# Recognition of Multiple Objects Using Geometric Hashing Techniques

Jeff Edwards and Dr. Rahmatallah Shoureshi
Engineering Research Center for Intelligent Manufacturing Systems
Purdue University
W. Lafayette, IN 47906

*Abstract* - This paper discusses the development of a robust robot vision system for implementation in a Flexible Assembly Cell. The vision system is capable of recognizing the identity and returning the three-dimensional position and orientation of each object in a physical scene. The scenes of interest may consist of one or more (possibly occluded) industrial objects. A typical unstructured factory lighting environment is assumed. The resulting vision system is model-based and learns an object either through CAD data or by physically viewing the object via a black and white CCD intensity image. The geometric representation of the object is generated off-line and stored in a hash table. During recognition, the hash table is accessed to identify the object and return a transformation matrix encoding its three-dimensional pose. The hash table makes use of the straight- and parallel-line invariance properties of the affine approximation to the perspective transform, and allows for the bulk of the computational load to be shifted off-line. A feature point classification approach has been implemented which results in significant reductions in both hash table size and on-line recognition times. The latest version of our software recognizes relatively flat industrial objects in any three-dimensional configuration and from any viewing angle. Experimental results for multiple real-world objects (such as screwdrivers and pliers) in occluded scenes have resulted in recognition times of less than one second per object. In addition, this geometric hashing technique may be easily extended to the recognition of general three-dimensional objects.

## INTRODUCTION

The dynamic, global, competitive market requires industries with the ability to easily change products and with the flexibility to use machines for processing a variety of products. This will require new techniques for flexible manufacturing and the integration of these advances into functioning prototype systems. One of the key components of our research goal is the establishment of a flexible assembly cell with the ability to assemble a wide variety of components while requiring little or no human intervention. The assembly cell currently envisioned will include features with the ability to deal effectively with rapidly changing production orders and unexpected events. The intelligence behind these features will consequently require a substantial amount of sensory data which can best be provided through visual feedback. This particular application of machine vision will take the form of object recognition in which industrial parts and component sub-assemblies are rapidly and robustly identified within the workspace and information regarding their position and orientation provided to the assembly cell Task Planner. The object recognition system discussed in this paper will be essential to the success of the flexible assembly cell. Expected applications include assembly error detection and recovery, flexible fixturing, coordination of multiple robots, and general part location tasks.

The purpose of our study was to focus on the development of recognition techniques that do not rely on structured lighting environments, binocular vision, or laser range data. With this goal in mind, studies done with affine transformations and geometric hashing became of interest. Wolfson, et. al., [1,2,3] have combined the affine approximation to the perspective projection with an efficient matching algorithm to create a system capable of recognizing multiple occluded objects from a single gray scale camera image. The matching algorithm shifts the bulk of the computational load off-line by computing the possible affine model-to-image transformations in advance and storing them in a hash table, thus considerably reducing the actual on-line recognition time. This approach has become the basis of the object recognition system being developed for our flexible assembly cell.

0-7803-0891-3/93$03.00 ©1993IEEE

## RESEARCH OBJECTIVES

For the purposes of our assembly cell vision system, the object recognition problem is defined as follows: the system is presented with a scene consisting of many (approximately flat) industrial components, possibly occluding one another, randomly positioned on the assembly table. An unstructured factory lighting environment is assumed. No a priori knowledge of the scene is available, but it is assumed that the vision system has previously learned each of the distinct object types and has stored their geometric representations in a data base. The goal is to associate each of the physical objects with a model stored in this data base, and to determine the three-dimensional pose of these objects with respect to a world coordinate frame.

## THE GEOMETRIC HASHING TECHNIQUE

As mentioned in Section II, we are currently dealing with the recognition of approximately flat objects (whose height variation is relatively small compared to their distance from the camera) resting on an assembly table and viewed from above. A set $P_I$ consisting of $N$ geometric interest points is extracted from the resulting images. Since we are dealing with flat objects, these interest points are assumed to lie on a single plane $z = z_0$ parallel to the work table:

$$P_I = \{(x_1, y_1, z_0), (x_2, y_2, z_0), \ldots, (x_N, y_N, z_0)\}$$

These object interest points are mapped into a set $P_O$ of observed points, via the perspective projection:

$$P_O = \{(u_1, v_1), (u_2, v_2), \ldots, (u_N, v_N)\}$$

The perspective mapping is as follows:

$$u_i = f\frac{a_{11}x_i + a_{12}y_i + a_{13}z_0 + t_1}{a_{31}x_i + a_{32}y_i + a_{33}z_0 + t_3} \qquad v_i = f\frac{a_{21}x_i + a_{22}y_i + a_{23}z_0 + t_2}{a_{31}x_i + a_{32}y_i + a_{33}z_0 + t_3}$$

This mapping is a function of the rotation matrix $A$ and translation vector $T$ which relate the model and camera reference frames, and the focal distance $f$. When the height $t_3$ of the camera above the object is sufficiently large in comparison with the term $a_{31}x_i + a_{32}y_i$ for each of the $N$ interest points, then the affine approximation to the perspective projection may be assumed:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

With the affine approximation valid, there exists a two-dimensional transformation between the set $P_I$ of object interest points and the set $P_O$ of observed image points. This transformation $T$ consists of a two-dimensional matrix $A$ which accounts for scaling, rotation, and skewing, and a vector $b$ accounting for translation:

$$T\begin{pmatrix} x \\ y \end{pmatrix} = A\begin{pmatrix} x \\ y \end{pmatrix} + b$$

A noncollinear set of three of the $N$ image points may be selected to form a two-dimensional linear basis which allows the remaining $N-3$ image points to be expressed in terms of the selected basis-triplet as follows:

$$\boldsymbol{p} = \alpha(\boldsymbol{e}_{10} - \boldsymbol{e}_{00}) + \beta(\boldsymbol{e}_{01} - \boldsymbol{e}_{00}) + \boldsymbol{e}_{00}$$

where $\boldsymbol{p}$ represents the image coordinates of one of the $N$ observed points, $\boldsymbol{e}_{00}$, $\boldsymbol{e}_{10}$, and $\boldsymbol{e}_{01}$ are the image coordinates of the three ordered basis points, and $\alpha$ and $\beta$ are the affine coordinates of the point $\boldsymbol{p}$. Note that the basis points $\boldsymbol{e}_{00}$, $\boldsymbol{e}_{10}$, and $\boldsymbol{e}_{01}$ will have affine coordinates (0,0), (1,0), and (0,1), respectively. Now if the set of interest points undergoes an affine transformation $T$, the affine coordinates of each point in the set will be unchanged provided that the same three ordered points are selected to form the basis triplet both before and after the transformation:

$$\boldsymbol{Tp} = \alpha(\boldsymbol{Te}_{10} - \boldsymbol{Te}_{00}) + \beta(\boldsymbol{Te}_{01} - \boldsymbol{Te}_{00}) + \boldsymbol{Te}_{00}$$

This invariance property under affine transformations is fundamental to the geometric hashing technique. The goal is to find a match between an image basis-triplet and a previously learned model basis-triplet such that the affine coordinates of the remaining interest points in both the image and model sets are equivalent. Once this is accomplished, the recognition of the object in the image is established, and the affine transformation mapping the model points to the image points may be generated. This information is then used to compute the pose of the recognized object with respect to some world coordinate frame.

The geometric hashing technique described in [1,2,3] is designed to speed the process of finding such a correspondence. It has two stages: off-line learning and on-line recognition. In the learning stage, the affine coordinates of an observed interest point are computed for all possible ordered basis triplets in $P_O$. For each of these possible triplets, a record of the form *(which model, which basis triplet)* is inserted into a hash table. The appropriate bin of the table is determined based on the affine coordinates $(\alpha,\beta)$ generated by each particular basis triplet. This procedure is repeated for all $N$ of the observed interest points. When completed, the entire set $P_O$ representing the model object will have been encoded using a hashing scheme invariant to affine transformations. This process is performed for each object to be learned by the system.

During the recognition stage, only a single image basis-triplet is selected. The resulting affine coordinates of the image interest points are computed and the appropriate bins of the hash table are accessed. Each of the records contained in these bins "votes" for its corresponding model and basis-triplet. The particular record receiving the most votes has a high probability of representing the correct identity and pose of the object.

The geometric hashing approach is both robust and efficient. If several interest points are missing or displaced due to image noise or occlusions, the remaining interest points can usually generate enough votes for the correct record to achieve a successful recognition. In addition, by computing the model affine coordinates for all possible basis-triplets in advance, the bulk of the computational load is shifted off-line.

## THE FEATURE POINT CLASSIFICATION APPROACH

The geometric hashing technique is very powerful; however, it has substantial room for improvement. The most serious shortcoming is the size of the hash table itself. Assuming that all possible ordered basis-triplets are in fact used during the learning stage, then there will be $N_B$ sets of model affine coordinates:

$$N_B = N(N-1)(N-2)$$

where $N$ represents the number of interest points associated with the particular model to be learned. Each of these sets will require entering $N-3$ records into the hash table. Consequently, the learning of a single object will require a total of $N_R$ records to be entered into the table:

$$N_R = N(N-1)(N-2)(N-3) \approx N^4$$

For complex real-world objects, the quantity of required records quickly becomes unmanageable. For example, during our experimentation we found that approximately 45 interest points were needed to accurately describe the geometry of a pair of pliers. This translates into roughly four million hash table records and requires huge memory resources. Even worse, on-line hash table access times are directly proportional to the number of records stored in the hash table. In addition, for even more complex objects, the hash table size increases at an exponential rate. This was unacceptable, and it became clear that $N_R$ needed to be reduced substantially.

One way of reducing $N_R$ would be to simply reduce the number of interest points $N$ extracted from a model or scene. For instance, by halving $N$, the number of required records $N_R$ is reduced by a factor of 16. However, this has the undesirable effect of discarding valuable geometric information describing the object.

Instead, we formulated an alternative feature point classification approach. Instead of treating all $N$ interest points equally, a small fraction $\eta$ of them are classified as robust points. The remaining $(1-\eta)N$ extracted interest points are designated as non-robust points. In our experiments, we found that high curvature was a reliable criterion for robustness. In other words, interest points at curve locations where the curvature was greater than a certain threshold $\theta_R$ were almost always extracted robustly from any given image of the object despite glare, shadows, or other noise effects.

Since the $\eta N$ robust points are extracted reliably, we imposed the following restriction: a valid basis-triplet may be composed only of three robust points. With this restriction, the number of possible basis-triplets $N'_B$ that must be processed in the learning stage is reduced substantially:

$$N'_B = \eta N(\eta N - 1)(\eta N - 2)$$

Of course, all $N$ of the original interest points needed to adequately describe the object geometry are still retained. Consequently, the total number of records $N'_R$ entered into the hash table is as follows:

$$N'_R = \eta N(\eta N - 1)(\eta N - 2)(N-3) \approx \eta^3 N^4$$

This allows the complete object geometry to be encoded into a much smaller hash table. For example, we found that the pliers object mentioned above exhibited 10 highly robust points. Consequently, using our feature point classification approach, only 45,000 records were required (a hash table size reduction of almost two orders of magnitude.) In addition, the relationship between hash table size and non-robust points is now linear rather than exponential. This allows us to generate artificial interest points to encode additional information about the object geometry without incurring a massive increase in hash table size. In our implementation, we generated such artificial points periodically along relatively straight curves and line segments where there would otherwise be no extracted interest points. This allows us to effectively encode entire lines and curves rather than simple noise-sensitive points. In addition, since only robust points are allowed to form basis-triplets, the process of selecting a successful basis during the on-line recognition stage is greatly simplified because a much smaller number of possible combinations are available for consideration. We have implemented several algorithms that successfully select a numerically stable basis-triplet.

## SOFTWARE IMPLEMENTATION

Although the geometric hashing and feature point classification techniques provide the basis of our object recognition approach, the overall scene analysis algorithm is quite complex. The following is a brief description of the primary steps involved in extracting the identity and pose of all objects present in a given scene:

*Curve Generation*: Once a Sobel edge-intensity map has been generated from the raw image, a set of curves are extracted using a ridge-tracking routine. A set of interest points are then generated from these curves using a segmentation algorithm discussed in Haralick and Shapiro [4]. These points, which represent the geometry of the scene object(s), are then adjusted to take into account the aspect ratio of the CCD camera. The resulting interest point curves are then divided into curve groups based upon their location within the scene. For example, if a scene consists of two non-occluded objects, then the curves associated with these two objects will be divided into two distinct groups. From this point on, each curve group is treated as if it represents an independent image. These derived images may now consist of either a single object or multiple occluded objects, but the case of multiple non-occluded objects has been eliminated.

*Curve Processing*: The curves within a group are now processed to generate a set of robust and non-robust points. Each interest point is tested to determine whether it represents a location of unusually high curvature. If so, it is designated a robust point; otherwise, it becomes a non-robust point. Artificial non-robust points are also generated at this time to encode additional geometric information (such as straight lines) not already represented by the original set of interest points.

*Basis Selection*: The basis selection algorithm initially assumes that only a single object is present in the curve group. A triplet of robust interest points is selected that generally maximizes the area of the enclosed triangle to promote numeric stability. If the curve group actually consists of two or more occluded objects, this approach may produce a triplet containing points on different objects. When such a basis is used to access the hash table, a meaningless match candidate will be generated and detected, and the system will try again by selecting a new basis. On the second and (if necessary) subsequent attempts, a different algorithm is employed that uses pre-processed information stored in the model database to efficiently search for a triplet of robust interest points all of which are associated with a single object.

*Vote Casting and Counting*: In our implementation, two hash tables are maintained: one built from robust points and the other from non-robust points. Using the selected triplet as a basis, the affine coordinates of both the robust and non-robust points are calculated. These coordinates are used to access the appropriate bins of their respective hash tables. Each record in the accessed bins votes for its respective model and basis; records accessed from the robust hash table carry extra weight. After casting, a vote-counting algorithm selects one or more match candidates. Each candidate represents a possible match between the selected image basis-triplet and one of the many basis-triplets generated during the learning phase (as well as a model ID).

*Candidate Filtering*: Each of the match candidates is tested to determine which one represents the best match. Several computationally inexpensive checks are used to filter out bad matches early. One such filter checks to see how many of the image robust points would match with the candidate model's robust points if a transformation based upon the candidate's basis-triplet were applied. Another filter checks whether or not the model-to-image transformation matrix is highly skewed (in which case the candidate is rejected.) If one or more candidates pass all the filters, the algorithm selects the candidate which maximizes an index of performance and rejects the others. If no candidates pass all the filters, a new basis is selected (see Step 3) and the process is repeated.

*Verification*: The match candidate selected by the filtering module is now verified. Basically, this is accomplished by superimposing the candidate model's curves over the extracted image curves using the candidate's model-to-image transform (computed by the filtering module.) Several measures of curve matching are calculated, and a fuzzy logic decision module then answers the question: "Does this candidate represent a valid match with an object in the scene?" If the match is deemed successful, the matched portions of the image curves are removed from the scene. If not, a new basis is selected and the process is repeated.

*Scene Evaluation*: Once a successful object match has been achieved and the recognized object has been removed from the scene, there may still be additional curves present in the image. In this case, a second fuzzy logic module answers the question: "Do the remaining image curves represent additional objects in the scene, or mere noise?" Several factors are used to make this decision, such as the percentage of curve segments remaining and the quantity of objects previously recognized. If the module decides to continue searching for objects, a new basis is selected using the updated image curves; otherwise the scene analysis for that particular curve group is completed.

## EXPERIMENTAL RESULTS

The algorithms discussed in this paper have been implemented using the C++ language. A vision software package has been developed that runs on X Windows-based workstations using any available image processor or frame-grabbing hardware. Currently, we are running the object recognition software on a Sun SPARC workstation. The vision hardware consists of an Imaging Technologies ITEX 151 image processor and a 512 by 512 pixel CCD camera mounted on the end effector of an ADEPTONE robot. An independent image acquisition driver has been developed to interface with the ITEX processor.

Our vision system has learned to recognize a set of tools. Figures 1a through 1g show some of the primary stages of the object recognition process when performed on a scene containing a pair of pliers. Both a screwdriver and a pair of pliers are identified and located in the scene of Figure 2. Finally, Figure 3 shows the results for the same two tools in an occluded configuration. The white outlines in these two figures show the calculated pose of the recognized objects superimposed over the respective raw images.

These results were achieved by using the image processing hardware to acquire the images and apply a Sobel edge detection convolution. This results in an edge-strength map of the scene which is then input to the vision software. We define our system's total recognition time to be the time required by the vision software to identify and locate all objects in the scene when given such an edge-strength image as input. Using this definition, our system achieved the following total recognition times:

| Figure | Scene Description | Tot. Recog. Time |
|---|---|---|
| 1 | Single Object (pliers) | 0.678 sec |
| 2 | Two Unoccluded Objects | 0.896 sec |
| 3 | Two Occluded Objects | 1.142 sec |

## CONCLUSION AND FURTHER RESEARCH

This paper has presented a technique for the rapid recognition of complex and occluded objects under normal lighting and operating environments. The geometric hashing technique has three major advantages: 1.) it operates on local features and hence allows both occluded and poorly illuminated objects to be recognized; 2.) it shifts the bulk of the computational load off-line for faster recognition times; and 3.) it can be generalized relatively easily to the recognition of non-flat 3D objects. The particular geometric hashing implementation discussed in this paper improves on previous approaches by incorporating feature point classification, which distinguishes between robust and non-robust interest points. This drastically reduces the number of records stored in the hash table; consequently, massive reductions in on-line hash table access times as well as hash table memory requirements have been achieved. In addition, the identification of robust interest points has enabled the basis selection module to perform much more efficiently. In summary, our technique represents an ideal solution to the vision problems associated with a flexible assembly cell.

It should be noted that the object recognition system being designed for our assembly cell is still in an early stage of development. Several major improvements are planned for the near future. One of our major goals is to generalize the software package to allow recognition of non-flat 3D objects from a single 2D image taken with an arbitrary camera orientation. A number of such algorithms are described in [2]. The major implications of such an extension are that a basis 4-tuple will be required rather than a basis triplet, and interest points will be specified by three coordinates rather than two.

In addition, we hope to develop a supervisory fuzzy logic module that will automatically adjust many of the system parameters (such as the curve extraction thresholds) to take into account such factors as lighting conditions and the particular characteristics of the class of objects that are being recognized.

## ACKNOWLEDGMENT

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

## REFERENCES

[1] R. Hummel & H. Wolfson. "Affine Invariant Matching." *Proceedings of the DARPA Image Understanding Workshop*. 1988, pp. 351-364.

[2] Y. Lamdan & H. Wolfson. "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Proceedings of the Second International Conference on Computer Vision*. 1988. pp. 238-249.

[3] Y. Lamdan, J. Schwartz, & H. Wolfson. "Object Recognition by Affine Invariant Matching," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1988. pp. 335-344.

[4] R. M. Haralick & L. G. Shapiro. *Computer and Robot Vision. Vol. 1* New York: Addison-Wesley Publishing Co., 1992. ch. 11. pp. 563-565.

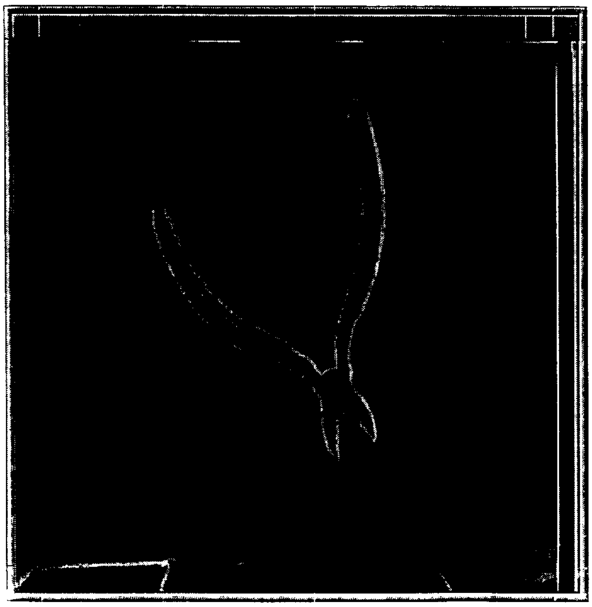
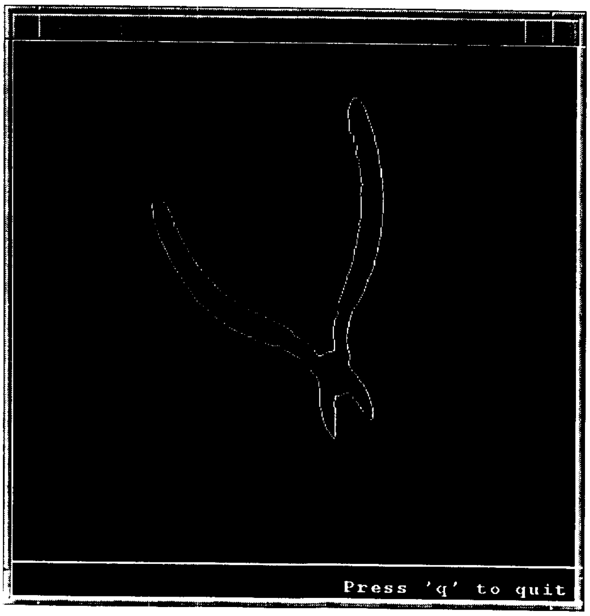
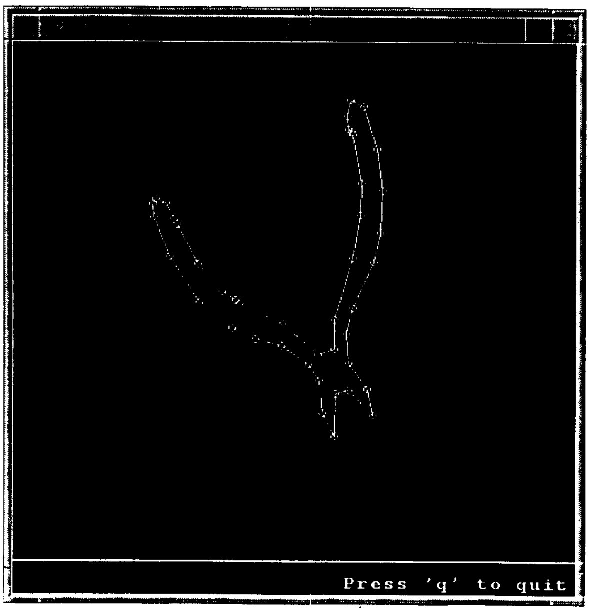


Fig. 1a)



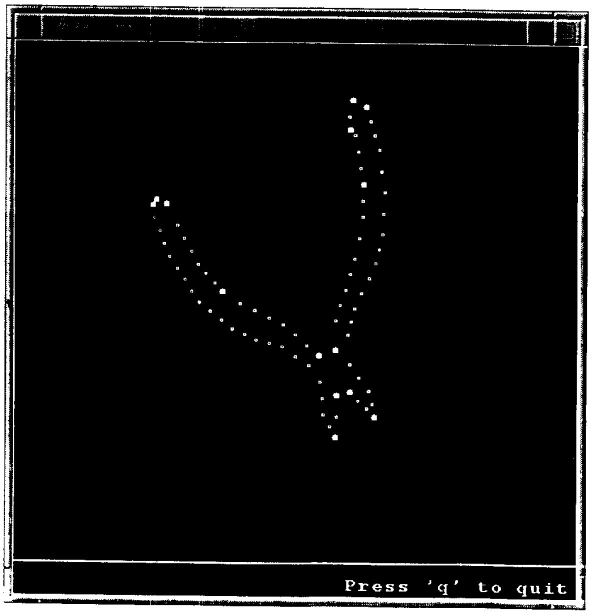
Press 'q' to quit

Fig. 1b)



Press 'q' to quit

Fig. 1c)



Press 'q' to quit

Fig. 1d)

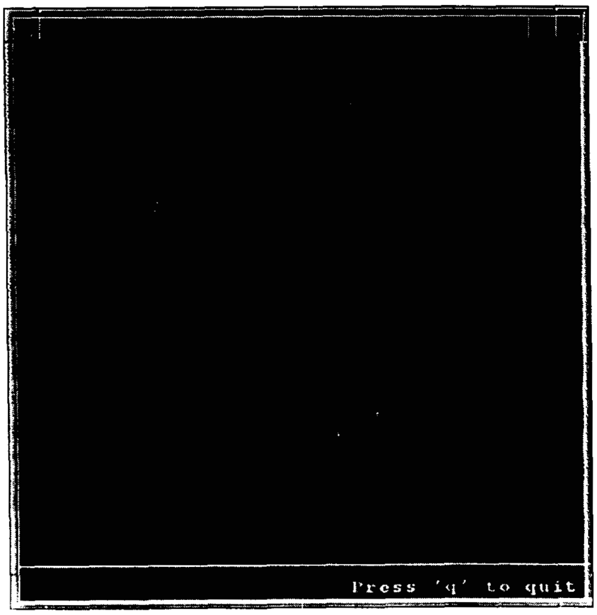


Fig. 1e)

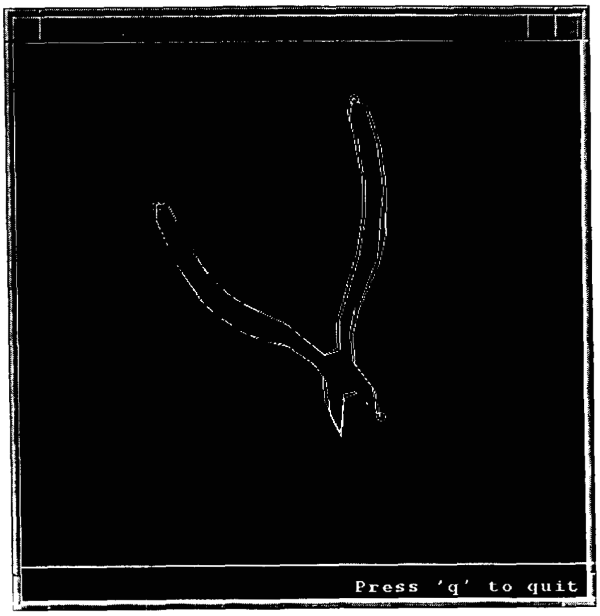


Fig. 1f)

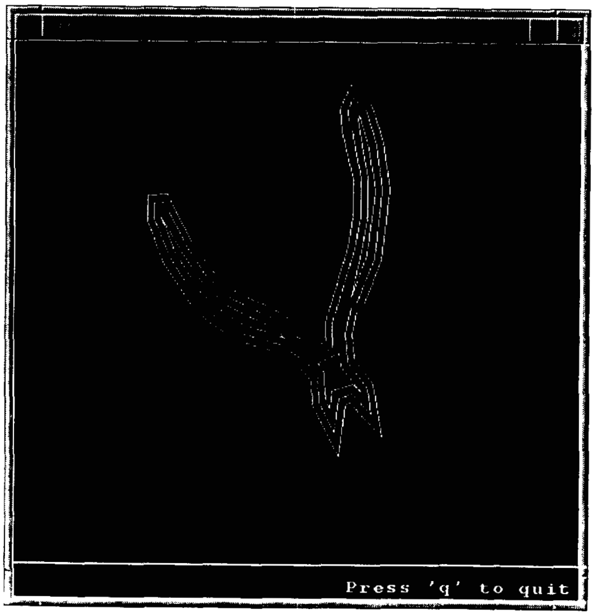


Fig. 1g)

Fig. 1: Main steps in recognition (a-g).

a) edge detection

b) curve extraction

c) curve segmentation

d) robust (larger circles)/non-robust point generation,

e) basis selection

f) hash table results

g) verification

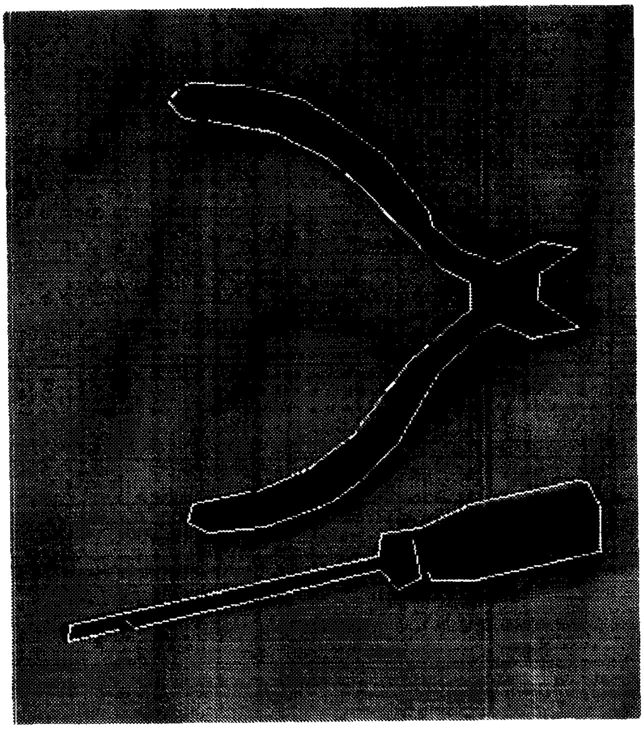


Fig. 2: Recognition results for two unoccluded objccts.

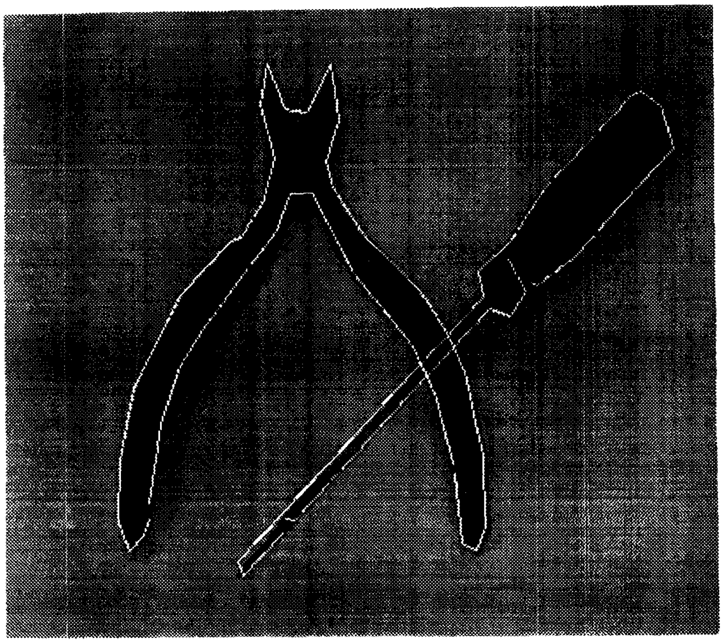


Fig. 3: Recognition results for two occluded ol jects.