A MACHINE VISION FOR COMPLEX INDUSTRIAL PARTS WITH LEARNING CAPABILITY

Masahiko Yachlda and Saburo Tsuji
Department of Control Engineering
Osaka University
Toyonaka, Osaka, Japan

Abstract

This paper describes a versatile machine vision system that can recognize a variety of complex industrial parts based on the previously learned models of them. It proposes a model guided approach for recognition in which models of objects direct recognition process by suggesting the features to be examined next and their predicted locations. For the system to be readily applied to new tasks, it can automatically generate the models of objects while a human operator showing example parts and teaching important features of them interactively on displays. The system has been applied to various sets *of* parts *of* small industrial gasoline engines and the result was satisfactory.

Introduction

A machine vision for simple objects has been studied by many people and is already performing a variety of simple tasks in laboratory settings (1,2,3). To apply these systems for real industrial uses of such as assembly and inspection, a machine vision system that can recognize complex real objects should be developed.

Recognition of real industrial parts is a new field. Ejiri et al. proposed a vision system which could be used to classify simple industrial parts(A). Olsztyn et al. made a good experimental system which located studs on hubs and stud holes in wheels and mounted wheels onto automobile hubs (5). Although it demonstrated the feasibility and usefulness of the machine vision for an industrial application, the system was restricted to the specific task and extensive reprogramming was necessary to perform other tasks. Recently, as a related project, Chien et al. are studying on automatic inspection of hybrid circuits(6).

To give a machine vision the versatility to perform a variety of tasks, the vision system should be incorporated with abilities of:

1. Effective method for extracting useful information from scene data for complex industrial parts with heavy noise.
2. Flexibility of the system chat can easily be adapted to perform new tasks.

This paper describes a versatile machine vision system that can recognize a variety of complex Industrial parts based on the previously learned models of them. We use a model guided approach for recognition in which models of objects direct the recognition process by suggesting which features to examine next and their predicted locations. These models can be automatically generated by the system with aids of a human operator showing example parts and teaching important features of them on displays in an interactive way so that the system can be readily applied to new tasks.

Input Pictures and Model Guided Approach

Fig. 1 shows examples of industrial parts used in our experiment. The task of the system is to recognize or classify each object in a scene when machine parts are carried on a belt conveyor. Images are taken from a nearly vertical direction by a T.V. camera and digitized into 6 bits of gray level. There are two sampling modes; one for sampling a quarter region of the T.V. frame in a high resolution and the other for sampling the entire frame in a low resolution. In both modes, a *128x128 digitized* picture *la* sent to and stored in a buffer memory. In this paper we assume that each part in a scene is isolated and not occluded by other objects. This assumption , however, does not restrict the system's performance since parts on a conveyor belt are usually isolated when they are seen from the vertical direction. Even when there were overlapping objects in a scene, the vision part could tell their location and the manipulator could crack and isolate them. Objects are not modified in any way to simplify the task of recognition algorithm, and are very noisy with dirts, grease and highlights. Examples of digitized Images and thier differentiated images are shown in Figs. 2 and 3. In the figures it will be noted that each industrial part has a few stable states and each face of it has different pattern from the other ones. Since we also want to know in which stable states each industrial part is, we call each face of the part as an object.

Most previous works on pattern recognition employed a serial approach of preprocessing, feature extraction and then recognition. This approach is, however, difficult to apply for industrial parts since it is difficult to extract complete features at the stage of feature extraction because of complexity of structures of objects and heavy noise caused by dirts, grease and highlights. We use the model guided approach in which models of objects guide the recognition procedure. A model is a description on the structure of each object and directs the procedure to search and identify a particular object in a scene. It proposes which features to examine next and their predicted locations. This makes the task of feature extraction easy since it is much easier to verify the existence of the predicted features rather than finding them without any knowledge. These models are given for each stable state of the industrial parts and a collection of models represent the system's knowledge on the objects included in the system's repertoire.

An outline of the recognition process is as follows. The system first extracts the most reliable or easy-to-extract information such as size and shape of the outline of an object in a

scene to get candidate models for the input object. Based on the first stage information, an analyzer examines differences between the input object and the models and proposes the moat promising one in a heuristic way to direct the recognition process. The selected model then comes in control of recognition process and suggests which features to examine next and their assumed locations as long as the observed information agrees with the expected one. When they contradict the predicted one (which means the proposed model was a false one), the other promising model is selected by the analyzer based on the Information collected up to now.

Next we consider the problem: how much and what kind of information should be given to the models since it determines the efficiency of the recognition process. Let us see some examples in Fig. 2. Size or shape of the outline of the object 1 is enough to distinguish it from the other objects. The outline information is not enough, however, for the object 3 and the line A is necessary to distinguish it from the object 2. When objects as shown in Fig. 3 are added to the system's repertoire, then a still other feature such as the hole B should be used to distinguish the object 3 from the objects 4 and 5. As we can see f/om the above example, complete descriptions of each object are not necessary and a few Important features are enough to discriminate it from the other objects. Therefore we give the models just enough information to distinguish each object from the other objects, in the order of more distinguishing features to more details. Although the models thus organized makes the recognition process efficient, they should be modified so as to give enough description to distinguish each model from the other models when a new object is Included in the system's repertoire.

The generation or modification of models should be done easily so that the system can be readily applied to new tasks without extensive reptogramming. In our system, it is done automatically by the system while a human operator is showing example objects and teaching important features of them on displays with a cursor In an Interactive way. Described below is a general strategy for learning a new object, where a model of the currently learning object and models of previously learned objects are called a new model and previous models respectively. Since the first features described in models are always the easy-to-extract ones such as size and shape of the outline, they are extracted first from the example object and stored in the new model when the example object is shown by the operator. Then a difference analyzer examines differences between the new model and the previous models and determines similar models. Display routines then show model structures and images of the example object and the similar ones, therefore it Is easy for the human operator to tell which features are the most useful to distinguish the example from the similar ones. He can teach it by specifying a suitable feature extractor and disignating their locations on the displayed image. After the instructed features are extracted and the models are updated, the difference analyzer decides the similar models based on the updated models and the operator instructs more details until the example object is sufficiently discriminated from the previous models.

The idea of teaching the machine interactively in itself is not a new one and was proposed by Tennebaum et al. recently for developing strategies to find a specified object in an office scene(7). The major difference of our learning system from theirs is that ours always tries to recognize the example object and displays similar objects at each time the operator instructs new features as described above. This allows the following advantages:

1) The operator can tell the distinguishing features of the example object easily since they can be found only by comparison with the previously learned objects.

2) He can notice when sufficient features have been instructed, thus the models are given just enough Information.

System Description

Preprocessing and First Stage Processing

For both of learning and recognition, the preprocessing and the first stage processing are performed to get coarse information on each object In a scene. The preprocessor first inputs a scene data in low resolution and detects outlines of objects to locate them in the scene. Since the precise outline is not necessary to locate each object in the scene, a simple procedure is used to find them for speeding up the processing. As the background is assumed to be darker than objects, those points having gray level greater than a certain threshold are considered to be object points. Therefore, a histogram of gray levels of the entire picture points is first computed to determine the threshold as shown In Fig. 4. In the figure, it is decided as the gray level having the first deep valley after the highest peak. When the threshold is determined, the picture data is scanned till an point having gray level greater than the threshold is found. After checking if it is a noise point, the preprocessor traces along the outline of the object in a clock wise direction seeing the object on the right. When the outlines of all objects in the scene have been found, the first stage processor is called and each object is separately analyzed.

It first inputs the picture data of the narrower region containing a certain object in a high resolution to obtain the finer outline of it. The procedure is similar to the one used in the preprocessing except that the histograms of local data containing a portion of the outline instead of the entire data is used to determine the threshold in this case. The reason of using the local histogram is to detect the outline precisely, adapting the threshold dynamically to variances of intensities of the background and the object(8). The sequence of applying local windows is shown in Fig. 5. The local window of the size of 11x11 points is first set surrounding the starting point of the outline obtained at the preprocessing. After computing the intensity histogram of this region, a new threshold Is determined in the similar way as the preprocessing. Then it searches the point having gray level

greater than the threshold and traces along the outline In the local window. When I t touches the current window wall, It sets a new local window* computes a new threshold and continues to track along the outline using the newly computed threshold. Fig-6 shows the outline obtained from the object 5 of Fig. 3.

When the outline of the input object has been found its properties such as size, thinness ratio (9) and shape are computed. The size S and the thinness ratio T are defined as

$S=A \qquad T=4\pi(A/L^2)$

where L: length of the outline
A: area bounded by the outline.

Although the size and thinness ratio are good for representing rough information on shape, they are not sufficient to describe shape information. To extract better information on shape we represent the shape of the outline in a polar coordinate system (r-θ plane) whose origin being the center of gravity of the object(5). As shown in Fig. 7 (a), the distance r(θ) is computed for each discrete value of θ (every 3 degrees) and transformed to a r-θ graph. In the graph values of both r and θ are digitized to discrete ones of 0 to 120. Fig. 7(b) shows the result computed from the outline of Fig. 6. As can be seen from the figure, the r(θ) graph represent features of shape well.

Two r(θ) graphs, $r_1(\theta)$ and $r_2(\theta)$, are compared by shifting $r_2(\theta)$ to the right. Let us define $D(\theta_s)$ for each shifted angle θ_s (every 3 degrees) of $r_2(\theta)$ as

$$D(\theta_s)=\sum_{n=0}^{120} |r_1(3n)-r_2(3n+\theta_s)|$$

Then $r_1(\theta)$ and $r_2(\theta)$ are defined to be similar if the minimum value of $D(\theta_s)$ is less than a certain threshold D_T. Fig. 8 shows $D(\theta_s)$ of objects 1 and 2. As shown in the figure, $D(\theta_s)$ curve has a deep valley if two objects O_1 and O_2 has the similar shape. It does not, however, have any deep valley for objects having different shape. Let us call the shifted angle having the deep valley as the matched angle θ_M, then it should be noted that θ_M represents the orientation of the object 2 relative to the object 1.

Model

The model M consists of a set of models M_1, M_2,... corresponding to each object class O_1, O_2, ... We utilize a tree structure for description of a model M_i which has a set of ordered component nodes C_1. C_2..... Associated with each component C_i is a feature type (FTYPE), a score (SCORE), a parameter list (PLIST) and an attribute list (ALIST). We use several kind of features such as OUTLINE, HOLE, LINE and TEXTURE for description of objects and FTYPE specifies which type of features is used to extract the component C_i. Each model has a corresponding example object E_i. which is stored in a disk. Values of PLIST and ALIST are those extracted from the example objects.

PLIST designates location of a component in the r-6 coordinate system whose origin being the center of gravity of an object $E_{\pm}$. PLIST of HOLE components, for example, has the r-6 coordinate of center of the hole (see Fig. 9). A LINE component is represented by a set of line segments and those end points (r_1, θ_1), (r_2, θ_2),... are given to PLIST. Two PLIST $P_1=(r_{11}, \theta_{11})$, (r_{12}, θ_{12}),... and $P_2=(r_{21}, \theta_{21})$, (r_{22}, θ_{22}),... are defined to be similar if

$$|r_{1i}-r_{2i}|\leq r_T \quad \text{and} \quad |\theta_{1i}-\theta_{2i}+\theta_M|\leq \theta_T$$
$$\text{for } i=1, 2,...,$$

where r_T and θ_T are predetermined thresholds and θ_M is the matched angle described in the previous section. For example, PLIST of holes H_1 and H_2 in Fig. 8 are considered to be similar since

$$|r_1-r_2|\leq r_T \quad \text{and} \quad |\theta_1-\theta_2+\theta_M|\leq \theta_T.$$

ALIST specifies properties of components and OUTLINE, for instance, has size, thinness ratio and r(θ) graph. Two ALISTs:

$A_1=A_{11}, A_{12},...$ and $A_2=A_{21}, A_{22},...$ are defined to be similar if $|A_{1i}-A_{2i}|\leq A_{Ti}$ for i=1,2,..., where A_{Ti} is a predetermined threshold.

Two component C_1 and C_2 are called to be similar if they have the same FTYPE, similar PLIST and similar ALIST. SCORE indicates reliability of extracting the component since some components can be reliably extracted while some others are difficult. It can take a discrete value of 1 to 5. OUTLINE component is always given SCORE S as it Is considered to be the most reliable feature.

Learning

Task of the learning process is to generate a new model M_n and if necessary to modify the previous models $M=M_1,...,M_{n-1}$. Fig. 10 shows a block diagram of the learning process and Fig. 11 gives an illustrating example of teaching the object 5 of Fig. 3 when the models of objects shown in Fig. 1 have already been generated. Explanation of the system will be derived from this example. When an example object E_n is shown to the T.V. camera, ALIST and PLIST of its outline are extracted by the preprocessor and the first stage processor. At each time a new information is obtained, a model updater add new nodes at appropriate position of M_n and if necessary of the previous models. At this time, it generates an OUTLINE node of M_n.

Next a difference analyzer examines the difference between M_n and $M_1,...,M_{n-1}$, and outputs similar models M_{s1}, M_{s2},... It keeps a score table for $M_1,...M_{n-1}$ to judge the similarity of them to M_n. Those models having score greater than a certain threshold S_T are regarded to be similar ones. Let us call the newly generated component node of M_n as C_n. Then the difference analyzer first examines M_1, M_2,... to see if they have similar components to C_n. For those models having no similar component, the value of SCORE associated with C_n is subtracted from their score table which has been initially set to a certain score. Then the difference analyzer checks the score table and sends the similar models having the score larger than S_T to display routines.

Our system is provided with a storage type vector display (SDISP) and a reflesh type color

display(RDISP) a* graphic oucput devices. The dlplay routines then show model structures of M_n and $M_{s1}, M_{s2}, \ldots$ on SDISP, and their images (E and $E_{s1}, E_{s2}, \ldots$) on RDISP. Fig. 11 (a),(b) shows them after OUTLINE feature is obtained. Since the OUTLINE information is not sufficient to discriminate M from $N_1, \ldots$, one should instruct more details. Seeing the displays, It is easy for the operator to tell which features are the most important to distinguish E from $E_1, \ldots$. In the figure, he will notice that the Role A is the most distinguishing one. He can teach this idea to the machine in the following steps while answering the machine's inquiry as shown in Fig. 11 (c)-(e).

1) The system first asks FEATURE TYPE and, in this case, the operator types In HOLE.
2) It then inquires WHICH OBJECT? to know from which object the specified feature should be extracted since It is sometimes necessary to add more description to the similar models. In this case* the operator types in NEW as he wants to extract the hole A of E_1.
3) The system then calls a hole finder to detect holes of E_1 and displays them on SDISP (Fig. d). Then the operator instructs the hole A by designating it with a cursor.
4) Then he inputs 5 for SCORE since the hole A seems to be reliably extracted although other holes may vary according to the lighting condition and noise.

The hole finder uses similar algorithm to the first stage processor and the attributes of size and thinness ratio are stored in ALIST of HOLE component. If the hole's shape is round, its radius is also added to ALIST. Then the similar models are determined in the same procedure as described before. Fig. (f) shows images of E_1 and E_{s1},. If the operator is not satisfied with the result, he can cancell the last instruction and can teach another feature. In this case, he regards that the last instruction was satisfactory since the number of the similar models is decreased from three to one, and thus instructs more complex features.

In Fig. (f), we observe that the lines with the arrow (->) are distinguishing E_1 form E_2. Thus the operator responds LINE to the system's inquiry (g), and then the system calls the gradient operator and displays the differentiated image of them (h). Since he wants to Instruct the lines of E_1, he Inputs NEW to WHICH OBJECT (1). Then the 8lfferentlated image of E_n is displayed on SDISP In a higher resolution and the operator teaches those lines by designating a few representative points of them with the cursor. Then the line finder detects the lines as follows. It first sets the region as shown in Fig. 12 where L is the predetermined length, and then searches an optimum curve in the region. The dynamic programming method proposed by Montanarl(lO) Is used to detect the optimum line. Although this method gives good results to detect lines in a noisy scene, it has disadvantages of consuming too much computation time and memory space. In our case, however, the search space Is restricted to the narrow region and thus the computation time and memory space can be considerably saved. When the lines have been found, the line finder approximates the curve into several line segments and stores r-6 coordinates of their end points in PLIST. ALIST of the LINE component is the average gradient value. Then the difference analyzer again searches the similar models. When no similar models exist, it means that just enough information has been given to the models. Fig. 11 (1) shows an outline, a hole and lines of $E_{,,}$ superimposed on the digitized image and Table 1 gives the generated model.

Recognition

A block diagram of the recognition process Is shown in Fig. 13. In the figure, the preprocessor, the first stage processor and the difference analyzer use the same program modules as the learning process. First of all, the preprocessor locates each object in a scene and sends one of them to the first stage processor which extracts the outline information of the input object. The difference analyzer then examines differences between the models and the observed Information and selects candidate models Mc_1, M_{c2}, ... These candidate models are selected in the same way as the similar models In the learning process although a different term is used. Those information such as the score and the matched angle computed by the difference analyzer are retained in the score table and the matched angle table respectively.

Next the model proposer selects the most promising model Mp among the candidate models M_c], M_{c2} ,....to direct the recognition process in the following steps.
1) Examine the score table and select the models having the best score among the candidate models.
2) Check SCORE of the next node of them and select the models having the largest value If step 1 selects multiple models.
3) Check FTYPE of them, and select the models having the computationally cheaper feature In the order of HOLE, LINE and TEXTURE.
A) If there still exist more than one model for Mp, then go to step 2 and check SCORE and FTYPE of the lower nodes of them.

Then the selected model M_p directs the recognition process in the following sequence:
1) Take next node of Mp to propose which feature to examine next and its assumed location. FTYPE of the node specifies the feature type to be used and PLIST predicts the location of the proposed feature by checking the matched angle of Mp in the matched angle table.
2) Extract the proposed feature by corresponding feature extractor.
3) Compare the observed parameters and properties to the values of PLIST and ALIST of the current node of Mp.
4a) If they are similar (In a sence defined at the previous section), updates the matched angle table based on the measured parameter, then go to step 1. If not, go to step 5.
5) The difference analyzer determines the candidate models based on the information collected up to now and the model proposer selects another promising model Mp. Then go to step 1.

The above loop ends when some model Mp reaches the terminal node, that is, nothing to do Is left in the model. Since the threshold ST I S initially set to a strict value to make the tree search fast, it sometimes happens that no candidate models

exist even if any model have not reached the terminal node. Then the threshold S_T is lowered until some model reaches the terminal node.

Fig. 14 shows an example of experimental results where (a) is an differentiated image of an input object and (b) shows an outline, a hole and a line that have been used to recognize the Input object. It was successfully recognized as a top face of the industrial part no. 20 (see OBJECT 3 of Fig. 2), and its position and orientation were also measured precisely.

Conclusions

This paper has described a versatile machine vision system that can recognize a variety of complex industrial parts based on the previously learned models of them. The system has been implemented in a mini-computer DEC PDP-8/E with a 12 KW core memory and an additional buffer memory (28 K bytes) which is used for the data structure of the digitized image and the models.

To test the proposed system, it has been applied to various sets of parts of small industrial gasoline engines each of which consisting of twenty to thirty parts. Models of them could be successfully generated even by the persons who arc not familiar with computer programming. Then these learned models have been used to recognize randomly placed industrial parts and most of them have been correctly recognized. Recognition time and instruction time for an object were about 30 seconds and 7 minutes respectively.

Computing time for recognition is too long for practical applications because no attempt is made to save i t , and we consider some programming effort could decrease it considerably. Color Information is not used in the current system since most industrial parts used in the experiment are gray. Color information,however, wiIl be useful for the first stage classification when objects have different color. The most restrictive assumption in the present system is assumption that each object in a scene is isolated, and a future study is necessary to recognize the occluded objects.

References

1] Ejiri, M. et al., " An intelligent robot with cognition decision-making ability". Second IJCAI, pp. 350-358, 1971.

2] Winston, P. H., " The MIT robot", Machine Intelligence 7, Edinburgh University Press, pp. 431-463, 1972.

3] Ambler, A.P. et al., " A versatile computer-controled assembly system", Third IJCAT, pp. 298-307, 1973.

4 Olsztyn, J.T. et al., " An application of computer vision to a simulated assembly task", Proc. of the First International Joint Conf. on Pattern Recognition, pp. 505-513, 1973.

5] Yoda, H & Ejiri, M, Conference of the Institute of Electronics and Communication Engineers of Japan, 1972 (i n Japanese).

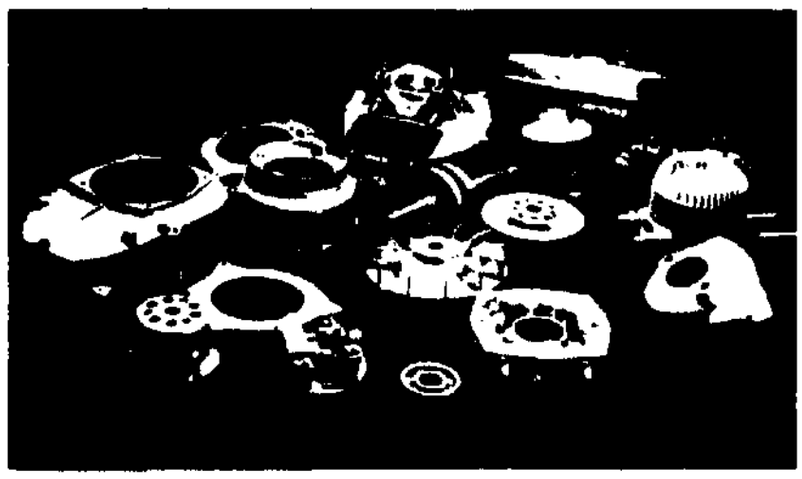
6] Chi en, R.T. et al., " Visual understanding of hybrid circuits via procedural models" (To be presented at this symposium).

7] Tenenbaum, J.M. et al., " ISIS: An interactive facility for scene analysis research", Proc. of the Second International Joint Conf. on Pattern Recognition, pp. 123-125, 1974.

8] Chow, C.K. & Kaneko T., " Boundary detection of radiographic images by a threshold method", S. Watanabe (Ed.), Frontiers of Pattern Recognition, pp. 61-82, Academic Press, 1972.

9) Duda, R.O. & Hart, P.E., " Pattern classification and scene analysis", p. 350, A Wiley Interscience, 1973.

10] Montanarl, U., " On the optimal detection of curves in noisy pictures", CACM, vol. 14, no. 5, pp. 335-345, 1971.



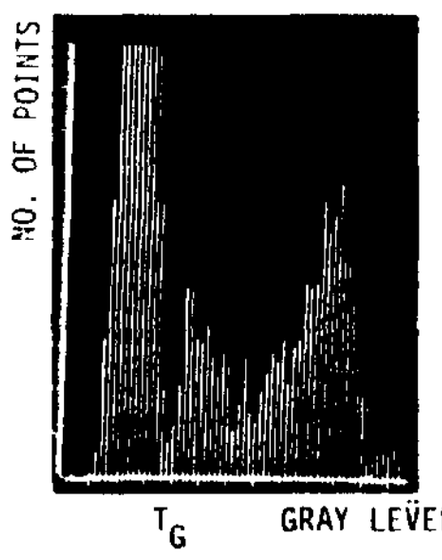


Fig. 4 Intensity Histogram

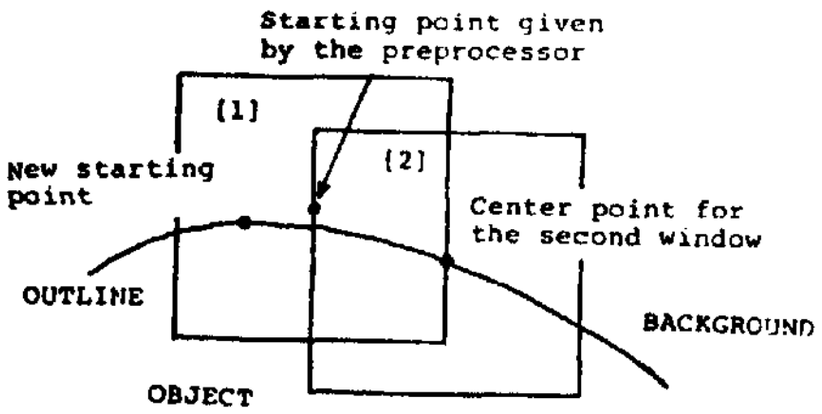
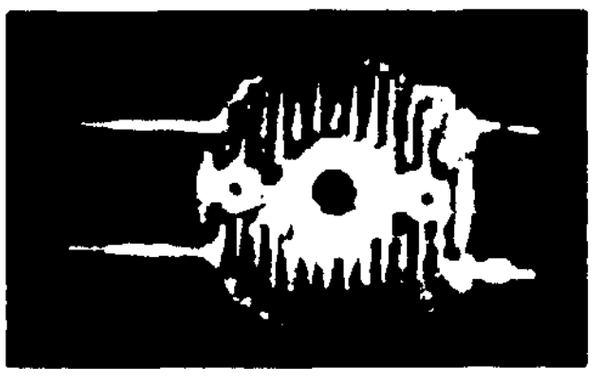
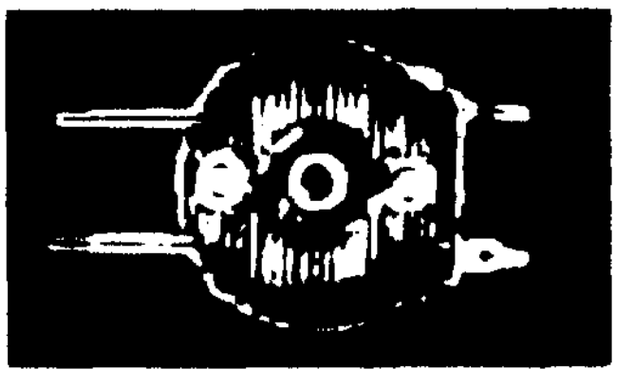


Fig. 5 Sequence of Applying Local Windows

OBJECT 1

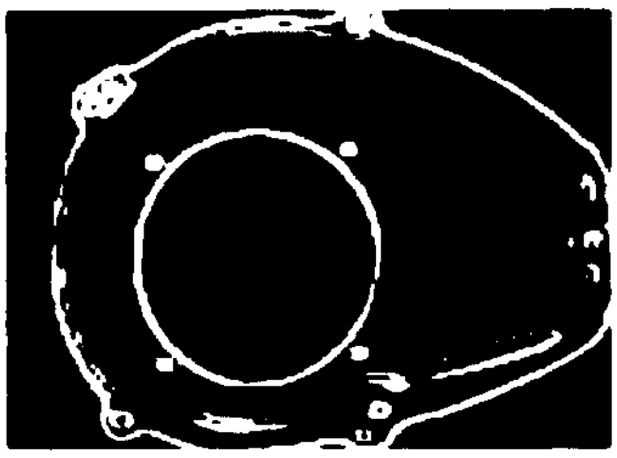
TOP FACE OF
PART NO. 9





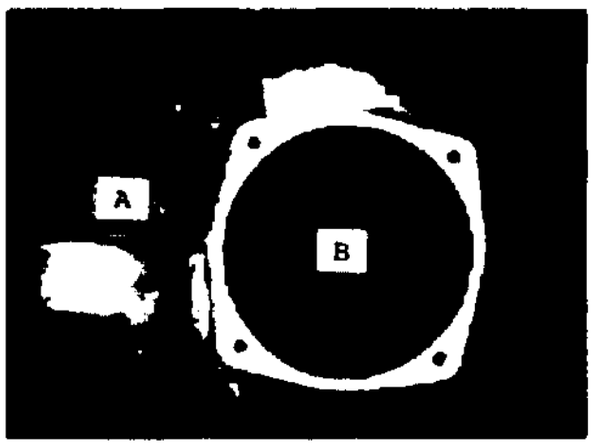
OBJECT 2

BOTTOM FACE OF
PART NO. 20



OBJECT 3

TOP FACE OF
PART NO. 20



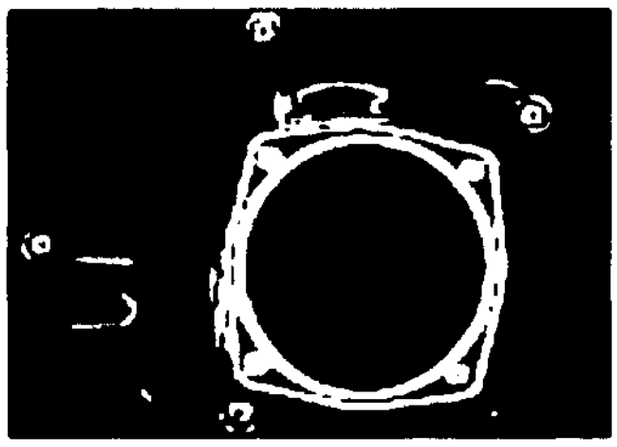
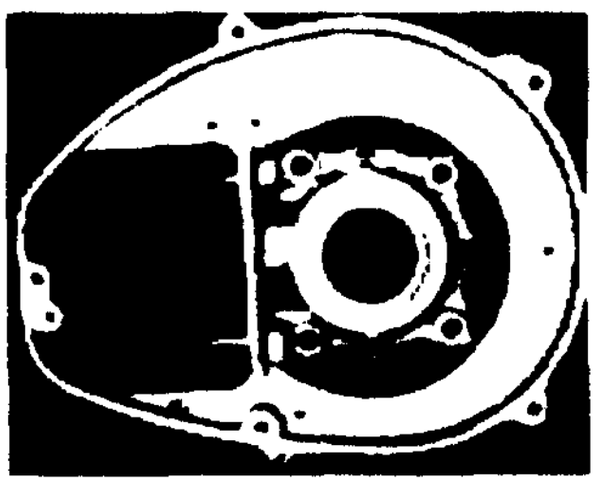
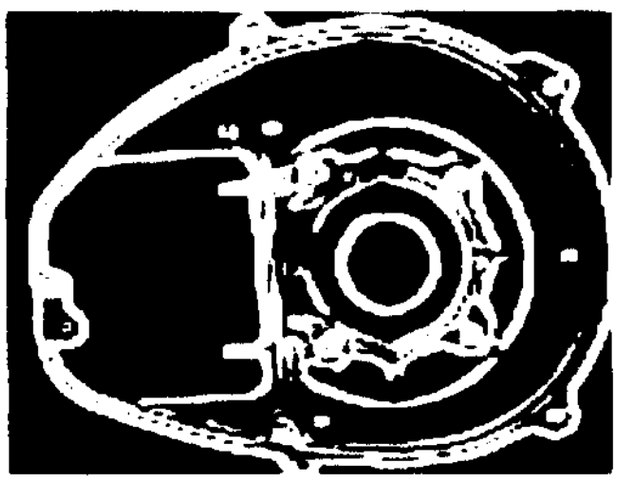


Fig. 2 Examples of Digitized Images and Their Differentiated Images

OBJECT 4

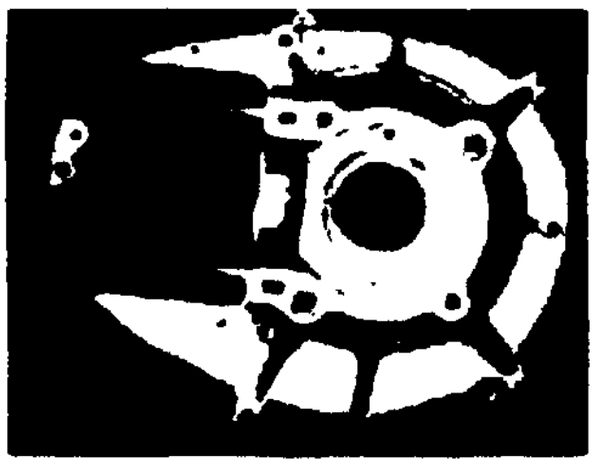
BOTTOM FACE OF
PART NO. 21





OBJECT 5

TOP FACE OF
PART NO. 21



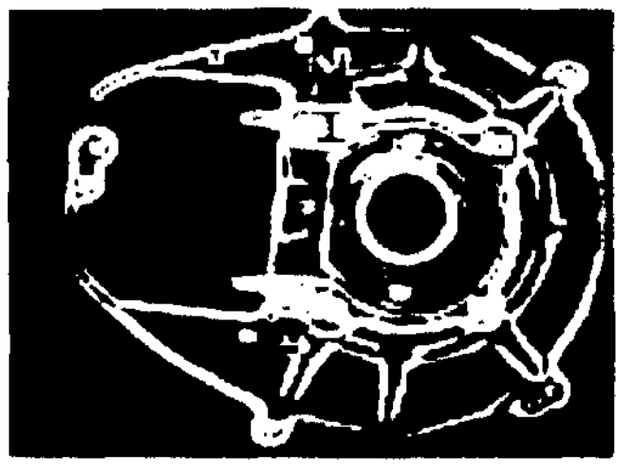


Fig. 3 Other Examples

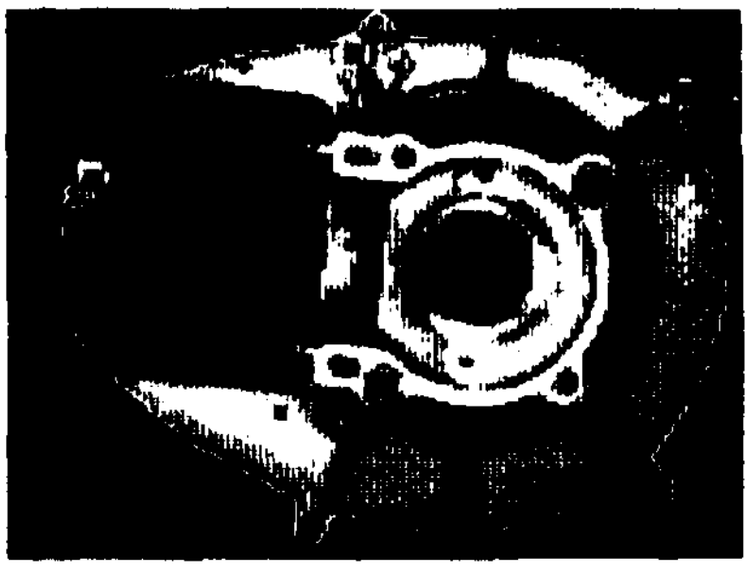


Fig. 6 Outline Obtained by the First Stage Processor

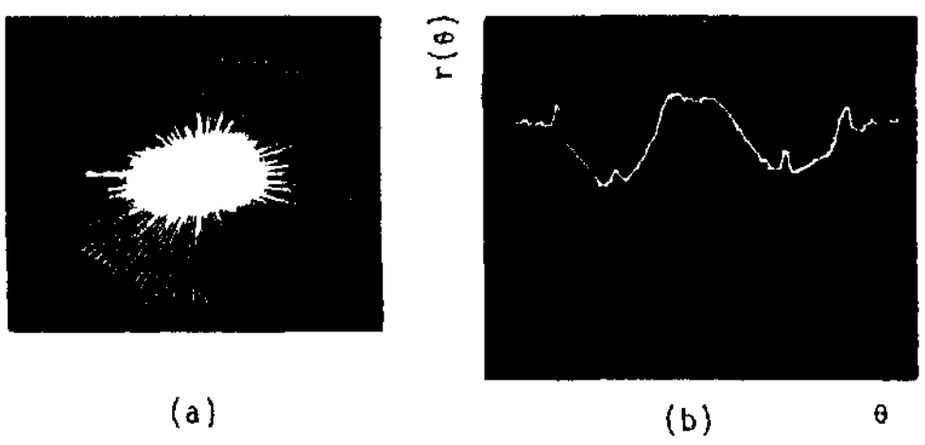


Fig. 7 r(θ) Graph of OBJECT 5

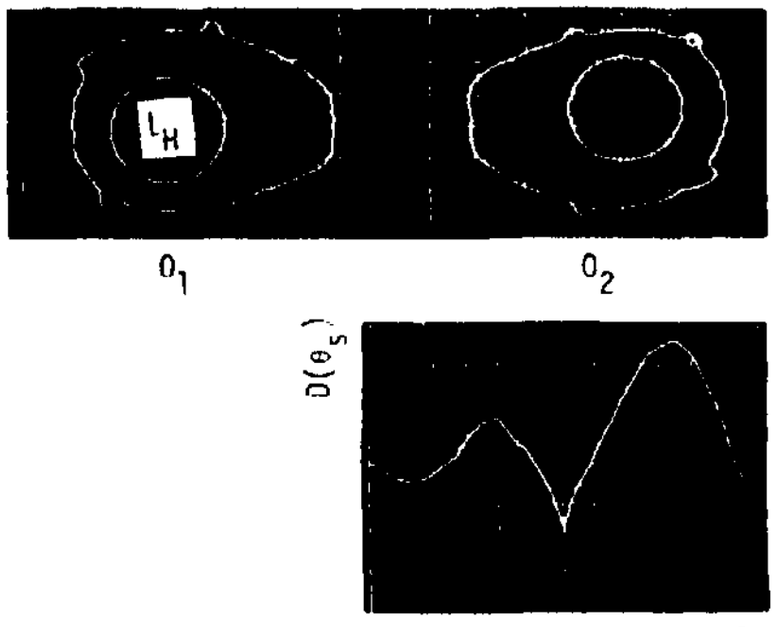


Fig. 8 $D(\theta_S)$ Graph of Objects O_1 and O_2

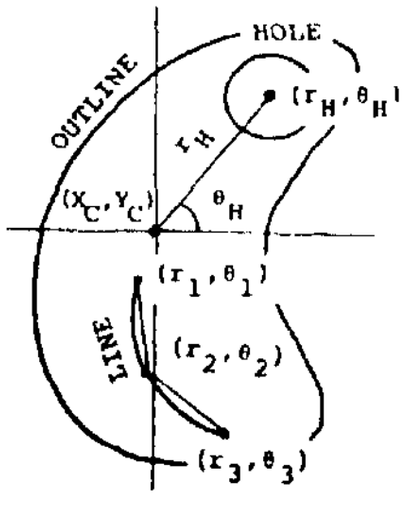


Fig. 9 Parameter List

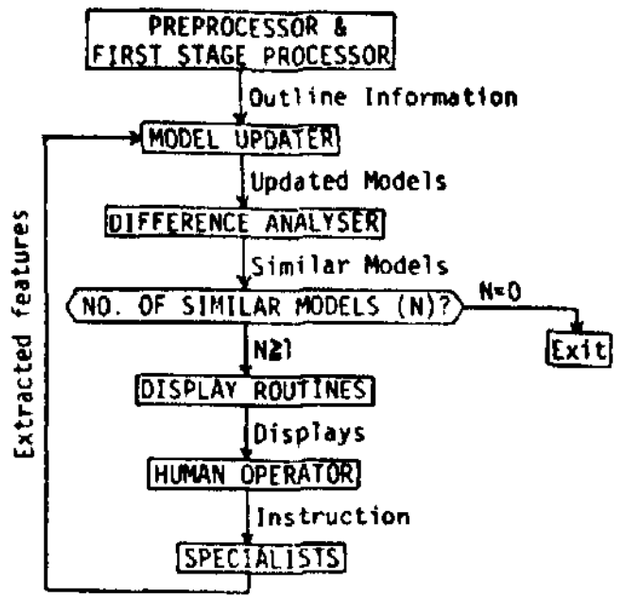


Fig. 10 Block Diagram of the Learning Process

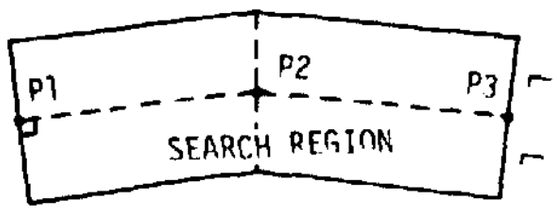


Fig. 12 Search Region

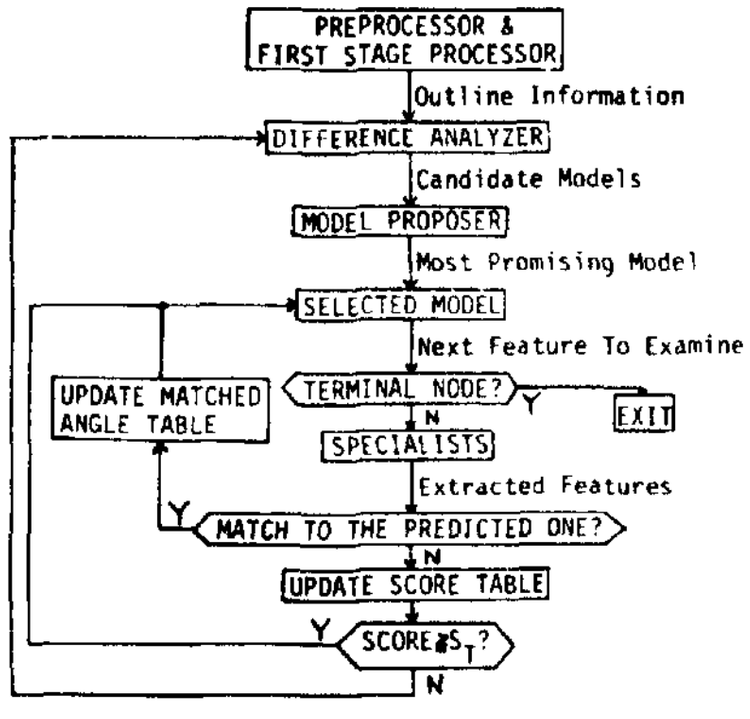


Fig. 13 Block Diagram of the Recognition Process

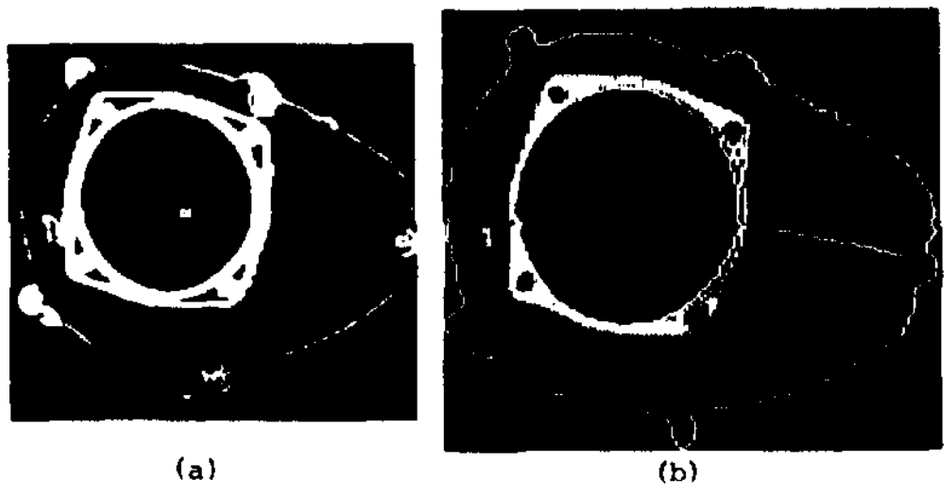
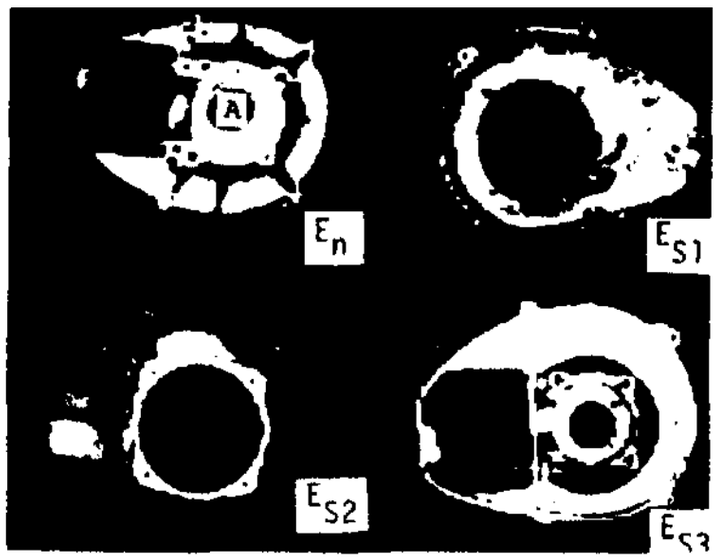
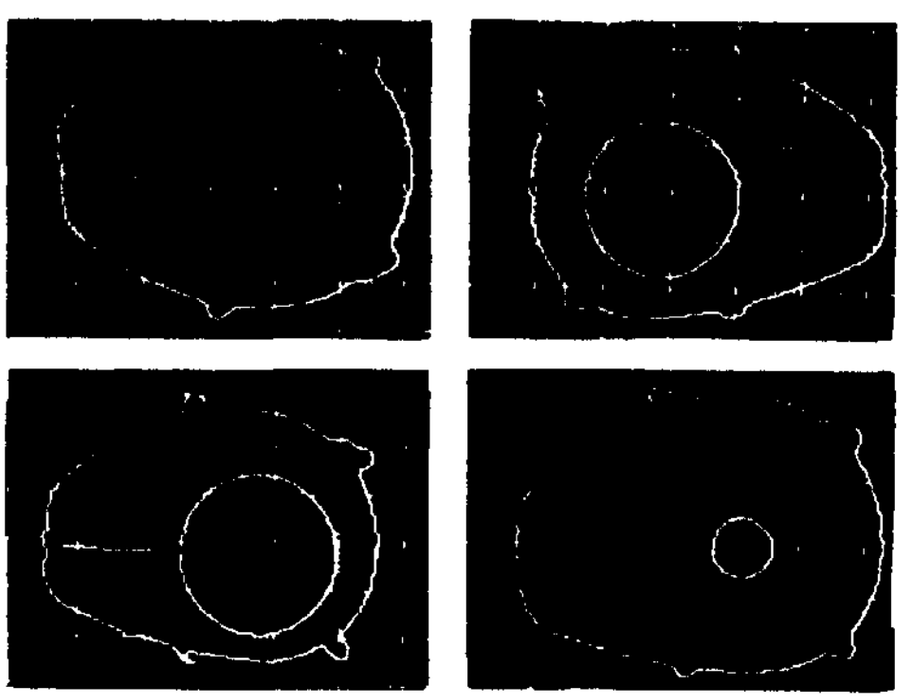


Fig. 14 An Example of Experimental Results

(a)



(b)

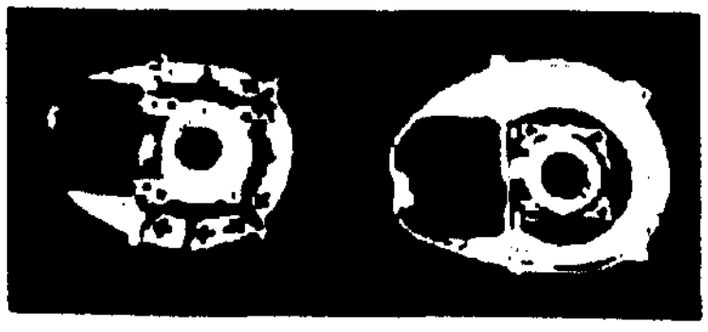


(c) FEATURE TYPE? HOLE
WHICH OBJECT? NEW

(d)

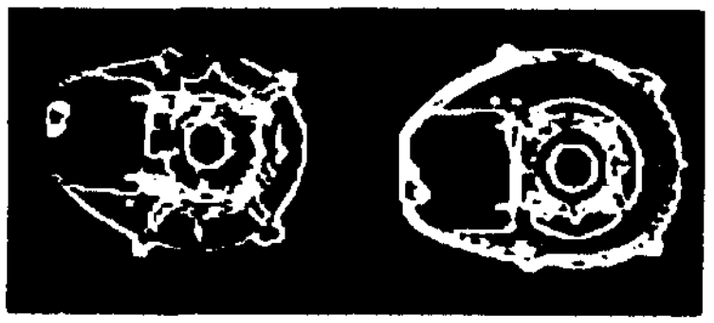
(e) SCORE? 5

(f)



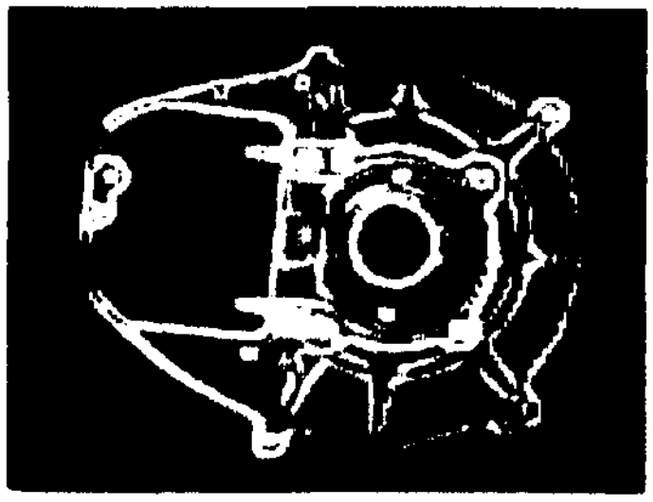
(g) FEATURE TYPE? LINE

(h)



(i) WHICH OBJECT? NEW

(j)



(k) SCORE? 1

(l)

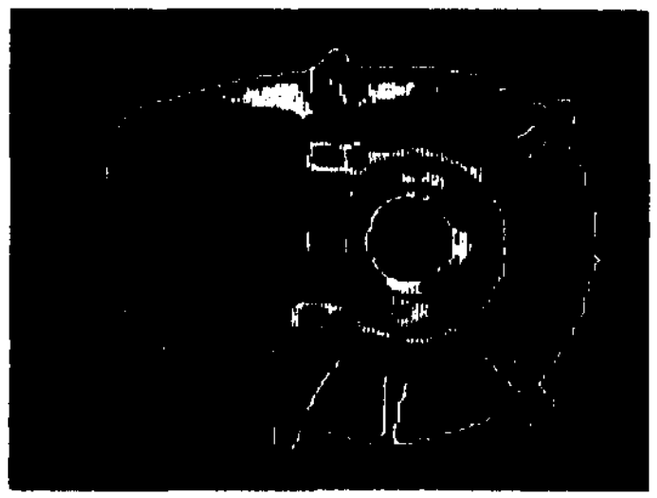


Fig. 11 An Illustrating Example of Learning Process

MODEL 41 PART NO. 21 TOP FACE

Component	FTYPE	SCORE	PLIST	ALIST
1	OUTLINE	5		AREA 986 THINNESS 22 r(θ) GRAPH
2	HOLE	5	(2,190)	AREA 24 THINNESS 12 RADIUS 8
3	LINE	1	(35,261),(50,253)	51
4	LINE	1	(37,280),(51,278)	48
5	LINE	1	(39,290),(53,280)	47
6	LINE	1	(44,318),(57,319)	45
7	LINE	1	(47,325),(60,320)	47
8	LINE	1	(49, 28),(59, 32)	50
9	LINE	1	(51, 35),(58, 36)	48

(ALIST: AVERAGE GRADIENT VALUE)

Table 1. The Generated Model