



STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

File Copy

January 1970

EXPERIMENTS IN SCENE ANALYSIS

by

Richard O. Duda
Peter E. Hart

To be published in the Proceedings of the
First National Symposium on Industrial Robots,
Chicago, Illinois, April 2-3, 1970.

Artificial Intelligence Group
Technical Note 20

SRI Project 8259

This research is sponsored by the Advanced
Research Projects Agency and the National
Aeronautics and Space Administration under
Contract NAS 12-2221.

EXPERIMENTS IN SCENE ANALYSIS*

by

Richard O. Duda
Peter E. Hart
Stanford Research Institute
Menlo Park, California 94025

Abstract

This paper describes an experimental computer program that analyzes pictures taken in a simple, but nevertheless real-world, robot environment. The analysis proceeds by building up, step by step, a partial line drawing representation of a digitized television picture. An interesting feature of the system is an executive program that uses detailed knowledge of the environment to control other programs that extract the partial line drawing. Examples are given to illustrate the operation of this experimental program.

* This research was sponsored by the Advanced Research Projects Agency of the Department of Defense and was monitored by the National Aeronautics and Space Administration under Contract NAS12-2221.

EXPERIMENTS IN SCENE ANALYSIS*

by

Richard O. Duda
Peter E. Hart
Stanford Research Institute
Menlo Park, California 94025

I INTRODUCTION

During the past several years the field of artificial intelligence has become increasingly involved in the problem of designing intelligent robots. While this is not the place to resurrect old questions about the definition of intelligence, an obvious prerequisite for interesting behavior in such a machine is the ability to obtain information about itself and its environment. We can imagine a number of ways in which this might be achieved. Perhaps the simplest is direct feedback about the geometrical configuration of the machine--kinesthetic feedback, one might say, to provide the machine with information about itself. Primitive information about the external environment of the machine can be obtained with simple touch or force sensors. Distance information about objects not within reach can be obtained through the use of some form of range finder. Finally, we can imagine sensors implementing the human senses of smell, hearing, or sight. Of all these possibilities (not to mention others for which there are no human counterparts), the richness and potential utility of the visual field has excited by far the most interest and attention. In this paper we will describe some experimental work aimed at developing processing tech-

* This research was sponsored by the Advanced Research Projects Agency of the Department of Defense and was monitored by the National Aeronautics and Space Administration under Contract NAS12-2221.

niques for extracting information from visual data.

The difficulty of the problem we are addressing is attested to by the paucity of published results. While the literature on classical pattern recognition is vast,^{*} there have been very few significant contributions to what we shall call scene analysis--the problem of describing the contents of a picture of a three-dimensional scene. The classic paper in the field is by Roberts (1965). His approach to the problem of describing pictures of simple geometrical solids is characterized by two distinct steps of processing. The first step attempts, through a series of operations, to reduce a digitized television picture to a perfect line drawing. The second step matches the line drawing, either in whole or in part, against a set of stored computer models of geometrical objects. The model achieving the highest degree of match with a given portion of the picture is taken as a description of that portion. Moreover, the spatial orientation of the model achieving the best match gives additional information about the position of the object with respect to the camera. A second important contribution was made by Guzman (1968). His work assumes the existence of a perfect line drawing of jumbles of geometrical objects. Each geometrical object is assumed to be solid (no thin sheets), hole-free, and bounded by planes, but is otherwise unspecified. From these assumptions, Guzman's methods are able to partition the picture into sets of regions such that each region in a set belongs to the same geometrical object. In other words, Guzman largely solved, under his assumptions, the problem of piecing together the visible parts of

* See, for example, Nagy (1968).

partially occluded objects.

If we put together the methods of Roberts and Guzman, we might arrive at the following paradigm for completely processing television scenes of jumbles of geometrical objects: make a perfect line drawing, apply Guzman's techniques to associate regions of the pictures with specific objects, then use Roberts' model-matching methods to identify each object. Although superficially reasonable, to our knowledge this paradigm has been successfully implemented only in very simple situations. The basic problem is that the pictures themselves contain perfect information only if the environment is severely restricted.* In general, for a given transducer--that is, a given visual pickup device and analog-to-digital converter--we might distinguish three different types of environments. In very simple environments the transducer can provide "perfect" visual information. In very complicated environments the transducer is hopelessly inadequate. Somewhere between these extremes are visual environments for which the transducer can provide information that is "adequate," but by no means perfect. Our research interest lies in this middle ground. For this class of environments, a scene analyzer will not be able to proceed in a purely hierarchical way. Instead, it will be forced to reconcile various pieces of evidence in the picture against each other and against prior information about the environment.

In this paper we will describe a scene analysis program that proceeds in this spirit. It consists of two components. The baseboard tracking program tries to delineate the boundary of the room by locating a baseboard at

* For an example of scene analysis in a clean environment see Pingle and Wichman (1968). Forsen (1968) illustrates some of the difficulties that arise in more complicated environments.

the bottom of the walls. The object recognition program attempts to locate and, if possible, identify simple geometrical objects situated in a laboratory room. The immediate motivation for both programs is the Stanford Research Institute mobile automaton. The automaton, described in more detail by Nilsson (1969) and Munson (1969), is a mobile, computer-controlled vehicle equipped with touch sensors, an optical range-finder, and a standard vidicon camera. The camera and associated analog-to-digital converter produce a digital picture quantized in space to 120 x 120 picture cells and in intensity to 16 grey levels [see Fig. 2(b) and 7(b)]. Digital pictures such as these are the raw data for the programs described in the following sections.

II BASEBOARD TRACKING AND FITTING

The first part of the scene analysis program is the baseboard tracker and fitter. The function of this rather special-purpose program is to identify the baseboard in the picture and project it back into the room, thereby providing information about the position of the robot with respect to the walls. The program has two basic parts: a baseboard-identification routine and a line-fitting routine. We discuss each of them in turn.

A. Baseboard Tracking

The baseboard identification routine capitalizes on the fact that the baseboard in our room is dark, and that it has a known width. Accordingly, the routine scans the original grey-scale picture, column by column, looking for dark sections. Likely candidate sections are scored according to darkness and deviation from ideal width. The ideal width of a section is determined by computing the image width of the actual baseboard. The output of the tracking routine is a list of picture points, no more than one per

column, that are likely to be the lower edge of the baseboard. Conceptually, it is convenient to think of this data as being a new binary picture (that is, a picture having only two levels of intensity, black and white) showing the image of the lower edge of the baseboard. Typically, this binary picture will show several irregular line-like segments. Long sections of baseboard will usually have some gaps in them, either because of occluding objects or because the tracking routine failed to identify visible baseboard. Conversely, there will often be false hits--short segments alleged to be baseboard that in fact are not. In any event, imperfect or not, these irregular segments constitute the input to the line-fitting routine.

B. Line Fitting

The line-fitting routine has built into it several kinds of information about the environment. It uses this information to constrain, or limit, the set of candidate lines that it might consider. The simplest information it has is that the data must be fit with either one or two straight lines. (The camera may be aimed toward a corner of the room, but the lens angle is not wide enough to see two corners, and hence three sides, of the room.) The second piece of information utilized by the routine is that the walls of the room meet at right angles. Finally, the routine has a rudimentary, but for its purpose adequate, concept of not being able to see through walls. The program uses these forms of information to control the application of a less-informed line-fitting operation whose details will be described in the next section.

The first step in the process is an approximate fit to the single longest segment produced by the tracker. This rough fit is then perturbed

systematically over a small region of the picture in order to find the line that captures the greatest total number of segment points. Next, the routine examines each segment in the picture and computes a score measuring how well the segment is "explained" by the fitted line. The closer and more nearly parallel a segment is to the line, the higher the score. If all the segments are sufficiently well explained, the routine computes the position of the line on the floor (using the floor-location operation described in the next section) and returns the final answer.

If all the segments are not well explained, and are not so short that they can be ignored as noise, the routine tries to find a second wall. To do this, it appeals to some relations from projective geometry that sharply constrain the location of the second line. In particular, it uses the fact that the images of two perpendicular baseboards must pass through conjugate vanishing points. A vanishing point for a line in the floor is the point at which its image intersects the horizon line of the picture. The important property of conjugate vanishing points is that, given either one of them, the other may be easily computed [appropriate equations are given in Hart and Duda (1969)]. Thus, if we have fitted a line to one baseboard, and if the fitted line has a vanishing point x_1 , then the vanishing point x_2 of the image of the perpendicular baseboard is completely determined. This conjugate vanishing point is used in two ways. First, by constraining the fit of the second line to pass through the conjugate vanishing point we are assured that the two fitted lines will project back onto the floor at right angles. Second,

we can use the conjugate vanishing point in conjunction with the segments already explained by the first line to limit the region of interest.

This limitation is illustrated in Fig. 1. The second line must fall in the shaded region, since a line in any other part of the picture would correspond to a wall that either (1) hid the first wall, (2) was hidden by the first wall, or (3) was floating in air above the horizon. The importance of the constraint on the region in which the second line must fall is primarily concerned with eliminating spurious segments. The forbidden region of the picture often contains spurious line segments produced by the tracking routine. By removing those segments from consideration, the tracker makes far fewer errors than it would otherwise commit.

III OBJECT RECOGNITION

The purpose of the object-recognition program is to identify objects in the picture and to locate the position of each object in the room. The only objects the program "knows" about are rectangular parallelepipeds, triangular wedges, and doorways. Any object in the picture may be occluded, either because it is partially off-camera or because another object is in front of it. The program, however, attempts to identify only unoccluded objects. Occluded objects, if found, are merely labeled as "objects." The experimental setting is a laboratory room that will be described in more detail below.

The object-recognition program operates on two levels. The lower level consists of a number of routines that perform various tests on

specified regions of the picture. The upper level is an executive scene analyzer. Its function is to explore the scene by applying tests in the repertoire of low-level routines. When a given test is completed, the executive evaluates the answer in light of both previous test results and its own internal knowledge of the constraints of the environment. It then decides what test should be performed next and where in the picture the test should be applied. This process is continued until the executive amasses enough evidence from test results to make a decision. The strategy the executive uses in exploring a scene is to form a hypothesis about the picture and then call a test that will tend to confirm and sharpen the hypothesis. Initially the hypotheses will be very vague, such as "There is part of an object at picture point (i,j)." As the analysis proceeds, the hypotheses become sharper until an object is identified and located in the room. If a given test tends to weaken a hypothesis, then an alternative hypothesis is selected and the analysis continues along this track.

A. The Executive Program

The executive program described has been implemented as a decision tree. Each node in the decision tree corresponds to a partial hypothesis about the scene. Concomitantly, each node also corresponds to a fixed low-level test that will tend to sharpen that hypothesis. The branches out of a node correspond to the possible answers that the given test can return. Accordingly, exploring a scene is accomplished by traversing a path through the tree.

One of the problems of a conventional decision-tree program is that a wrong decision at any point usually results in an incorrect

final answer. If we are to overcome this problem, we need a mechanism that enables us to regard each decision as being only tentative. To this end, each low-level test associates a numerical measure of confidence with each of its answers. Thus, since branches correspond to test answers, each branch out of a node possesses a confidence value. At any stage in the scene analysis, various paths of the tree will have been traversed to various depths. Each partial path corresponds to a hypothesis, and a confidence for each partial path is computed from the confidences of the branches along that path. After each test is performed, the confidences for the various hypotheses are updated, and further exploration of the tree proceeds from the most confident hypothesis. In this way if a wrong decision is made and subsequent tests yield results of low confidence, we have a mechanism for backing up and exploring alternative hypotheses.*

When a path in the tree terminates and an object is recognized (or a failure announced), the region of the picture in the vicinity of the object is removed from consideration and a new analysis is begun on the remainder of the picture. This process is repeated until no more objects can be found. An intuitive understanding of the object-recognition program is most easily gained through examples illustrating its operation on typical scenes. Before giving examples, however, we must first describe the repertoire of low-level operators.

* For a discussion of some theoretical questions concerning decision-making in tree structures, see Hart (1969).

B. Low-Level Operators

Each of the low-level tests is performed upon a so-called gradient picture. The gradient picture is extracted from the original digitized picture by an operation that tends to enhance edges, or discontinuities, in grey level. We have used the operator described by Forsen (1968). Specifically, the point (i,j) in the digital picture is replaced by $|I_{ij} - I_{i+1,j+1}| + |I_{i+1,j} - I_{i,j+1}|$, where I_{ij} is the grey level of the digital picture at the $(i,j)^{\text{th}}$ cell. A line in the gradient picture thus corresponds to an edge in the original picture. Examples of thresholded gradient pictures are shown in Figs. 2(c) and 7(c). In each case the gradient has been computed and a point displayed if the value of the gradient was greater than or equal to two.

Most of the low-level operators are based on a simple procedure known as template matching or masking. A masking operation basically consists of computing the average gradient between two specified picture points in order to determine the likelihood of a straight line existing between the two points. A low-level operator may return either one answer or several, depending upon the operator and the scene. Every answer has associated with it a number between 0 and 100 that indicates the confidence of the answer or, more precisely, the strength of the response. The following is a list of short descriptions of each operation available to the executive-level program:

- (1) Line Connection. This routine does a limited local search in order to find the best line between two specified endpoints. It places masks between all pairs of points generated by small perturbations of the specified points and returns the endpoints associated with the strongest response.
- (2) Spurs. Spurs are short segments of lines radiating away from the endpoint of a specified line. The spur finder returns a list of such spurs and their confidences. It operates by computing mask responses for a set of 52 short masks arranged like spokes on a wheel. The table of mask responses as a function of angle is searched for local maxima, using hysteresis smoothing to avoid small irregularities without losing angular resolution. The spur corresponding to the maximum nearest to the specified line is rejected as probably being the line itself, and the routine returns those remaining spurs whose confidence exceeds some threshold.
- (3) Directed Spurs. This routine returns that spur at the end of a specified line that comes closest to running in a specified direction. The cosine of the angular error is used to measure confidence.
- (4) Verticals. This operator finds vertical lines by placing masks in a generally vertical direction. The operator

employs a relation derived from projective geometry* to compute the so-called vertical vanishing point--the point in the picture plane through which all images of vertical lines must pass. All the masks, when extended, run through this vanishing point. When a mask gives a strong response, a more detailed examination takes place. Typically, the vertical line (or lines) found will contain gaps because of noise. If a decision to bridge a gap is made solely on the basis of length, an unacceptably high number of errors results. Accordingly, the vicinity of a gap is examined for the presence of baseboard or spurs. A spur signifies that the gap is "real." A gap in the vicinity of a baseboard is usually caused by the lack of contrast between the dark baseboard and the object, and is therefore bridged. The use of this type of more "global" information, augmented by the purely "local" information about gap size, results in a far more reliable vertical line finder. Increased reliability, in turn, allows us to use vertical lines as the keystone of the decision-tree scene analyzer.

- (5) Following. This routine starts from a specified point and follows the gradient along a specified direction as long as the trace is sufficiently straight. Although this

* See Ahuja and Coons (1968) or Hart and Duda (1969).

operation is not too reliable, particularly regarding the determination of the proper endpoint, it usually provides a better determination of the angle between a given line and one of its spurs than can be obtained by the spur finder.

- (6) Baseboard. This routine tests whether endpoints of vertical lines occur in the vicinity of the baseboard. It is the only means by which information obtained by the baseboard tracker is passed to the object-recognition program.
- (7) Picture Point. This simple routine merely measures the distance between a specified point and the boundary of the picture. It is usually used to warn the executive that a line of interest may be running off the picture and out of view.
- (8) Floor Location. The final step in the analysis of a scene is often to locate the position of an object in the room. More precisely, given a point in the picture known to be on an object (a vertex of a cube, for example), we want to locate in space the corresponding actual point. To do this we make use of the fact that the process of taking a picture can be modelled as a collection of rays of light that pass through the camera lens and strike an image plane. Given a point on the image plane, we can follow the ray out through the lens

and on into space. In general, or course, we cannot determine where on this ray the corresponding actual point lies. Suppose, however, that we know at least one additional piece of information about the actual point. In particular, suppose we know that the actual point lies on the floor plane. Then we can compute where the ray of light intersects the floor and thus locate the position of an object with respect to the camera. In practice, then, when the image of an object is identified by the scene analyzer, the analyzer selects some points on the object known to be on the floor (the lower endpoints of vertical lines, for example) and passes these image points to the floor-location routine. The floor-location routine performs the necessary trigonometric calculations to locate the actual point.*

IV EXAMPLES

In this section we will illustrate the operation of the scene-analysis programs on two different television pictures. We remind the reader that the purpose of the programs is to extract certain types of information from the picture, not to convert the original picture into a perfect or complete line drawing.

*The details of the calculations are given in Hart and Duda (1969). The idea of using a point on the floor to establish location was called the support assumption by Roberts (1965).

The first scene to be processed is shown on the television monitor in Fig. 2(a). The digitized picture is shown in Fig. 2(b), where we have displayed the brightness in each picture cell as one of 16 levels of intensity. Figure 2(c) shows the gradient picture derived from the digitized picture; a point in this picture is displayed if the gradient at that point has a magnitude of at least two. Note that the wedge is partially out of the field of view, and that lack of contrast caused the back edge to be almost completely missed.

The scene analysis begins with the baseboard tracking procedure. Figure 3(a) shows the points in the picture selected by the tracking procedure as lying on the lower edge of the baseboard. Once the track is obtained, the line-fitting routine begins operation. Its first step is to fit, approximately, each segment of baseboard track by a straight line, as shown in Fig. 3(b). The longest of these segments is extended to the boundaries of the picture. In our example, this long line fails to "explain" two of the segments, so the vanishing point of the line is computed and the conjugate vanishing point found. The remaining segments are fitted as well as possible by a second long line constrained to pass through the conjugate vanishing point. The two long lines so determined, together with the track itself, are shown in Fig. 3(c). The final step is to project the fitted lines back onto the floor in order to determine the position of the walls. In general, of course, the final long line fits will differ from the initial rough segment fits. To illustrate this difference, Figs. 3(d) and 3(e) show two plan views of the robot's world. In each case the robot's position is at the midpoint of the bottom of the map. The cone of vision of the robot is delineated by the two lines

extending approximately northeast and northwest from the robot's position. In Fig. 3(d) we have shown the projected positions of the short segments, whereas in Fig. 3(e) we have superimposed the two long lines on the short segments. The two long lines meet at right angles, while the short segments do not. The projections of the long lines back onto the floor are accepted as the wall locations.

At this point the object-recognition program is called. Its first step is to find all the vertical lines in the picture. Figure 4(a) shows the first strong mask response encountered. As often occurs, there are breaks in the line. Three things can happen in such a case: a gap can be either bridged or not bridged, or a line segment can be deleted. As shown in Fig. 4(b), the bottom segment was deleted because it was very short, while the upper two segments were joined since no spurs were present to indicate a "true" endpoint and the gap itself was not very large. Fig. 4(b) also shows the mask response of the second vertical. In this case the gap was bridged, because the presence of the baseboard nearby indicated that the gap was probably the result of lack of contrast. Figure 4(c) shows the final result of the vertical line-finding operation. It constitutes the initial data for the decision-tree analyzer. The corresponding vague hypothesis is merely that there is at least one object in the picture and the vertical lines are associated with the object(s).

The first step in the decision tree is to examine the lower endpoint of the right-most vertical to find possible spurs. A confirming spur was in fact found, and an attempt was made to connect the lower endpoint to the lower endpoint of any other vertical. This failed, so as a last resort the spur was followed, as shown in Fig. 5(a). Spurs at this endpoint were

also found and followed, as shown in Figs. 5(b) and 5(c). The success of each of these tests added evidence that an object indeed existed in this area of the picture, and the angle of the final line indicated that the object was a wedge. At this point attention shifted to the other vertical. Neither the upper nor the lower endpoint gave any evidence of a spur, reducing the likelihood that the vertical line was "real." Moreover, when the lower endpoint was projected back onto the floor plane it fell behind the wall. For these reasons the program could not explain the line, and dismissed it as noise. Actually, the line was real--it was caused by light reflecting from a chrome strip that carpenters had used to cover a wall seam. Nevertheless, the scene analysis does extract most of the important information from the picture, as shown in the floor plan view of Fig. 6. The walls are correctly located, and the wedge correctly identified. The position of the wedge is found by projecting back the lower boundary [see Fig. 5(c)], and is accurate to within the resolution of the hardware. Thus, we would consider this particular analysis to be a successful one.

Our second scene is shown in the monitor view of Fig. 7(a). This scene is more complicated, involving two objects, both partially off camera, and a partially occluded open door. The digitized picture and gradient picture are shown in Fig. 7(b) and 7(c). The baseboard was found in a correct if unexcitingly routine fashion and we will say no more about that phase of the analysis. The first vertical mask response encountered is shown in Fig. 8(a). The lower short segments were eliminated because of length, and the single vertical line of Fig. 8(b) was accepted. In a similar fashion, all other verticals were found correctly, as shown in Fig. 8(c) and constituted the initial data for the decision tree. The

leftmost vertical line, shown in Fig. 9(a), was inspected for spurs at the lower endpoint as usual. A spur was found, and the search for a connection to a lower endpoint of another vertical resulted in the relatively poor fit and partial hypothesis shown in Fig. 9(b). The upper endpoints of both of these verticals are off the picture, and lower endpoints each yielded further spurs. For scenes without occlusion, this configuration is illegal; thus, attention shifted to the next vertical [Fig. 9(c)]. The same sequence of a partial hypotheses and tests resulted in the connection shown in Fig. 9(d). In this case, however, the upper endpoints were both on the picture, and a spur, when followed, connected them [Fig. 9(e)]. In particular, the followed spur did not form a triangle, providing further evidence that both verticals were "true." The left vertical yielded two more spurs, which, when followed, ran off the picture with no indication of a triangular face [Fig. 9(f)]. Thus, the object in question was accepted as a box (rectangular parallelepiped), rather than a wedge, even though the box was not completely within the picture. Moreover, the approximate location of the box was found from the lower boundary. At this point, attention centered on the next unexplored vertical. Coincidentally, this situation was the mirror image of the one just finished, and the same path through the decision tree was followed with the same results [Fig. 9(g)]. The final result of the analysis is shown in the floor-plan view of Fig. 10. The wall location is shown, and the two boxes are located conservatively--the unknown parts of their boundaries are not shown. The open doorway, which corresponded to the partial analysis of Fig. 9(b), was not found. Since the program was not designed to handle occlusion, this is about the best that could be expected. As an aside, we believe that it would not be

difficult to modify the program to handle such simple occlusions. Thus, although this analysis was not entirely successful, most of the important information in the picture was extracted.

The types of successes and failures that the scene analyzer exhibits in these examples are typical of its operation on most pictures taken in our environment. While the diversity of possible pictures makes it difficult to generalize, it seems fair to say that the analyzer will perform perfectly on only very simple scenes, but that it will extract some correct information from all but quite complicated scenes. Some failures can be fixed by incorporating in the analyzer an ever-increasing amount of knowledge about the world--for example, information about chrome strips, wall outlets, and so forth. Many other kinds of failures, however, can be fixed only by using more global information to direct each step of the processing.

V CONCLUSIONS

The research reported in this paper is part of a continuing effort. The next-generation scene analyzer that we construct will very likely combine some of the ideas reported here with the approach described by our co-workers Brice and Fennema (1969). In our opinion, the most noteworthy aspect of the present work is that it has resulted in the realization of a scene analyzer that evaluates evidence from a picture in the light of both other evidence and prior information about the environment. To illustrate the distinction between the approach reported here and previous work (including some of our own), suppose the object-recognition program attempted to find all the straight lines in the picture by means of an algorithm that knew only about straight lines. The

danger is not merely that such an approach is likely to be inefficient, but that the algorithm is likely to find many spurious lines and fail to find many valid ones. Sorting out the wheat from the chaff may be (and for our environment, in fact, was), a hopeless task. By way of contrast, the object-recognition program described here applies only an initial vertical line-finding routine in an "uninformed" way--that is, without any checking of whether the results it obtains make sense. This operation is performed merely because the algorithm must start somewhere, and vertical lines are generally found reliably. All other operations proceed in small steps, and at each step there is a check of the result of a given test with previous results. This procedure was continued not until a perfect line drawing line was obtained, but only until enough evidence was gathered to make a reasonably confident decision. Note that in the examples the analyzer did not complete any drawings. We believe that perfect line drawings should be constructed after, not before, the scene has been analyzed.

One interesting comment about our approach is that we have not yet been able to draw any firm conclusions about the use of confidence measures to control the tree search. The idea remains an appealing one, but in most of our experiments any failure of the object-recognition program was sufficiently catastrophic to make error-recovery through backup unlikely. It may be that the tree is not sufficiently elaborated--that not enough tests are performed to check the results of previous tests. The resolution of the question will have to await further experiments.

Another comment about the object-recognition program is that

much of its effort is devoted to identifying parts of objects in the scene. In other words, it is devoted to solving the so-called figure-ground problem. The baseboard-tracking program solves this problem in a trivial way--the baseboard is almost always darker than its immediate background. In more complicated situations we are in the uncomfortable position of not being able to extract the figure from the background unless we can first recognize the figure. The situation is complicated even more by the fact that any object might be partially out of the field of view. Thus, the difficulty for the object-analysis program is not so much a function of the number of object types (within reasonable bounds), as is it a function of the background or general environment. If the only allowable objects were boxes, but the environment remained as it is, we anticipate that the object-recognition program would not be very much more reliable than it is now.

The baseboard program provides an interesting contrast to the object-recognition program. The single most important simplification is undoubtedly the fact that the baseboard can be easily extracted from the background by virtue of its dark color. A second simplifying factor is its linearity--each column of the digital picture is traversed by at most a single baseboard. These simplifying considerations, together with the fact that every picture contains either no visible baseboard, a single straight baseboard, or two baseboards meeting at right angles, enable us to incorporate in an algorithm much of the prior information that a human might use. Indeed, most human observers were not significantly better at choosing straight-line fits to the tracking data than was the line-fitting program. The key difference, of course, is that a human analyzing a pic-

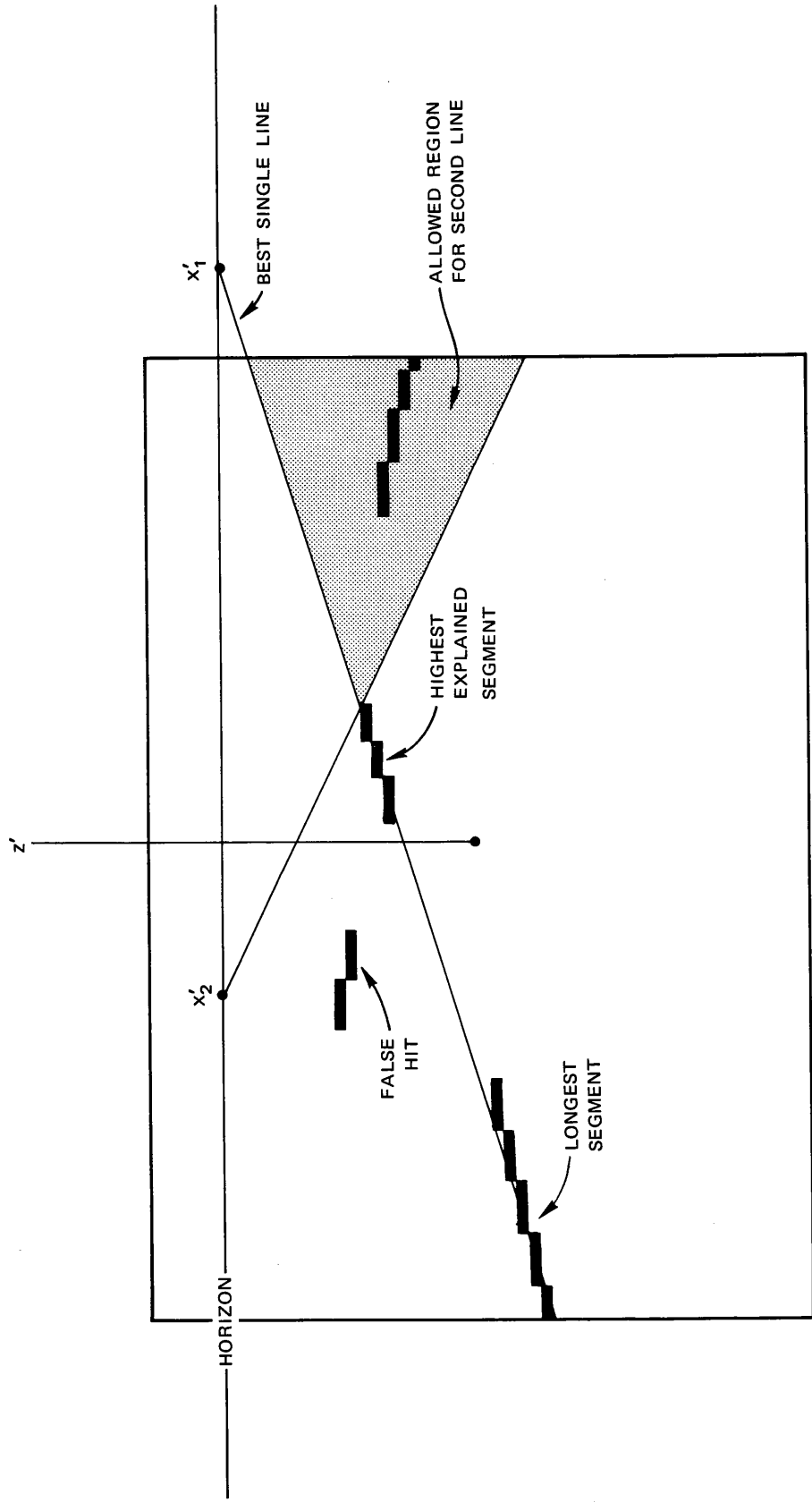
ture to locate the walls does not restrict his attention solely to the baseboard. He makes use of many other clues as well. One of the weaknesses of the programs described is that they do not interact with each other in any important way. The baseboard tracker, for example, should provide information to the object recognizer to help it extract a figure from the background. The trend in scene analysis, we suspect, will be toward this sort of consolidation--toward programs that incorporate more comprehensive knowledge of the visual environment, and that use this knowledge more extensively in their operation.

ACKNOWLEDGMENTS

The work reported in this paper was the result of the efforts of several people. We would like to thank Helen Chan, William Duvall, and Stephen Weyl, whose programming contributions were essential to the realization of the scene analyzer. We would also like to thank Claude Brice for his important work in designing the logic of the decision tree.

REFERENCES

- Ahuja, D. B., and S. A. Coons, "Geometry for Construction and Display," IBM Systems Journal, Vol. 7, Nos. 3 and 4, pp. 188-205 (1968).
- Brice, C. R., and C. L. Fennema, "Scene Analysis of Pictures Using Regions," Technical Note No. 17, Artificial Intelligence Group, Stanford Research Institute (November 1969).
- Forsen, G. E., "Processing Visual Data with an Automaton Eye," in Pictorial Pattern Recognition, pp. 471-502, G. C. Cheng et al., Eds. (Thompson Book Company, Washington, D.C., 1968).
- Guzman, A., "Decomposition of a Visual Scene into Three-Dimensional Bodies," Proc. FJCC, pp. 291-304 (December 1968).
- Hart, P. E., "Searching Probabilistic Decision Trees," Technical Note No. 2, Artificial Intelligence Group, Stanford Research Institute (February 1969).
- Hart, P. E., and R. O. Duda, "Perspective Transformations," Technical Note No. 3, Artificial Intelligence Group, Stanford Research Institute (February 1969).
- Munson, J. H., "The SRI Intelligent Automaton Program," Proc. First Nat. Symp. on Indust. Robots (April 1970).
- Nagy, G., "State of the Art in Pattern Recognition," Proc. IEEE, Vol. 56, pp. 836,862 (May 1968).
- Nilsson, N. J., "A Mobile Automaton: An Application of Artificial Intelligence Techniques," Proc. Int. Joint Conf. on Art. Int., pp. 509-520 (May 1969).
- Pingle, K. K., and W. M. Wichman, "Computer Control of a Mechanical Arm Through Visual Input," IFIP Congress 68, Applications 3, Booklet H, pp. H140-H146 (Edinburgh, August 1968).
- Roberts, L. G., "Machine Perception of Three-Dimensional Solids," in Optical and Electro-Optical Information Processing, pp. 159-197, J. T. Tippett et al., Eds. (MIT Press, Cambridge, Massachusetts, 1965).



TA-7494-40

FIGURE 1 GEOMETRY FOR BASEBOARD FITTING

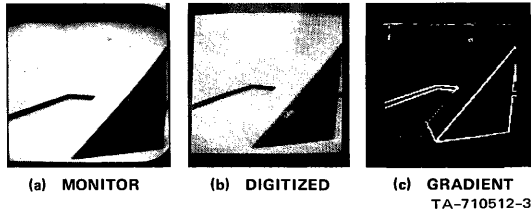


FIGURE 2 A SCENE CONTAINING A WEDGE

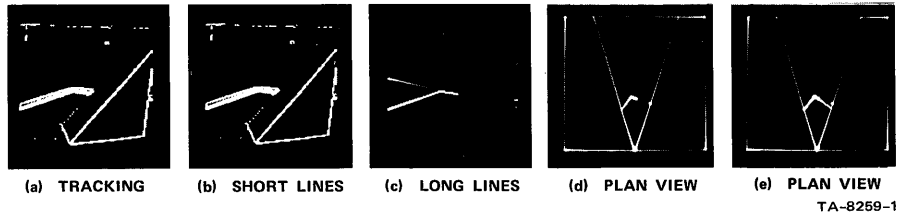


FIGURE 3 BASEBOARD TRACKING AND FITTING

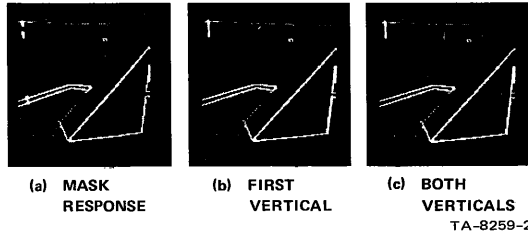


FIGURE 4 VERTICAL LINE FINDING

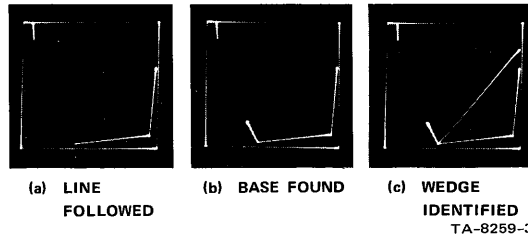


FIGURE 5 OBJECT IDENTIFICATION

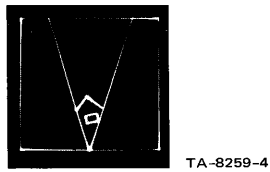
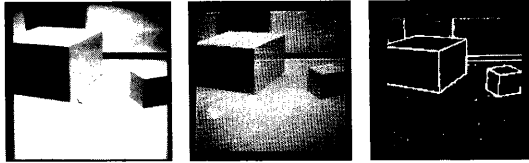
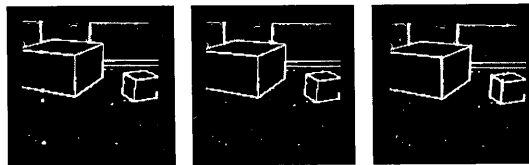


FIGURE 6 FINAL PLAN VIEW



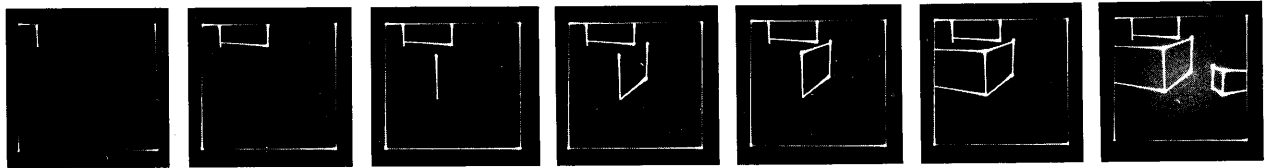
(a) MONITOR (b) DIGITIZED (c) GRADIENT
TA-8259-5

FIGURE 7 A SCENE WITH OCCLUSION



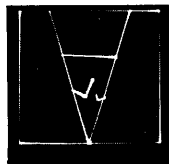
(a) MASK RESPONSE (b) FIRST VERTICAL (c) ALL VERTICALS
TA-8259-6

FIGURE 8 VERTICAL LINE FINDING



(a) FIRST VERTICAL (b) DOOR: REJECTED (c) NEXT VERTICAL (d) VERTICALS CONNECTED (e) SIDE FOUND (f) BOX FOUND (g) BOTH BOXES FOUND
TA-8259-7

FIGURE 9 STEPS IN SCENE ANALYSIS



TA-8259-8

FIGURE 10 FINAL PLAN VIEW