Experiments in Occluded Parts Recognition Using the Generalized Hough Transform

by

J. L. Turney, T. N. Mudge, R. A. Volz and M. D. Diamond
Robotics Research Laboratory
Department of Electrical and Computer Engineering
University of Michigan
Ann Arbor, MI 48109
313/764-4343

ABSTRACT

When recognizing objects within a scene the input image is usually reduced to an edge image to simplify the recognition process. If the objects within the scene are occluded the edges of any one object are broken into a series of edge segments. In the experiments that will be described in our paper we use a collection of objects about which certain prior knowledge is assumed. In particular, edge templates of each object are available and the complete set of objects that can appear in any scene is known. Recognition is based on template matching. The difficulty with template matching is that it is time consuming and generates many false matching positions. By applying a generalized Hough scheme and constraining the matching process an efficient recognition algorithm is developed--fewer false matches occur. In addition, by appropriately weighting the template points so that the correlation between the edge template of the desired object and the edge image of the other undesired objects is small, one achieves a feasible technique for recognizing occluded objects. The weights used are based on prior knowledge that can be derived from the set of all objects. Also discussed is the concept of automatically calculating the prior information, such as the template and the template weightings, from a CAD data-base. Experiments have been performed in the two dimensional recognition of flat objects and the results will be presented.

INTRODUCTION

As solid state cameras and other hardware for computer vision have become less expensive, it has become reasonable to employ a vision system in many industrial inspection applications. One particular application where vision has been applied is in the problem of part recognition. In this application a vision system is employed to determine the orientation and position of industrial parts.

If the objects in a scene are nonoverlapping then it is often possible to employ a simple vision system such as the SRI vision module [Gleason]. With the SRI module one first determines the non-occluded edge boundary of an object and then recognizes the object using measurements of its silhouette such as area, moments, length of perimeter and other easily computable features that distinguish the object from a finite set of other possible objects. This finite set will be referred to as the domain set. Each object present in a scene is assumed to be a member of the domain set. The features of each object in the domain set are precalculated and stored together as an entry in a library of objects. A simple decision tree is used to determine which object is being viewed.

There is a major drawback to this approach, however. When the object of interest is occluded (i.e., it is overlaid by or overlays other objects, or lies partially outside the boundary of the scene) the area, moments, etc. observed for the object no longer aid in recognition. The approach embodied in the SRI module has not proven readily extensible to the recognition of occluded objects.

Although the boundary of an occluded object may be broken or may appear to merge with the boundary of another object, there is still a large amount of useful information in the boundary. If a object has a smooth surface with little variation in surface depth, its edges contain much of the information about the object. Furthermore, edges of a object are easier to work with than surface intensity and more quickly and readily obtained than other information such as surface depth. Therefore, it is still reasonable to base the recognition of an occluded object on the edges of the object.

One possible method of recognition is to match a prestored edge template to the object's edge in the image. In an industrial application, one is working in a restricted domain of objects. One knows exactly which objects to expect in the scene, and therefore the task of recognition should be much simpler than recognition of objects in a natural scene. It is reasonable to store an edge template for each view of each object expected in the scene, where the template could consists of a chain of edge points which form the edge boundary of the object.

In normal template matching an edge template is moved across the edge image. If the template overlaps any edge points in the image, the number of points overlapped is recorded in a two dimensional array, which will be referred to as the accumulator array. The accumulator array is incremented at the same position as the center of the template (see Fig. 1). This approach of moving the template across the image must be repeated for each orientation of the template. In this method of template matching the accumulator at the center of the template is incremented for each overlap of the template with the edge image. There are two difficulties with this form of edge matching. First, the approach of moving the entire template for all orientations across the image to check for matches is time consuming. Second, specifying a match simply because the template overlapped the image edge point allows too many false matches to be recorded in the accumulator array. When there is no occlusion (ignoring noise) the location with the largest accumulated value will be the location of the center of the object. However, when there is occlusion the location with the largest accumulated value is no longer guaranteed to be the correct location. When the object is occluded it becomes essential to reduce the possibility of false matches.

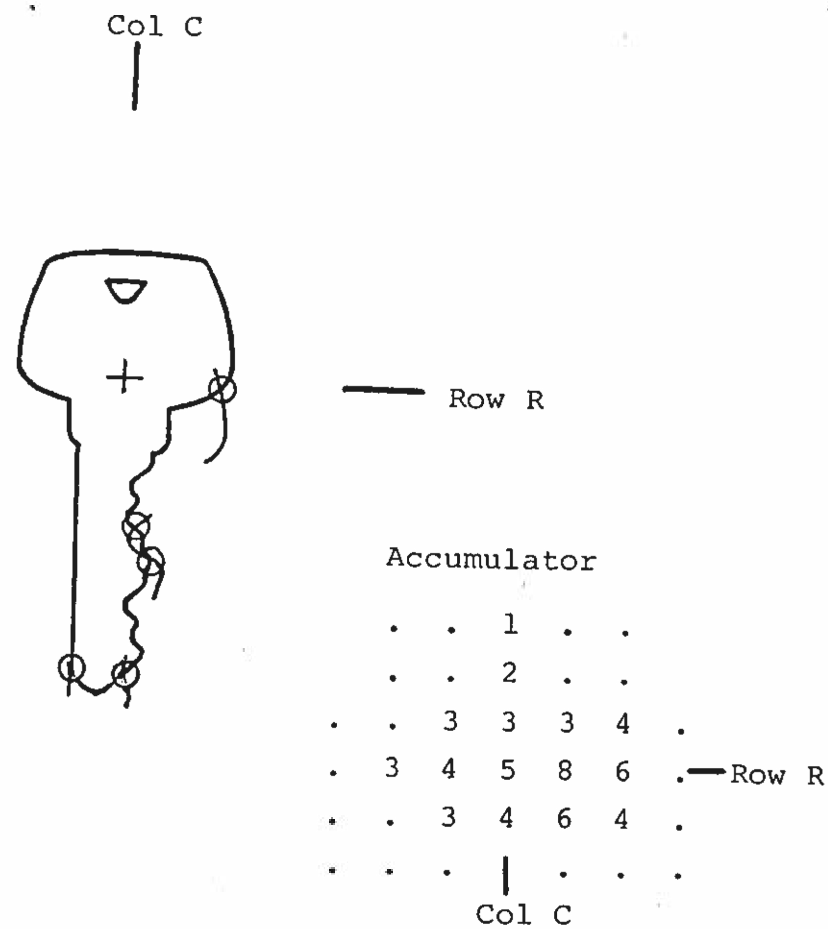

Figure 1. The number of points that overlap are stored in the accumulator.

THE GENERALIZED HOUGH TRANSFORM APPROACH

The first objection to template matching is that it can be time consuming, however, employing the generalized Hough transform [Sklansky] [Ballard], an efficient form of template matching, can alleviate this problem. The generalized Hough "inverts" the template matching strategy. Instead of starting with a template point and checking for matches the matching process starts with the edge point and employs table lookup to determine which template points would match the edge point. Rather than moving a template across an image and recording the matches in the accumulator, the generalized Hough transform records vector displacements from template points to the center of the template. Upon encountering an image edge point, the displacement vectors are attached to the image edge point and potential locations for the template center are located by the vectors. These potential locations are incremented in the accumulator array (see Fig. 2). This process represents a more efficient form of template matching, since all the template points that the edge point would have matched in the normal process of template matching are recorded at one time.

The second objection to normal template matching, that of generating too many false matches, can be dealt with in two ways. One is by allowing a match to be recorded in the accumulator array only if information in the neighborhood around the template point matches corresponding information around the image edge point which it overlays. This restricts matches and increases the probability that a correct match is recorded. One simple method is to allow a

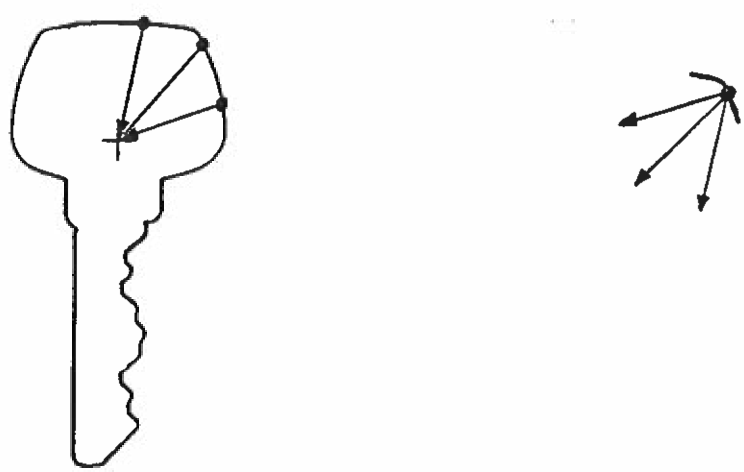

Figure 2. Displacement vectors from template points to template center are determined and then later used to locate potential template centers.

match if the template point has the same slope as the image point. The slope of a point is determined during edge detection and therefore incorporates more information than the slope between adjoining pixels which is restricted to increments of $\frac{\pi}{4}$. This is the approach used by Ballard in his version of the generalized Hough transform. Many other methods are feasible. For example, one can allow matches on the basis of the sum of the squares of differences between a template segment and the image segment, where the template segment and the image segment are represented in intrinsic coordinates such as theta-arclength space. In all these cases fewer contributions are vectored to the accumulator array, and hence fewer false peaks occur.

Another way of deemphasizing false peaks is to weight template points. When a weighted template segment matches a image edge segment the contribution vectored out is not a constant value but rather a weight preassigned to the template segment. In this way matches between the template and the desired image edge can be emphasized and those between the template and incorrect image edges can be suppressed. This weighting amounts to a method of automatic feature selection. The "salient features" of the desired object are determined in terms of weighted template segments (see Fig. 3).

The template weights can be adjusted as follows. The template of object A is matched against the edge of object B. The accumulator array records the number of matches at a location and in addition the specific template segments of object A which contribute to each match in the accumulator. A large number of matches recorded at one location result in a peak. An optimization approach is used to minimize all peaks by adjusting the value of the segment weights for object A.

A simplex algorithm can be employed to determine the weighting scheme. A new constraint on the template weights can be added for each peak in the accumulator. This results in too many constraints; even a template of modest size yields a large number of match points and hence accumulator peaks. A suboptimal scheme has been developed.

The object edge templates can either be extracted from real images or may be determined from an CAD data base where the object is represented in terms of cubic splines or polygon surfaces. Stable positions are determined for the objects and edge templates are determined once a view is decided [Volz, Mudge, Gal]. These templates are then assigned weights using the algorithm mentioned above, where each object's edge template is taken as the template to be weighted and the other objects are taken to be edge images. By adjusting the weights, each object's template is made as "orthogonal" as possible to the other object's templates, i.e., the weighted template of one object will poorly match any other object's edge image.

EXPERIMENTAL RESULTS

Fig. 4 displays a set of templates of two similar keys. The key's edges were extracted using a Frei and Chen edge detector, thinned and then linked.

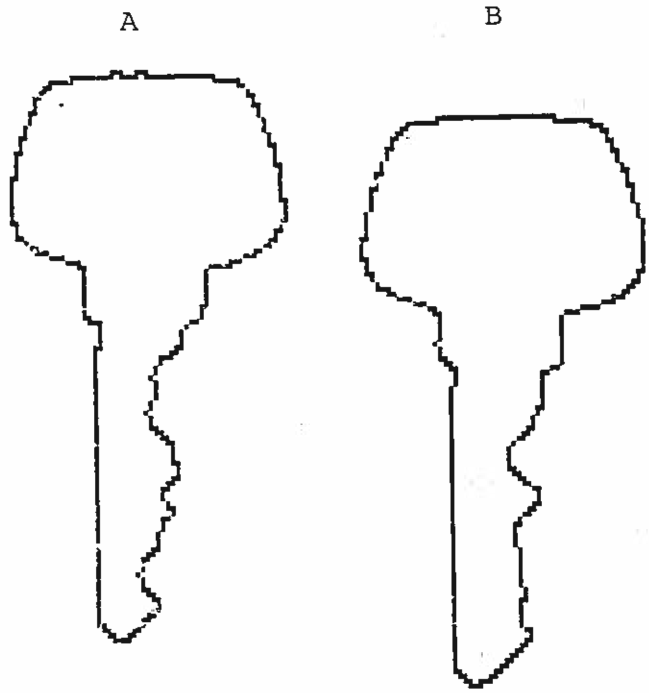

Figure 4. Two key templates.

Fig. 5 shows the adjusted weights obtained from the automatic weighting algorithm when template A is applied to image B. In other words, by choosing the weights of A as shown in Fig. 5 (the darker points correspond to more heavily weighted template points), matching template A to image B will yield very small peaks. Matching template A to image A will still yield a high peak. Image B is shown for comparison. It is obvious that the segments where A most differs from B, the notches of the keys, are the most heavily weighted, as would be expected.

Fig. 6 shows keys A and B occluded by other keys. Fig. 7 shows the accumulator array after matching B to the image. Fig. 8 shows the best match of key B to the image, i.e., its location. Fig. 9 shows the best match of key A to the image, i.e., its location.

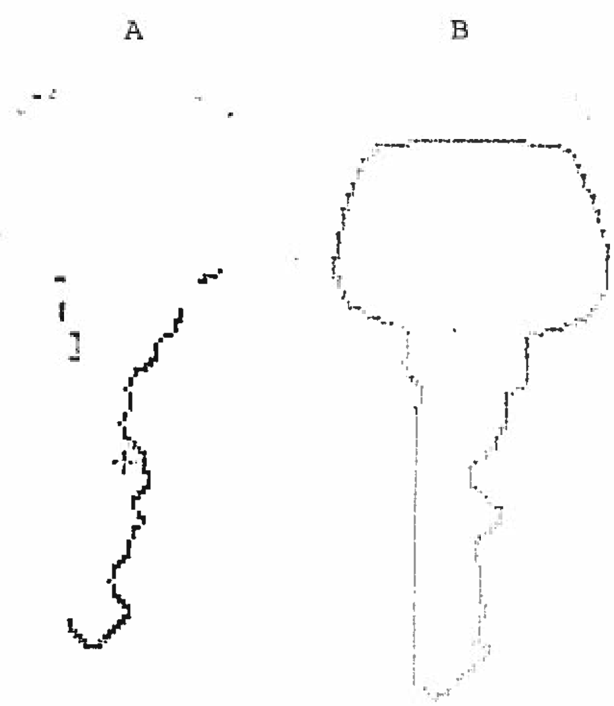

Figure 5. Weights are determined automatically.

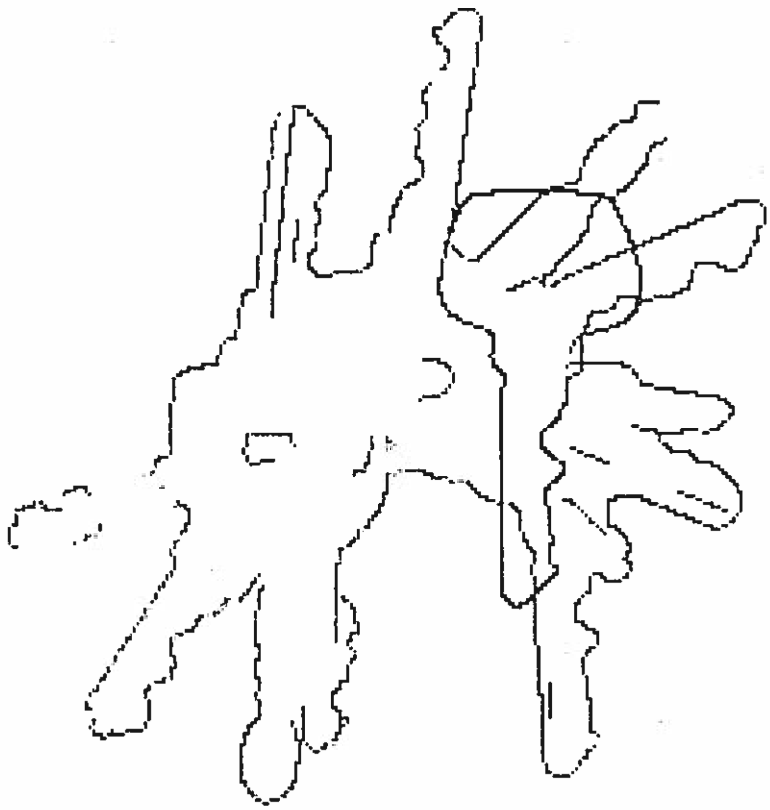

Figure 6. Keys are occluded.

Figure 7. Accumulator array after matching process.

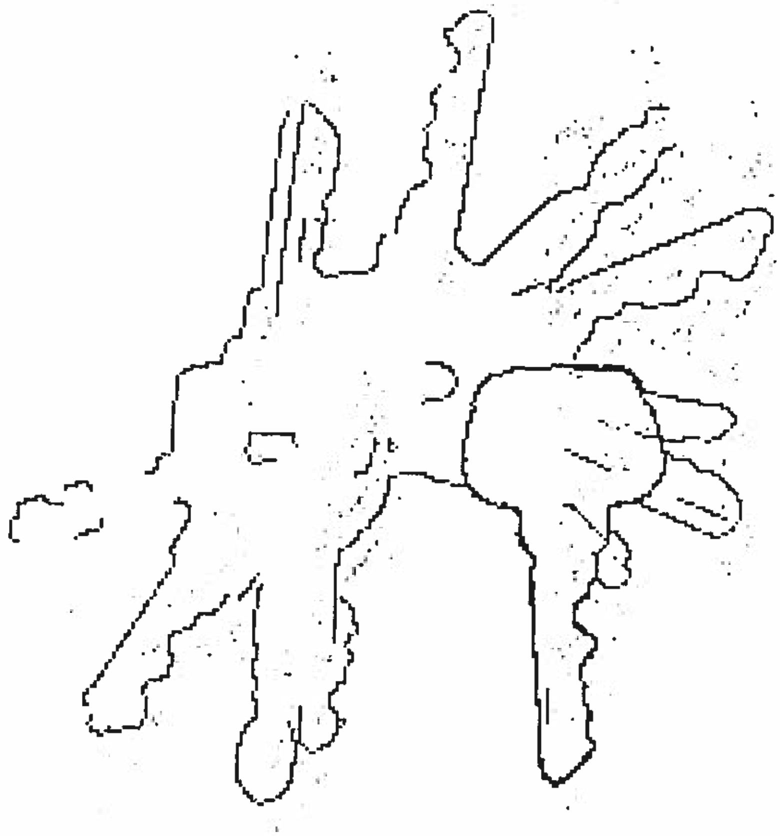

Figure 8. Best match for key B.

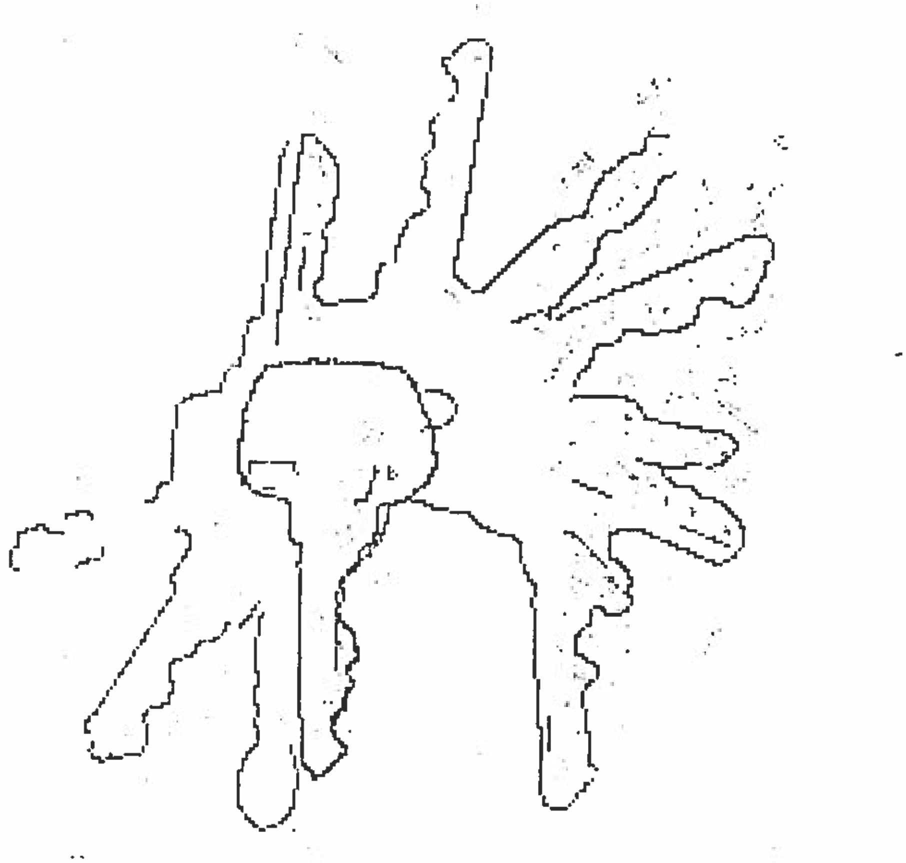

Figure 9. Best match for key *A*.

REFERENCES

[Ballard]
D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, No. 2, 1981, pp. 111-122.

[Gleason]
G. J. Gleason, "Vision Module Development," Ninth Report, NSF Grants APR75-13074 and DAR78-27128, SRI Projects 4391 and 8487, Stanford Research Institute, Menlo Park, CA, August 1979, pp. 9-16.

[Sklansky]
J. Sklansky, "On the Hough Technique for Curve Detection," *IEEE Transactions on Computers*, Vol. C-27, No. 10, October 1978, pp. 923-926.

[Volz, Mudge, Gal]
"Using Ada as a Robot System Programming Language," *Proceedings of the 13-th International Symposium on Industrial Robots and Robots 7 Conference*, Chicago, April 1983, pp. 12-42--12-57.