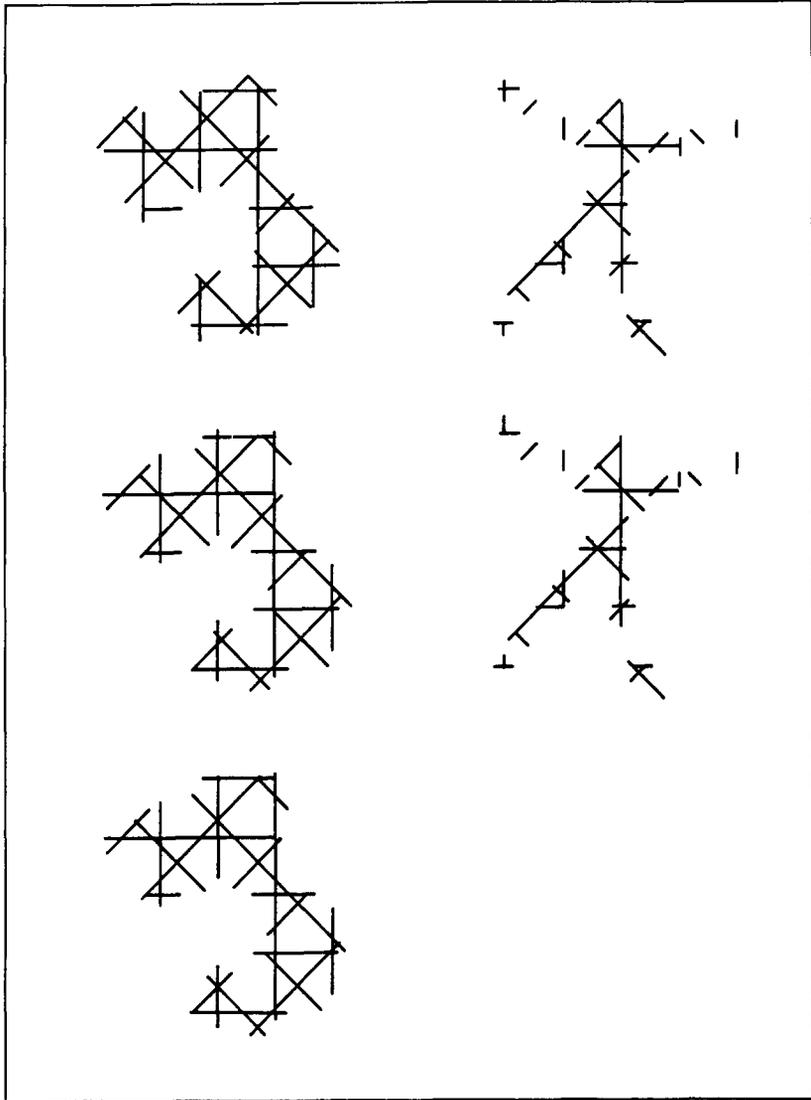


Fig. 13. Examples of legal placements for all the data four passes.

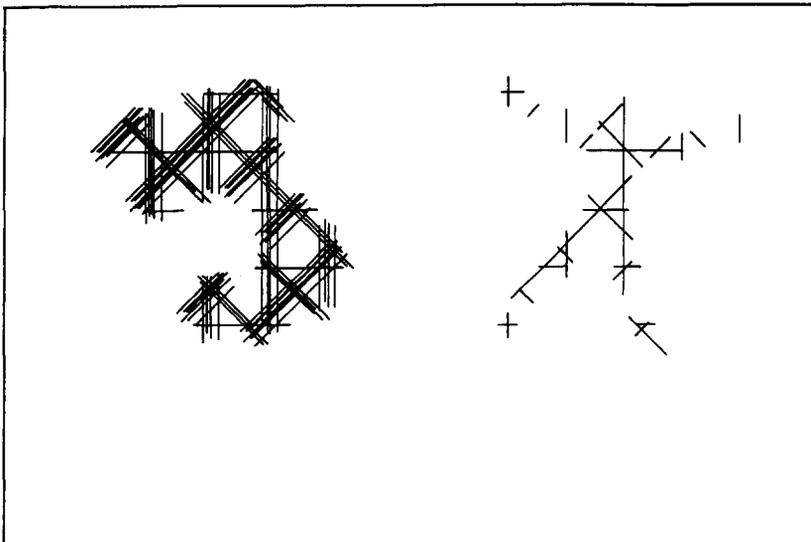


Fig. 14. Different topologically nonequivalent solutions superimposed.

how to approximate the shape of two-dimensional objects moving in translation, given relatively small numbers of edge crossing measurements that provide the edge direction and the velocity perpendicular to it. An algorithm that combines information from independent sets of measurements, in order to find constraints on classes of shapes that are consistent with all the available data was also developed.

First, we showed that the object velocity can (usually) be determined for every sweep of the object over the detector array. A backprojection algorithm was used in order to position a set of points on the contour of the object. This information is supplemented by edge slope information at those points and a set of line segments which are known to belong to the interior of the object. If data from only one passage of the object on the detector array is available, then only these sets of points, the corresponding edge slopes and "included" segments (parallel to the direction of velocity) are determined and they are known to sample the true boundary and interior of the object. In fact, if many detectors would be densely packed over the area through which the object sweeps, they would sample the object at very small spacing and the shape reconstruction from one pass would consequently be very good. We however assumed a sparse sampling during every passage over detector arrays and analysed ways to combine information from multiple passes in order to improve the shape estimate.

In many cases perceiving an arbitrary shape is not as important as the identification of patterns from a given library of objects. One example can be instincts which have influence on the behavior of animals. It seems that they are "wired in" and are not subject to easy modifications by learning, and also, do not require a complete, fully identified stimulus for triggering. Another example is computerized inspection systems in manufacturing processes which have the prior information on the expected shapes from the design stage. The scheme described in this paper is suitable for this task since the known points on the edge together with their corresponding edge slopes can be used as a direct input to the generalized Hough transform method proposed by Ballard in Ref. (10). In principle this method works as follows: for every shape, in the learning stage, an arbitrary point is chosen as an origin. A Table containing directions and distances to edge points from this origin, together with the corresponding edge slopes serve as a representation of the shape. This knowledge can be based either on external knowledge or on a learning stage using the shape approximation from many passes. In the recognition stage every edge measurement yields votes for possible origin positions and the votes are accumulated in a two dimensional array. This process is done separately for every assumption, i.e. for every different possible shape from the library, and in a cumulative way for every edge point. After the process is finished the shape with the greatest number of votes for some bin is selected. The shape is thus identified

and so is its placement with no need for the whole shape acquisition. This technique may also be adjusted to identify an object rotated w.r.t. its representation.

Some generalizations of this work can readily be thought of. For example, the restriction of constant speed object translation can easily be removed and more general models can be assumed at the expense of increased number of measurements and more calculations. For example, assume that the movement is translation by the velocity: $\vec{V} = \vec{V}_0 + \vec{V}_1 \cdot t + \vec{V}_2 \cdot t^2 + \dots + \vec{V}_N \cdot t^N$. With this assumption the velocity determination requires $2N$ measurements and the solution of a linear system of $2N$ equations. Another motion model comprising constant translation added to object rotation at a constant rate requires the solution of nonlinear equations but is analytically tractable. In these cases, approximation of shape from multiple passes cannot be done with the method proposed above. However, the tasks of finding points on the boundary, fitting convex extent polygons and shape recognition using Hough transform remain feasible. These generalizations, as well as others, like dealing with grey level images and with multiple objects are currently under investigation.

Some other works use data similar to the type we use but in a different way (see Ref. (11) for an example). In these works the intersection of the shape with a set of straight lines is checked and statistical properties like average length of chord, chord count etc. are computed and are shown to have a direct relation with simple shape parameters like area and perimeter. No statistical analysis was made in our work and the measurements were used directly to determine a non-parametric shape approximation.

The main problem that arose when integrating data from several passes was the determination of legal relative placement of points known to be on the object boundary within their own master coordinate system (corresponding to information from one pass). This led to the interesting and general mathematical problem of placing P points so as to have their relative position vectors belong to predetermined sets. This problem was solved for the particular case of convex polygonal regions as constraint sets via linear programming. We note however that the general placement problem seems to be quite difficult. In fact, if the constraint sets $R_p[j|i]$ comprise a discrete number of vectors, the problem becomes a difficult combinatorial one. (It is at least as difficult as NP -complete, as can be shown via reduction of a special case to set partitioning.⁽⁶⁾)

After determining a legal set of edge point locations, the reconstruction of the object shape is not yet complete, since also the order of those points along the boundary has to be determined. This process is crucial to the next stage which will be the approximation of the boundary with some curve, taking into account the slope information. A solution to the edge point ordering problem can be based on the following idea: The lines on which the interior segments are placed (the backprojection rays), divide the plane into

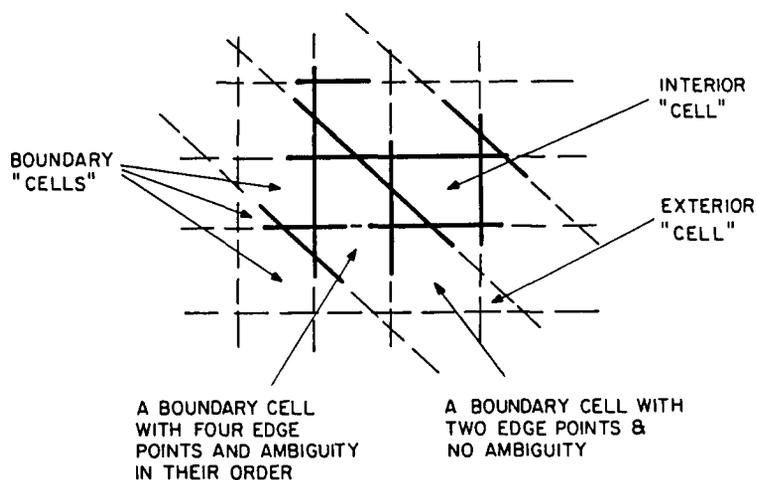


Fig. 15. Different types of regions on which ordering of edge points can be based.

polygonal regions. These regions can be classified into three categories. "Interior" regions are ones which are bounded only by black segments, "exterior" regions are ones which are bounded only by white segments and "boundary" regions are the rest (see Fig. 15). The object contour curve passes through the boundary regions, and in order to cross the border between two adjacent "cells" it must pass on a boundary point. When only two edge points are on a boundary of a cell one must follow the other in the sequence of points along the boundary, but when there are four or more there is an ambiguity in the determination of order. However, all the possible orderings of passage can readily be listed and analysed.

Another interesting problem we did not discuss yet is the determination of optimal placement for the local velocity detectors. The optimality criterion may be a requirement that the maximal spacing between detectors in directions perpendicular to the object velocity will be minimal over all the possible directions of object velocities.

In conclusion, we presented here a new way for acquisition of information on dynamic scenes, different from the traditional ones. We have shown that such a system can analyse motion, find position, and recognize a known pattern through a small number of measurements. The system can in principle also learn about new objects, i.e. make approximation for their shapes, although the computational burden for doing this is quite large.

Acknowledgements—We thank Andrei Broder of DEC-Western Research, Palo-Alto, for some over-the-network

discussions on complexity, Shmuel Wimer, of IBM's Haifa Research Center, Haifa, for supplying us a linear programming package and Hanna Bismut for the nice drawings.

REFERENCES

1. D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice Hall, Englewood Cliffs, NJ (1982).
2. S. Ullman, Analysis of visual motion by biological and computer systems, *IEEE Comp.* **14**(8), 57–69 (1981).
3. S. M. Anstis, PHI movement as a subtraction process, *Vision Res.* **10**, 1411–1430 (1970).
4. K. Hausen, Motion sensitive interneurons in the optomotor system of the fly, *Biol. Cybernet.* **46**, 67–79 (1982).
5. W. H. Zaagman, H. A. K. Masterbroek and J. W. Kuiper, On the correlation model: performance of a movement detecting neural element in the fly visual system, *Biol. Cybernet.* **31**, 163–168 (1978).
6. M. R. Garey and D. S. Johnson, *Computers and Intractability*. Freeman, San Francisco, CA (1979) and Personal communication with A. Broder.
7. P. Kahn, Local determination of a moving object contrast edge, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**, 402–409 (1985).
8. B. K. P. Horn and B. G. Schunk, Determining optical flow, *Artif. Intell.* **17**, 185–203 (1981).
9. L. S. Davis, Z. Wu and H. Sun, Contour based motion estimation, *CVGIP* **23**, 313–326 (1983).
10. D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition* **13**, 111–122 (1981).
11. F. W. Sinden, Shape information from rotated scans, *7th Int. Conf. Pattern Recognition, Montreal*, pp. 1278–1280 (1984).
12. H. A. Taha, *Integer Programming*. Academic Press, New York (1975).

About the Author—MICHAEL LINDENBAUM was born in Tel-Aviv, Israel on January 20, 1956. He received the BSc. Degree in Electrical Engineering from the Technion, Israel Institute of Technology, in 1978. From 1978 to 1985 he has served in the IDF as an electrical engineer. Now he is completing his MSc. degree in Electrical Engineering in the Technion.

About the Author—ALFRED M. BRUCKSTEIN was born in Sighet, Transylvania on January 24, 1954. He received the BSc. and MSc. Degrees in Electrical Engineering from the Technion, Israel Institute of Technology, in 1977 and 1980, respectively, and the Ph.D. degree in Electrical Engineering from Stanford University, Stanford, California, in 1984.

Since October 1984, he has been with the Faculty of Electrical Engineering at the Technion, Haifa. His research interests are in estimation theory, signal and image processing, algorithmic aspects of inverse scattering, point processes and mathematical models in neurophysiology.

Dr Bruckstein is a member of SIAM and Sigma Xi.

APPENDIX

IMPLEMENTATION OF LOCAL VELOCITY MEASUREMENT

Optical flow is a general way to describe motion in an image. It is a vector field related to the dynamic image, the field value at every point being the local velocity vector of the same point in the original image. The optical flow can sometimes be obtained from a series of successive static images, which are time samplings of the dynamic image, using methods that are classified into token matching techniques vs intensity based ones.⁽²⁾

The methods of the first class identify feature points in one static image, and search for them in the following images using some assumptions about the nature of motion. Then the relative displacement of points and the time interval between frames provide the velocity. This method usually yields the best results but is clearly neither local nor computationally efficient.

The alternative methods are based on local processing, however, locality implies that it is not possible to determine the true velocity vector at each point but only its component in the direction of intensity gradient. (This problem is known as the aperture problem and is inherent to local processes. For example, any edge, when observed locally, is seen as a straight one and as such the velocity component tangent to its direction is impossible to detect.) We shall call this data the "locally measurable" velocity field. It constrains the true local velocity to a line in the velocity space. If the optical flow is to be determined a further problematic processing must follow in which the true velocity field is determined from the locally measurable velocities by assumptions on the motion in the dynamic imagery.⁽⁸⁾

Our method is based on the locally observable velocity field, therefore we shall briefly mention two ways of implementing local velocity detectors.

Gradient based schemes rely on the equation

$$dI/dt = I_x \cdot V_x + I_y \cdot V_y$$

relating the time derivative of the intensity at the point in the image, to components of the true velocity (V_x, V_y), via the intensity gradient in the x, y direction (I_x, I_y). Given $dI/dt, I_x, I_y$, the above relation becomes an equation of a line in V_x, V_y plane, to which the true velocity is constrained. (Note that in order to determine I_x and I_y it is also required to have access to a static image).

Correlation methods are more adequate for our problem since the gradient of a binary image is undefined at edges and also there is no need for static images. Such detectors provided a successful model for processing of visual information by insects, as measured with experiments on their optomotor reflexes.⁽⁵⁾ A typical one dimensional detector of this sort is illustrated in Fig. A1a. In Fig. A1a, p_1 and p_2 are intensity detectors; an image moving in the direction from p_2 to p_1 , at such a velocity that the signal at p_1 will be equal to the signal from p_2 after the delay will provide full correlation between the readings of the two detectors and therefore maximum output. For example if the input signal is a short

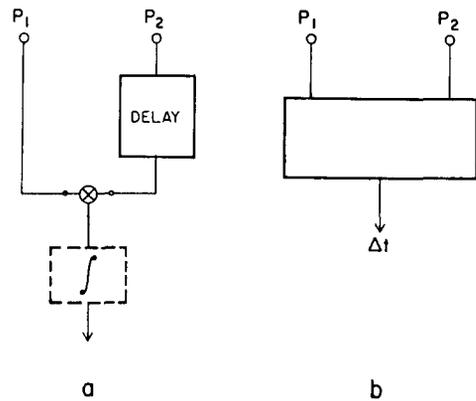


Fig. A1. Correlation detector.

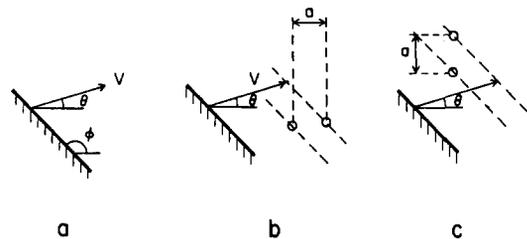


Fig. A2. Local detector for straight edge velocity measurement.

pulse and we will integrate the output of the multiplier the result will be maximum only at the "correct" particular speed and deviations from this speed may be inferred from the output.

This scheme has some drawbacks, like the limited range of velocities that can be detected and an inherent ambiguity in the determination of the velocity for periodic signals. These faults can be cured by using detectors with several different time delays instead of one. For our purpose what is important to observe is that this detector basically measures the time elapsed between the appearance of the signal in p_2 and its appearance in p_1 . From now on, we assume that such a detector only gives the time between two such events, and we shall not be concerned with the particular implementation of this function (Fig. A1b).

It is not difficult to see that two such detectors provide the edge crossing information mentioned in the formulation of the problem (Section 2), i.e. the component of velocity normal to the edge, the slope of the edge and, of course, the time of passage (which is trivially just the time one of the intensity detectors changes its output). Suppose a straight edge is moving with velocity V and direction θ (Fig. A2a). If the edge passes a horizontal detector (Fig. A2b) the detector will sense a passage time of $T_h = a \cdot \sin(\phi) / V \cdot \cos\left(\phi - \frac{\pi}{2} - \theta\right)$. Similarly a vertical detector (Fig. A2c) will sense a passage time of $T_v = a \cdot \cos(\phi) / V \cdot \cos\left(-\frac{\pi}{2} - \theta\right)$. By dividing these times we

can get the slope of the edge - ϕ , and then we can find the normal component of the velocity $V_n = V \cdot \cos\left(\phi - \frac{\pi}{2} - \theta\right)$. Clearly, we shall not be able to find V and θ separately, no matter how many detectors, in different directions we will place on the image (the aperture problem). A double detector (one horizontal and one vertical) with dimension much

smaller than the minimal radius of curvature of the object, can indeed provide the information we need. Note that an array of three intensity detectors p_1, p_2, p_3 with time of passage monitoring, arranged on the vertices of a right angled triangle, are equivalent to having two such correlation detectors. In fact it is true that any triangular array of intensity sensors will do.⁽⁷⁾