

October 1970

SOME CURRENT TECHNIQUES FOR SCENE ANALYSIS

by

Richard O. Duda

Artificial Intelligence Group

Technical Note 46

SRI Project 8259

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

I INTRODUCTION

The purpose of the visual system is to provide the automaton with important information about its environment, information about the location and identity of walls, doorways, and various objects of interest. By adding new information to the model, the visual system gives the automaton a more complete and accurate representation of its world. The role of vision is not independent of the state of the model. If the automaton has entered a previously unexplored area, the visual scene must be analyzed to add information about the new part of the environment to the model. In this situation, the model can provide so little assistance that it is often not referenced at all. On the other hand, if the automaton is in a thoroughly known area, the role of vision changes to one of providing visual feedback to correct small errors and verify that nothing unexpected has happened. In this situation, the model plays a much more important role in assisting and actually guiding the analysis.

Until recently our attention has been directed primarily at the general scene-analysis problem. Every picture was viewed as a totally new scene exposing completely unknown area. More recently we have addressed the problem of using a complete, prespecified map of the floor area to update the automaton's position and help in tasks such as going through a doorway. Another use of this kind of visual feedback would be the monitoring of objects being pushed.

In trying to solve these problems, we have tended to take one or the other of two extreme approaches. Either we tried to develop general methods that can cope with any possible situation in the automaton's

world, or we tried to exploit rather special facts that allow an efficient special-purpose solution. The first approach involves the more interesting problems in artificial intelligence, but it provides more capabilities than are needed in many situations, and provides them at the cost of relatively long computation times. The second approach provides fast and effective solutions when certain (usually implicit) preconditions are satisfied, though it can fail badly if these conditions are not met. Eventually, of course, some combination of these two approaches will be needed, since the automaton actually operates in a partially known world, rather than one that is completely unknown or completely known. However, we have decided to concentrate on these two extreme situations before addressing the intermediate case. The remainder of this note describes the current status of our work in these areas.*

II REGION ANALYSIS

A. The Merging Procedure

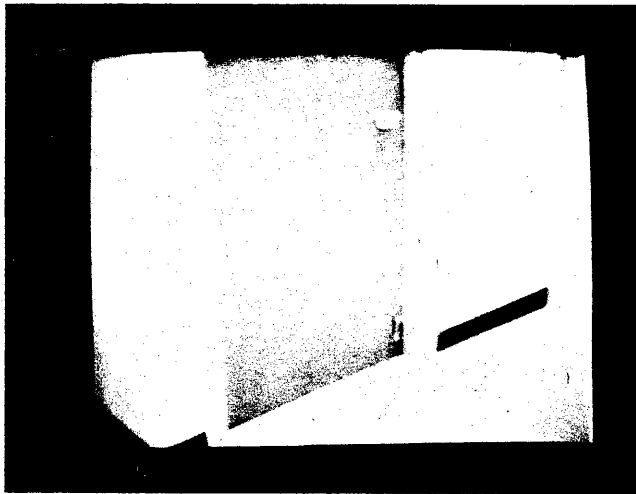
Our work in general scene analysis is based on dividing the picture into regions representing walls, floors, faces of objects, etc. The basic approach has been described in detail elsewhere,³ and only a brief summary will be given here. The procedure begins by partitioning the digitized image into elementary regions of constant brightness. This usually produces many small, irregularly shaped regions that are fragments of more meaningful regions. Two heuristics are used to merge

* Our earlier work in scene analysis is described in Reference 1. Additional information on more recent work is contained in References 2-5. References are listed at the end of this report.

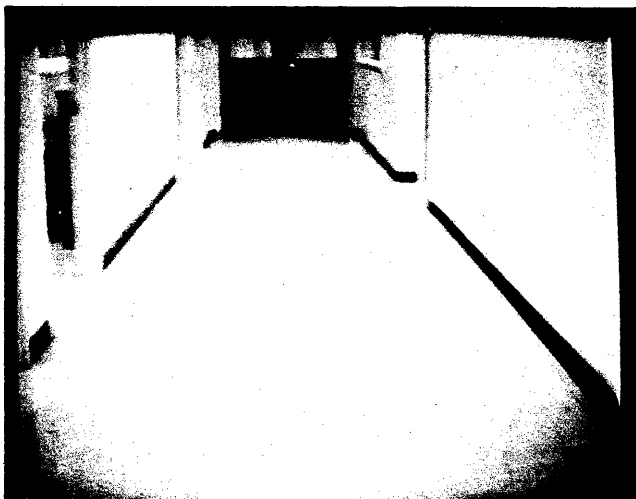
these smaller regions together. Both of these heuristics operate on the basis of fairly local information, the difference in brightness along the common boundary between two neighboring regions. The heuristics are not infallible; they can merge regions that should have been kept distinct, and they can fail to merge regions that should have been merged. However, they reduce the picture to a small number of large regions corresponding to major parts of the picture, together with a larger number of very small regions that can usually be ignored.

The effect of applying these heuristics is best described through the use of examples. Figure 1 shows television monitor views of three typical corridor scenes. Figure 2 shows the results of applying the merging heuristics to digitized versions of these pictures. The boundaries of the regions in these pictures are directed contours, and can be traced using the correspondences shown in Table I. Generally speaking, important regions can be separated from unimportant regions purely on the basis of size. Figure 2a, for example, contains four large, important regions. Three of them are directly meaningful (the door, the wall to the right, and the baseboard), and the fourth is the union of two important regions (the floor and the wall to the left). An inspection of Figure 2b shows similar results. Figure 2c shows the result of applying the technique to a complicated scene; while some useful information can be obtained, the resolution available severely limits the usefulness of the results.

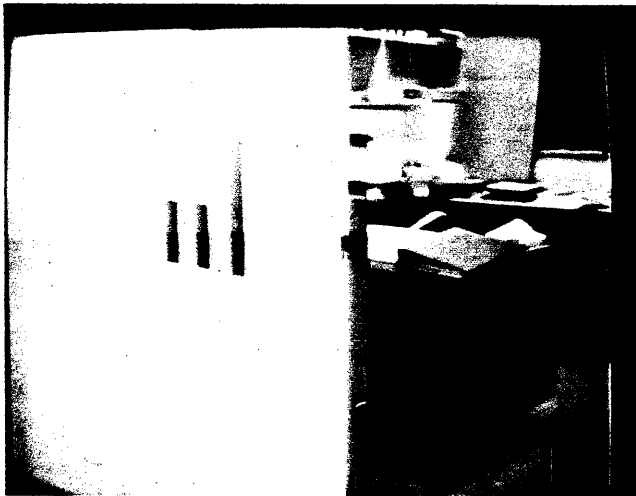
Our only complete scene-analysis program is oriented toward identifying boxes and wedges, objects with triangular or rectangular



(a) DOOR



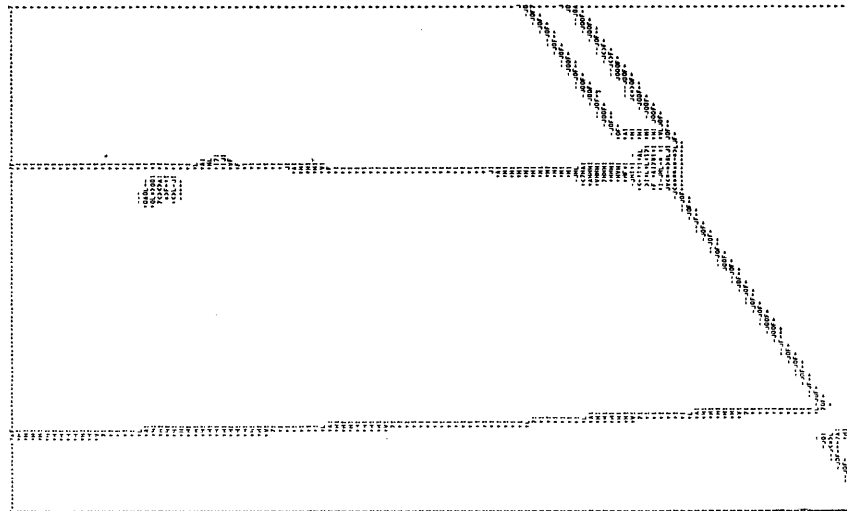
(b) HALL



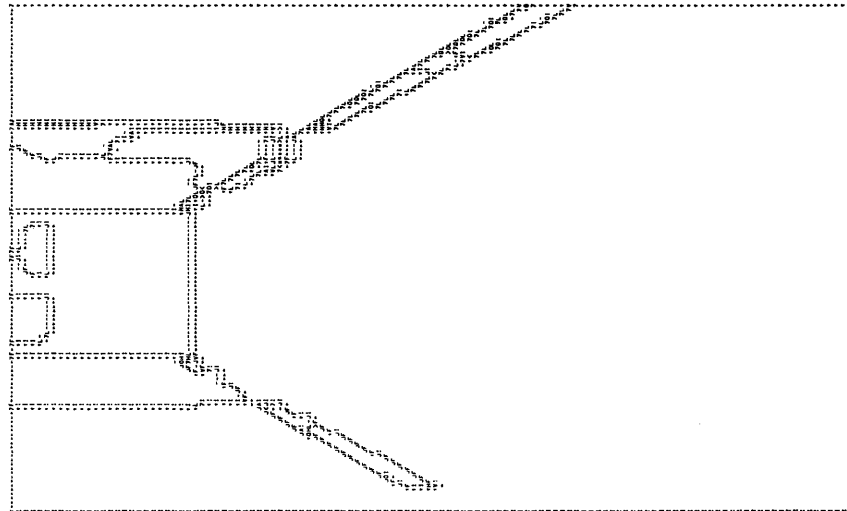
(c) OFFICE WITH SIGN

TA-8259-20

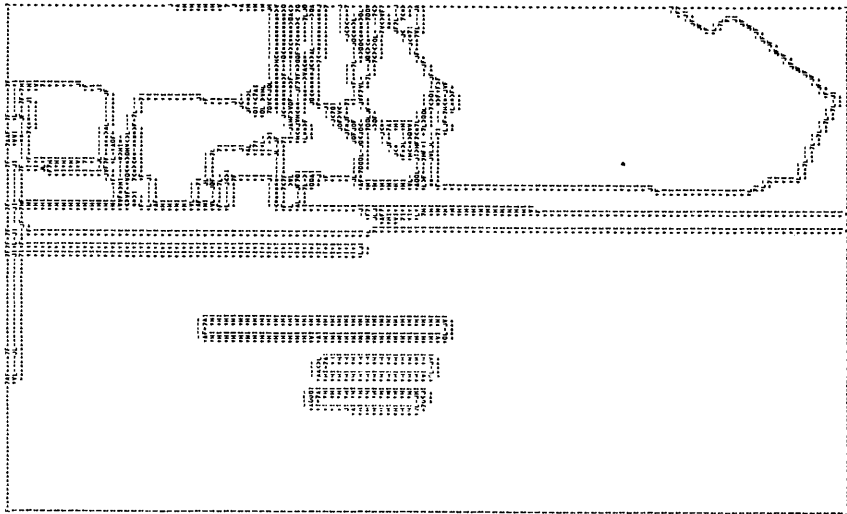
FIGURE 1 THREE CORRIDOR SCENES



(a) DOOR



















(b) HALL



(c) OFFICE WITH SIGN
TA-8259-21

FIGURE 2 RESULTS OF MERGING HEURISTICS

TABLE I CORRESPONDENCE BETWEEN
BOUNDARY SEGMENT CONFIGURATIONS
AND CHARACTERS USED IN PRINTOUT

CONFIGURATION	CHARACTER	CONFIGURATION	CHARACTER
	I		l
	—		L
	↑		H
	J		V
	←		F
	=		∇
	Z		A
	>		O

faces, in a simple room environment.³ For this task, we begin by fitting the boundaries of the major regions by straight lines. Regions are identified as being part of the floor, walls, baseboards, and faces of objects by such properties as shape, brightness, and position in the picture. Objects are identified by grouping neighboring faces satisfying some of the simpler criteria used by Guzman.⁶ In the process, certain errors caused by incorrect merging are detected and corrected. We have yet to complete a similar analysis program for the conditions encountered in corridor scenes. However, we have investigated the problem of obtaining a scene description that is internally consistent; the next section describes the analysis approach for this problem.

B. A Procedure for Scene Analysis

If we assume temporarily that the merging heuristics have succeeded in the sense that all of the large regions are meaningful areas, then the only basic problem remaining is the proper identification of each region. Examination of the corridor pictures indicates the need to be able to identify a number of different region types, including the following:

- (1) Floor
- (2) Wall
- (3) Door
- (4) Door jamb
- (5) Object face
- (6) Baseboard
- (7) Baseboard reflection
- (8) Sign*
- (9) Window

*By "sign" we mean a dark vertical bar on the wall used, as illustrated in Figure 1c, to identify an office.

- (10) Clock
- (11) Doorknob
- (12) Thermostat
- (13) Power outlet
- (14) Automaton.

Each of these regions has certain properties which tend to characterize it uniquely. For example, the floor region is usually large, bright, and near the bottom of the picture. However, most regions can be identified with greater confidence if the nature of their neighbors is considered as well. Thus, the presence of a baseboard or baseboard reflection at the top of a region almost guarantees that the region is the floor; conversely, the presence of wall area immediately above a region guarantees that it can not be a baseboard reflection. If regions are identified without regard to how that choice affects the overall scene description, the chance for error is increased. Moreover, the resulting description can be nonsensical.

Many, though by no means all, of the relations between types of regions relate to neighboring regions. Table II indicates those types of regions that can and cannot be legal neighbors. We can easily add to this further restrictions, such as the fact that the baseboard must have the wall as a neighbor along its top edge. These are some of the important known facts about the general nature of the automaton's environment. The problem is to use facts such as these to aid in the analysis of the scene.

One approach to solving this problem is to use these facts as constraints to eliminate impossible choices. Suppose that each significantly large region in the picture is tentatively classified

TABLE II. REGIONS THAT ARE LEGAL NEIGHBORS

	FLOOR	WALL	DOOR	DOOR JAMB	OBJECT FACE	BASEBOARD	BASEBOARD REFLECTION	SIGN	WINDOW	CLOCK	DOORKNOB	THERMOSTAT	POWER OUTLET	AUTOMATON
FLOOR		+	+	+	+	+	+							
WALL	+	+	+	+	+	+		+	+	+	+	+	+	+
DOOR	+	+		+	+	+			+		+			+
DOOR JAMB	+	+	+		+	+					+			+
OBJECT FACE	+	+	+	+	+	+	+	+	+		+	+	+	+
BASEBOARD	+	+	+	+	+	+	+						+	+
BASEBOARD REFLECTION	+				+	+	+							+
SIGN		+			+									+
WINDOW		+	+		+									
CLOCK		+												
DOORKNOB		+	+	+	+									
THERMOSTAT		+			+									
POWER OUTLET		+			+	+								+
AUTOMATON	+	+	+	+	+	+	+	+					+	

TA-8259-25

on the basis of the attributes of that region alone. Suppose further that a score is computed for each region that measures the degree to which it resembles each region type.* For any selection of names for regions, we can define the score for the resulting description as the sum of the individual scores. Then, we can analyze the scene by trying to find highest scoring legal selection of region names. With no loss in generality and some gain in convenience, we can work with the losses incurred by selecting other than the highest scoring choice. In terms of losses, we want the legal description having the smallest overall loss.

This problem is basically a tree-searching problem. The start node of the tree corresponds to the first region selected for naming. The branches emanating from that node correspond to the possible choices of names for that region. A path through the tree corresponds to a unique labeling of the picture. Thus, if there are N possible region names and R regions, there are potentially N^R possible paths through the tree. Each path passes through $R+1$ nodes from the start node to the terminal node. Every terminal node has a loss value, which is the sum of the losses incurred for the choices along the path to that node. A goal node is a terminal node corresponding to a complete, legal scene description. We seek the goal node with the smallest overall loss.

This is a standard problem in tree searching, and optimum search procedures are known. Assume that some choices have been made for some of the regions so that we have a partially expanded tree.

* This score might be interpreted as the logarithm of the probability that the given region is of the indicated type.

Using the Hart-Nilsson-Raphael terminology,⁷ some of the terminal nodes of this tree are open nodes, candidates for further expansion. Each open node has an associated loss \hat{g} , the sum of the losses from the start node to that node. If we assume that there is no reason to believe that zero-loss choices cannot be made from that node on, then the optimal search strategy is to expand that open node having the minimum \hat{g} .

To expand a node, we must select a region not previously considered and examine the possible choice for that region, ruling out any choices that are not legal. Different strategies can be used for selecting the next region. It seems advantageous to ask it to be a neighbor of the regions selected previously, since this maximizes the chance of detecting illegalities. In general, we will have several neighbors for candidate successors. Of these, it seems reasonable to select the one having the highest score, under the assumption that the first choice name for this region is most likely to be correct.

After a region has been selected, it is necessary to examine the choices one can make for its name to see which ones are legal. If we limit ourselves to pairwise relations between neighboring regions, we need merely compare each choice with previously made choices on the path to this point and test each for legality.* The node expanded is removed from the list of open nodes, the resulting new nodes are added, and the process is repeated until the algorithm selects a goal node for further expansion. This is our final result, a legal scene description having the minimum loss.

* When an illegality is found, that choice is deleted. One can argue that few relations are so strong as to be absolutely illegal, and an alternative approach would be to introduce various additional losses for the different observed relations.

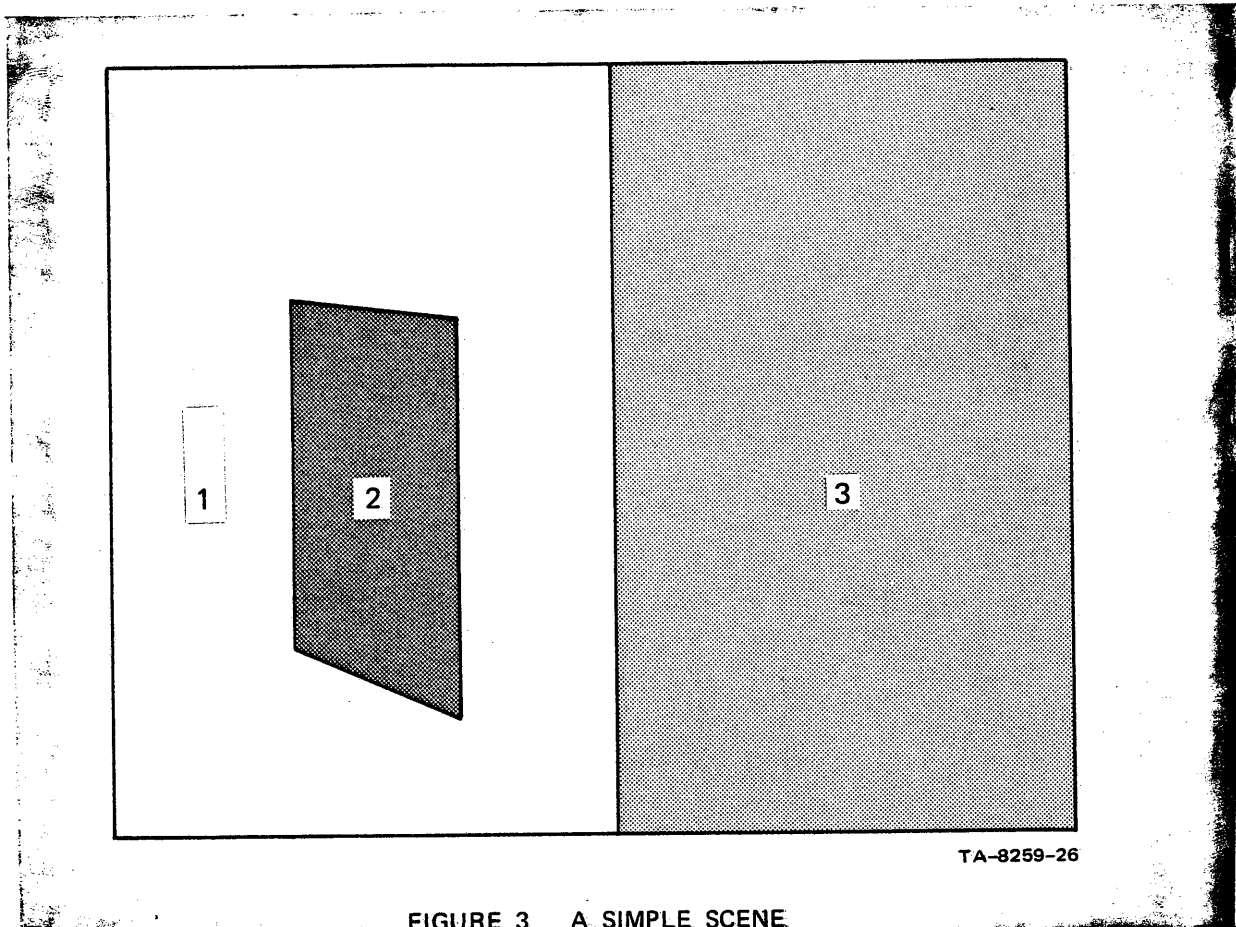
C. Examples

The following examples serve to illustrate the action of this scene-analysis procedure. Consider first the simple scene shown in Figure 3. For simplicity, we assume that there are only five types of allowed regions--floor, wall, door, baseboard, and sign. Consider Region 1. On the basis of its brightness, size, vertical right boundary, and possession of a hole, it should receive a high score as a wall, and lower scores as floor, door, sign, and baseboard. Region 2 might, perhaps, score highest as a door, and so on. Thus, the following table of scores, although purely imaginary, is not unreasonable. Missing entries correspond to scores too low to be seriously considered.

Region \ Type	Floor	Wall	Door	Base-board	Sign
1	5	6	2		
2			7	1	5
3	3	3	5		1

The following table gives equivalent information in terms of the losses associated with each choice.

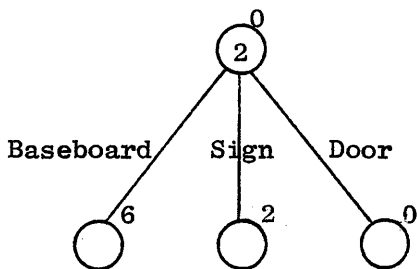
Region \ Type	Floor	Wall	Door	Base-board	Sign	Max Score
1	1	0	4			6
2			0	6	2	7
3	2	2	0		4	5



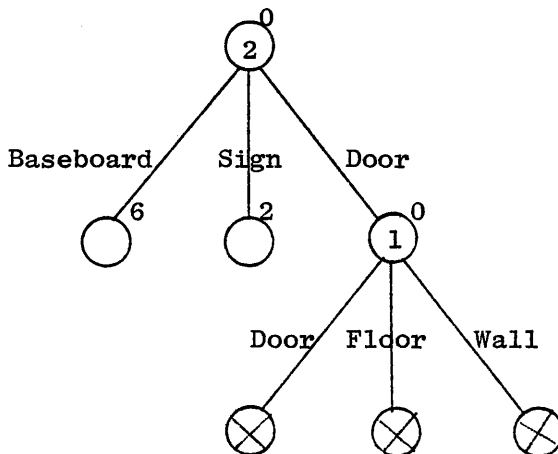
TA-8259-26

FIGURE 3 A SIMPLE SCENE

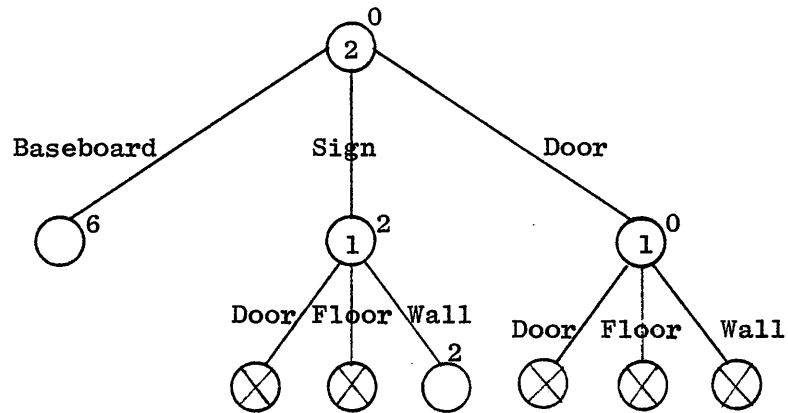
Let us use our tree-searching algorithm to obtain the minimum-loss, legal description of this scene. Initially the successor function is unconstrained by neighbor restrictions, and selects Region 2 merely because it has the highest score. At this point, all of the choices for Region 2 are legal, and the tree has three open nodes; the numbers shown next to each node give the loss accumulated in reaching that part of the tree.



The search algorithm requires that the open node having the least loss be expanded next, which corresponds to tentatively calling Region 2 a door. The successor function finds only one neighbor to choose from, Region 1, and considers its alternatives: wall, floor, and door. None of these choices is a legal neighbor surrounding Region 1, and hence all are rejected. Thus, this open node has no successors.

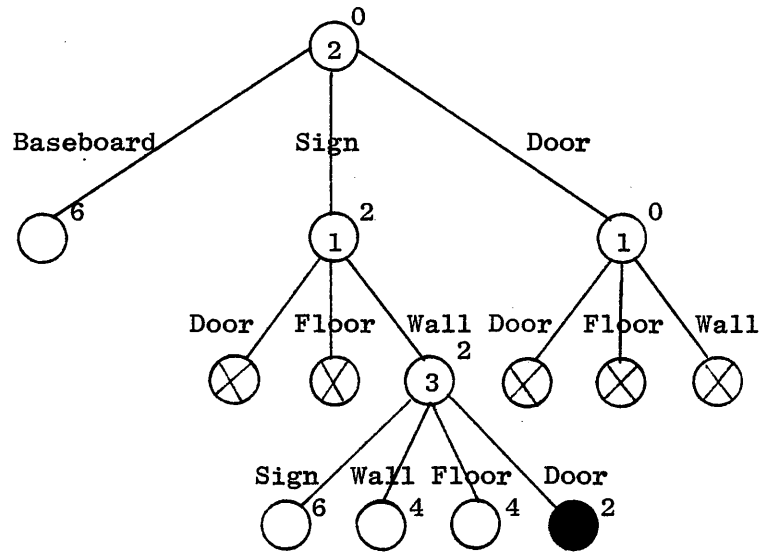


Returning to the choices for open nodes, Region 2 is tentatively called a sign. The successor function again selects Region 1, and this time finds one legal successor, the wall.* The loss associated with this choice is 0, and the overall loss is 2. The list of open nodes still contains two members.



The search algorithm selects the open node with loss 2, and the successor function has only Region 3 to select from. All of the choices for Region 3 are all legal with respect to calling Region 2 a sign and Region 1 a wall. The least loss results from calling Region 3 a door, and the scene analysis is completed.

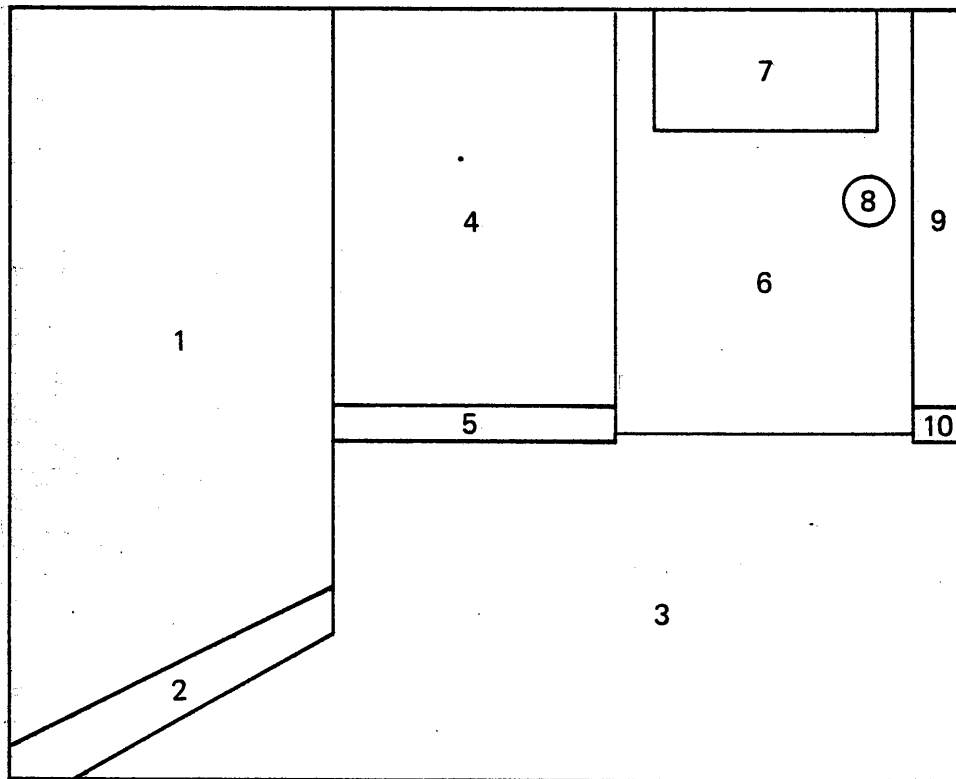
* Note that our successor function will always produce a tree with R+1 levels. At any level, the same region will always be selected by the successor function. The actual successors, however, will be limited by the legality requirement.



A somewhat more realistic example involving 10 regions and 14 region types is illustrated in Figure 4. Table III gives the hypothetical scores. Based on these scores alone, half of the regions would be incorrectly identified. Figure 5 shows the tree produced by the search algorithm. The development of this tree is too complicated to describe in detail. It should be noted, however, that considerable backtracking occurred because a low-scoring third choice was needed for Region 8, the doorknob. Whether or not this can be circumvented without causing other problems is not known.

D. Remarks

To date, this procedure has only been used on some hypothetical examples. We have modified a general tree-searching program to adapt it to some special characteristics of this problem. However, we have not started the important task of writing programs to measure characteristics of regions and to use these characteristics to produce recognition scores.



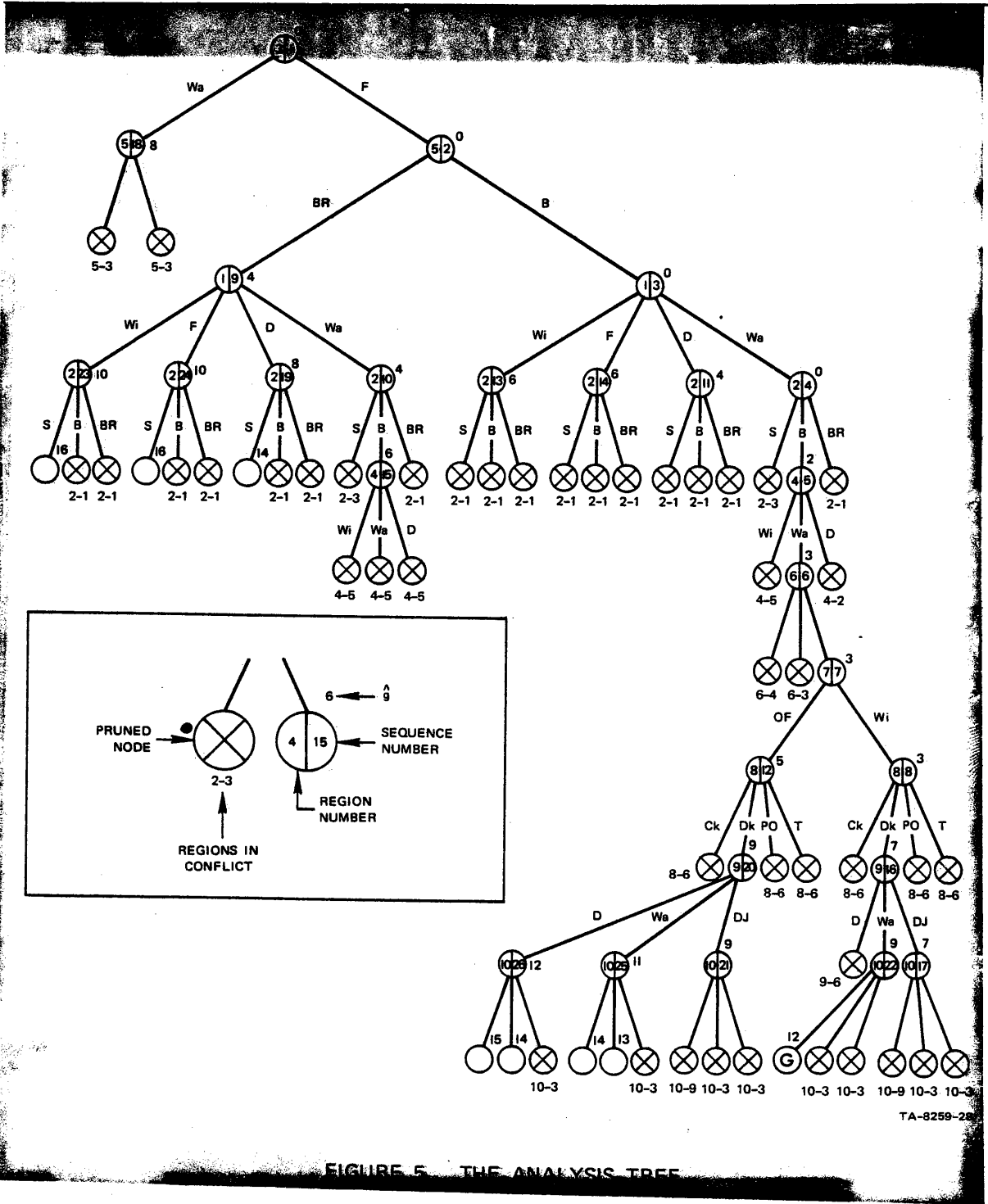
TA-8259-27

FIGURE 4 A MORE COMPLICATED SCENE

TABLE III HYPOTHETICAL REGION SCORES

TYPE	REGION									
	1	2	3	4	5	6	7	8	9	10
FLOOR	1		11			2				
WALL	7		3	5		5			4	
DOOR	3			6		6			3	
DOOR JAMB									6	
OBJECT FACE							6			
BASEBOARD		5			9					3
BASEBOARD REFLECTION		7			5					
SIGN		1								6
WINDOW	1			2			8			
CLOCK								1		
DOORKNOB								2		
THERMOSTAT								6		
POWER OUTLET								3		4
AUTOMATON										

TA-8259-29



TA-3259-28

FIGURE 5. THE ANALYSIS TREE

In addition, we have not implemented any legality conditions beyond the simple conditions given in Table II.

This approach to scene analysis has several potential advantages. It is not necessary to identify every region correctly at the outset to obtain a correct analysis, provided that the "syntactic" rules are sufficiently complete. By providing a limit on the allowable loss, a partial scene description can be obtained that may be useful even though incomplete. Perhaps most important, the operations of merging, feature extraction, classification, and analysis are clearly separated, allowing fairly independent modification and improvement. In particular, the general knowledge about the environment can be expressed explicitly as rules for legal scenes, and if the environment is changed it is possible to confine the program changes to modifying these rules.

One of the major problems with this approach is the lack of an obvious way to detect erroneous regions, regions that are fragments of or combinations of meaningful regions. We are currently working on this problem, since progress toward its solution is needed before implementation of this system can be begun. Another problem is that it is not clear how specific information contained in the model can be used to guide the analysis. This problem of working in a world that is neither completely known nor completely unknown is one of the major unsolved problems in visual scene analysis.

III LANDMARK IDENTIFICATION

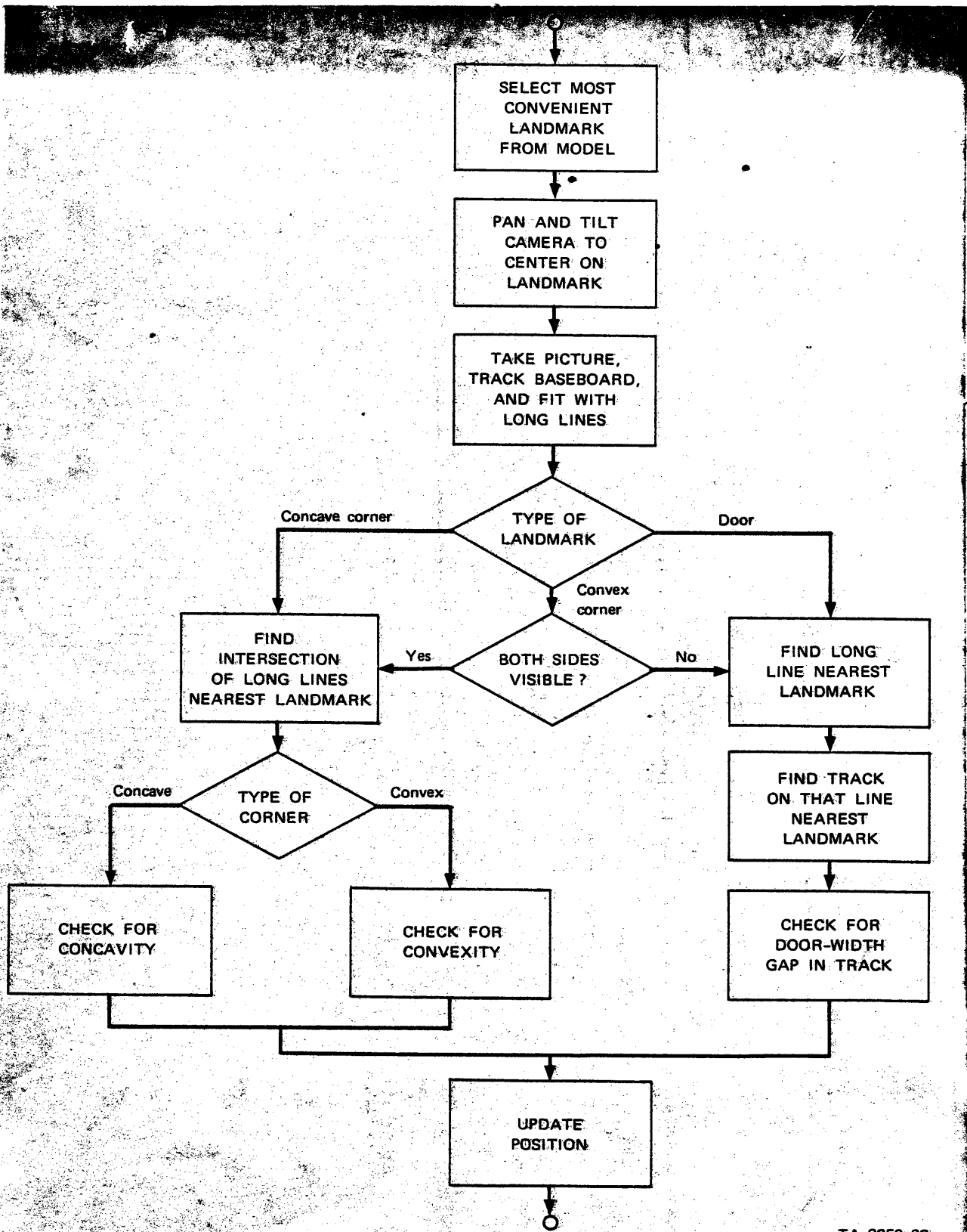
When the environment is completely known, the visual system can provide feedback to update the automaton's position and orientation.

The x-y location of the automaton and its orientation θ can be determined uniquely from a picture of a known point and line lying in the floor.* Such distinguished points and lines serve as landmarks for the automaton. This section describes our present program that uses concave corners, convex corners, and doorways as landmarks to update position and orientation.

A flowchart outlining the basic operations of this program is shown in Figure 6. The program begins by selecting a landmark from the model that should be visible from the automaton's present position; if more than one candidate exists, one is selected on the basis of range and the amount of panning of the camera required.* The camera is then panned and tilted the amount needed to bring the landmark into the center of the field of view, and a picture is taken. The baseboard-tracking routine described previously² is used to find the segments of baseboard in the picture and to fit them with long straight lines.

Exactly what happens next depends on the landmark type. For a door, the long line nearest the center of the picture is selected, and the true image of the landmark is assumed to be the endpoint of the baseboard segment on that line and nearest the center of the picture. An additional check is made to see that the gap from that point to the next segment is long enough to be a passageway. A convex corner viewed from an angle such that only one side is visible is treated as if it were a door. Otherwise, the intersection of long lines nearest the center

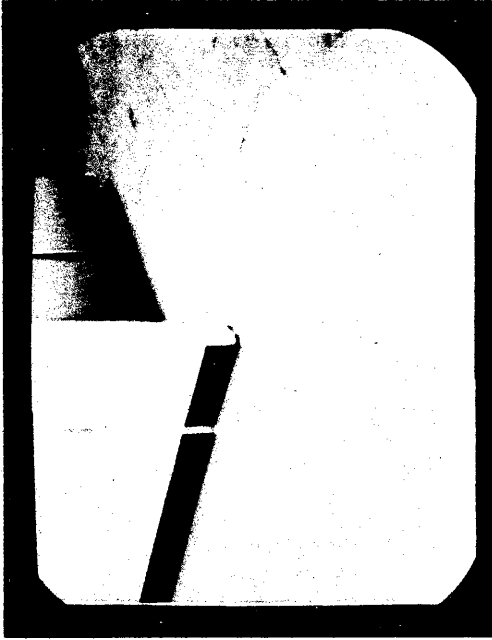
* If no landmark is in view, a suitable message is returned together with a suggested vantage point from which a landmark can be seen. This is one of several "error" returns that can be obtained from the program. The program can also be asked to select a specific landmark, or a landmark different from the ones previously selected.



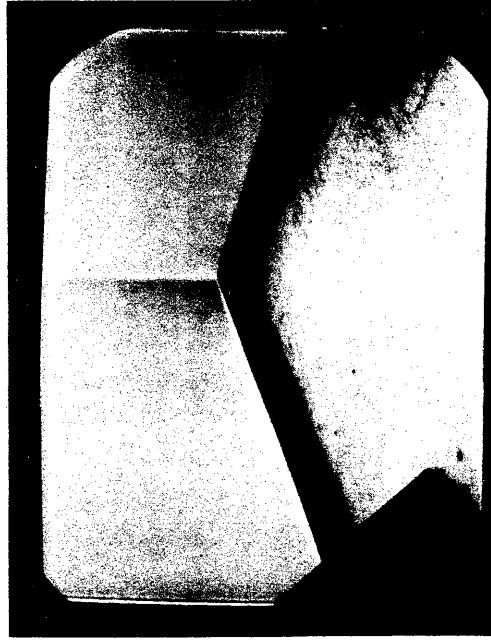
TA-8259-22

of the picture is assumed to be the true image of the landmark, and a check is made to see that the baseboard segments near this point have the right geometrical configuration. The location of the landmark in the picture gives the information needed to compute corrections for the automaton's position and orientation.

The operation of this program is illustrated in Figure 7. In this experiment, the automaton was approximately 7.5 feet away from a wall along which there were four landmarks, both sides of a doorway, a convex corner, and a concave corner. The pictures in Figure 7 show how closely the panning and tilting brought the landmarks to the center of the pictures. For scenes as clear as these, the program operates very reliably. Presently, we can use this routine to locate the robot with an accuracy of between 5 percent and 10 percent of the range, and to fix its orientation to within 5 degrees. Since the errors are random, the accuracy can be improved further by sighting a second landmark. Further increases in accuracy, if needed, will have to be obtained by improving the tilt and pan mechanism for the camera.

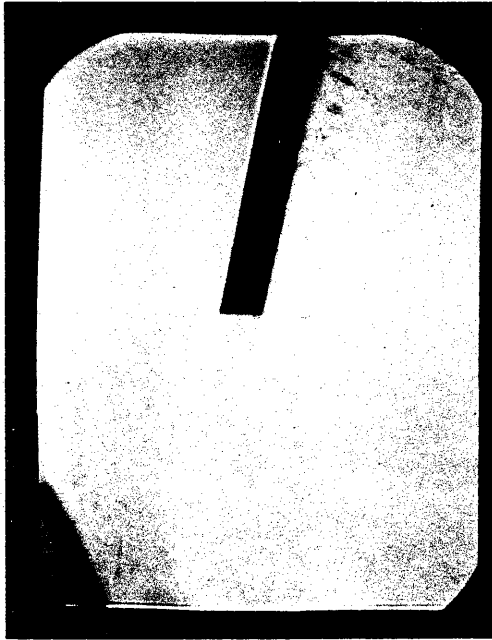


(b) LEFT DOOR

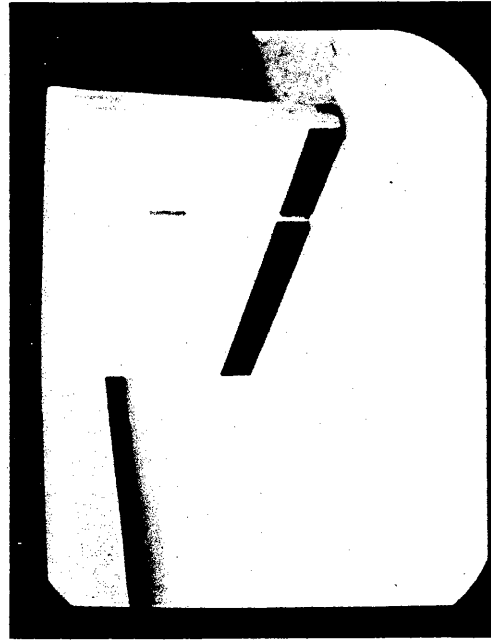


(d) CONCAVE CORNER

TA-8259-23



(a) RIGHT DOOR



(c) CONVEX CORNER

FIGURE 7 LANDMARKS

REFERENCES

1. L. S. Coles et al., "Applications of Intelligent Automata to Reconnaissance," Final Report, Contract F30602-69-C-0056, SRI Project 7494, Stanford Research Institute, Menlo Park, California (November 1969).
2. L. J. Chaitin et al., "Research and Applications--Artificial Intelligence," Interim Scientific Report, Contract NAS12-2221, SRI Project 8259, Stanford Research Institute, Menlo Park, California (April 1970).
3. C. R. Brice and C. L. Fennema, "Scene Analysis Using Regions," SRI Artificial Intelligence Group Technical Note 17, Stanford Research Institute, Menlo Park, California (April 1970).
4. R. O. Duda and P. E. Hart, "Experiments in Scene Analysis," SRI Artificial Intelligence Group Technical Note 20, Stanford Research Institute, Menlo Park, California (January 1970).
5. R. O. Duda and P. E. Hart, "A Generalized Hough Transformation for Detecting Lines in Pictures," SRI Artificial Intelligence Group Technical Note 36, Stanford Research Institute, Menlo Park, California (July 1970).
6. A. Guzman, "Decomposition of a Visual Scene Into Three-Dimensional Bodies," Proc. FJCC, pp. 291-304 (December 1968).
7. P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Sys. Sci. Cyb., Vol. SSC-4, pp. 100-107 (July 1968).
8. P. E. Hart and R. O. Duda, "Perspective Transformations," SRI Artificial Intelligence Group Technical Note 3, Stanford Research Institute, Menlo Park, California (February 1969).