RSD-TR-20-83

RECOGNIZING PARTIALLY OCCLUDED PARTS

J. L. Turney
T. N. Mudge
R. A. Volz

Department of Electrical and Computer Engineering

The University of Michigan

Ann Arbor, Michigan 48107

December 1983

CENTER FOR ROBOTICS AND INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109

This work was supported in part by the Zimmer Foundation and in part by the Air Force Office of Scientific Research under contract F49620-82-C-0089

ABSTRACT

The problem of recognizing an object from a partially occluded boundary image is considered and the concept of saliency of a boundary segment introduced. Saliency measures the extent to which the boundary segment distinguishes the object to which it belongs from other objects which might be present. An algorithm is presented which optimally determines the saliency of boundary segments of one object with respect to those of a set of other objects. An efficient template matching algorithm using templates weighted by boundary segment saliency is then presented and employed to recognize partially occluded parts. The results of these experiments illustrate the effectiveness of the new technique.

Index terms--Occluded parts, saliency, weighted template matching, Hough transform, least squares.

TABLE OF CONTENTS

1. INTRODUCTION

The problem of recognizing partially occluded parts is of considerable interest in the field of industrial automation. While it is possible to employ shakers, conveyor belts and custom machinery to separate, palletize or otherwise prearrange the parts for easy recognition, a vision system which can recognize the parts even though they may be partially occluded and in random positions is much more flexible. Attempting to match a model, or template, of the boundary of the part with the boundary of the image has been suggested for this problem [Per78, Per80]: if a position of the template can be found for which a large number of its boundary pixels match those of the image, it may be concluded that the part has been found. Unfortunately, in its basic form, this procedure requires an excessive amount of computation and can produce a number of false matches resulting in unreliable recognition. This paper presents an efficient method of template matching which is based on matching subtemplates and in which the subtemplates are optimally weighted to decorrelate those of the object being sought from those derived from other objects which might be present.

In a typical industrial scene, the parts may be intermixed, partially occluded, and of unknown pose (position and orientation); however, the types of parts that will be present are almost always known *a priori*. We assume that there is available a database describing the geometry of all the parts that could conceivably appear in the scene. This database could be constructed during a "learning" phase, or it could be constructed by a Computer Aided Design (CAD) system as a byproduct of the design process. The latter method is likely to predominate in the future as CAD systems become more sophisticated and capable of being integrated into a "factory-of-the-future" environment. Indeed, work in this direction has already begun, for example, silhouettes and grasp-points for parts can automatically be extracted from CAD data and organized into a suitable database for part handling [Bau81, Bau82, VMG83, WWV82].

>From the database of part-geometries, boundary templates for every stable position of every part are determined, and a set of subtemplates which most differentiate the parts is identified. The subtemplates in this set are called *salient* features. Salient features depend on the set of parts being considered. To illustrate this point consider Figure 1. Figure 1(a) shows the salient features when parts A and B comprise the set of parts, while Figure 1(b) shows the salient features when parts A and C comprise the set of parts (salient features are shown as heavy lines). Those features of A's boundary which most distinguish it are clearly dependent on the set of parts, i.e., on the context in which A could potentially be viewed. To make this concept more precise a measure of *saliency* is introduced to denote the distinctiveness of a subtemplate. The saliency of a subtemplate corresponds to the optimal (in the least squares sense) weighting mentioned above and is defined on the closed interval [0,1] (see Section 4). By giving saliency a continuous range, the pose of a part that is partially occluded can be accurately determined if the visible portion of the part matches some set of subtemplates with enough combined saliency.

Preliminary versions of some of the concepts used in the approach to partially occluded part recognition developed in this paper were first reported in [TMV83a, TMV83b]. For the most part, the approach is most suitable for fairly flat parts of known scale. However, work in progress suggests that the technique is more general and may be extended to three dimensions, in which case the concept of saliency still applies, but the subtemplates are no longer boundary segments, rather surface subregions. Even with the above restrictions the technique applies to a wide range of industrial applications.

The paper is organized as follows. Section 2 gives the background for the techniques to be employed including a brief summary of the Hough transform which motivated portions of this work. Section 3 develops an efficient template matching scheme based on an extension of the Hough transform that uses subtemplates. Section 4 addresses the problem of determining the saliency of subtemplates for a set of parts.

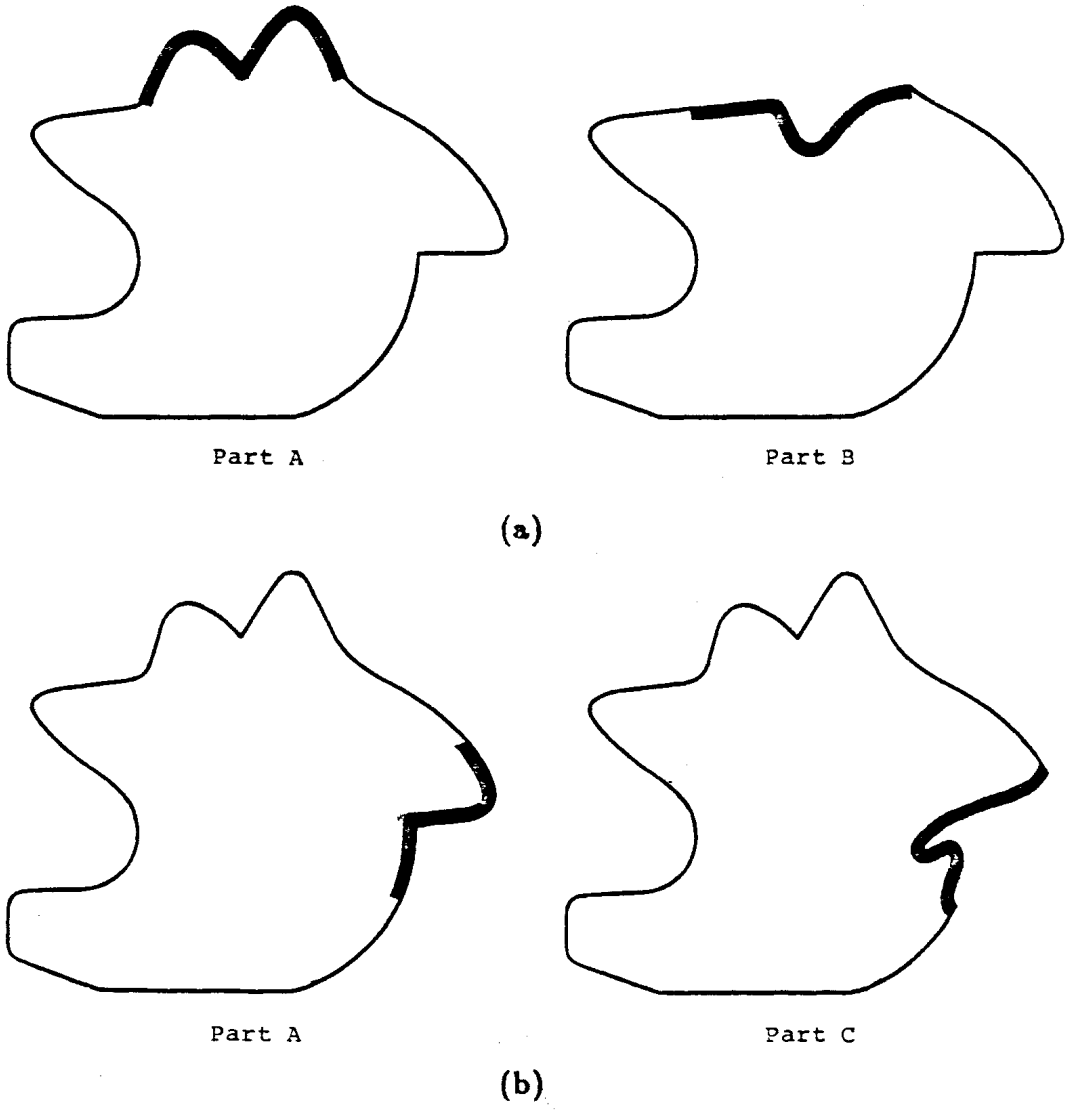


Figure 1. Context Dependency of Salient Features.

Section 5 presents some experimental results which illustrate the effectiveness of using saliency for recognizing partially occluded parts. Section 6 concludes with a discussion on further directions for research. In particular proposals for improving recognition times.

2. BACKGROUND

The approach to recognizing partially occluded parts developed in this paper works with the shape of their boundaries. In the context of the intended application--recognizing and locating fairly flat machined parts--this choice retains that aspect of the parts which conveys a significant amount of information useful for recognition; at the same time it allows us to work with a relatively compact boundary representation of the scene.

The scene in which the parts appear will be referred to as the application scene. It is digitized into a two-dimensional array of pixels, $I(1 \leq x \leq M, 1 \leq y \leq N$: x and y integer), which can take on values from a set of gray-levels. The boundary representation of the parts in the application scene is called the boundary image. Although its machine representation is usually in a compact form, such as a linked list, it can be thought of as a function, $B(x,y)$, which is defined on the same domain as I and which has the value one at boundary pixels and zero elsewhere. Boundary pixels and their slope angles[1] are extracted from the application scene and thinned so that only the strongest boundary pixels which maintain boundary continuity remain [DeC83]. The output of the thinning operation is the boundary image. It is the boundary image against which the templates are matched to locate the part in the application scene. For this discussion, a template, $T(x,y)$, can also be thought of as a binary valued function defined similarly to B, but with a smaller domain, $(1 \leq x \leq m, 1 \leq y \leq n$: $m << M$ and $n << N$). Furthermore, template pixels will be considered to be those pixels that are, strictly speaking, boundary pixels of the template, i.e., where $T(x,y)$ has the value one.

As mentioned earlier, template matching is performed using an extension of the Hough transform. The following subsection gives a brief summary of the Hough transform and its relationship to normal template matching.

---

[1] A pixel's slope angle is the angle of the tangent to the boundary at that pixel.

2.1. The Hough Transform and Template Matching

The Hough transform was originally designed to identify colinear subsets of points within larger sets of points [Hou62]. Subsequently, a number of significant improvements have been made to the original transform. In particular, Merlin and Farber generalized it to identify arbitrary sets of points [MeF75]. We begin by describing how this version of the transform can be used to identify the position of "best match" of a template $T(x,y)$ to a boundary image $B(x,y)$.

Assume that $B(x,y)$ is produced from the application scene in the manner outlined above. Associated with each pixel in B is an accumulator whose purpose is to record template matches. This two-dimensional array of accumulators is assumed to be initially zero. A reference point is selected that is fixed with respect to T; we used T's centroid because it was useful in a related project where a robot was required to grasp the part, but any appropriate reference point may be used. A template, $T^*(x,y)$, is formed by rotating T 180^0 about its centroid. The centroid of T^* is placed at a boundary pixel in B and every accumulator that coincides with a pixel of T^* is incremented by one (see Figure 2). This procedure is repeated for each boundary pixel in B. When all the boundary pixels have been visited, the accumulator with the largest value identifies the position of the centroid of template T that gives the best match.

Sklansky demonstrated in [Skl78] that the function produced in the accumulator array by the above procedure is the convolution of B with T^* which is simply the cross-correlation or template matching of T with B. Since the Hough transform is equivalent to template matching it performs the function of a matched filter and therefore is the optimum filter when B is corrupted by additive white Gaussian noise under a wide variety of criteria: signal-to-noise, likelihood ratio, and inverse probability (see [BlZ76] Chapter 7). However, none of these criteria implies that it is optimal for recognizing one shape in the presence of others (see the discussion on normalized

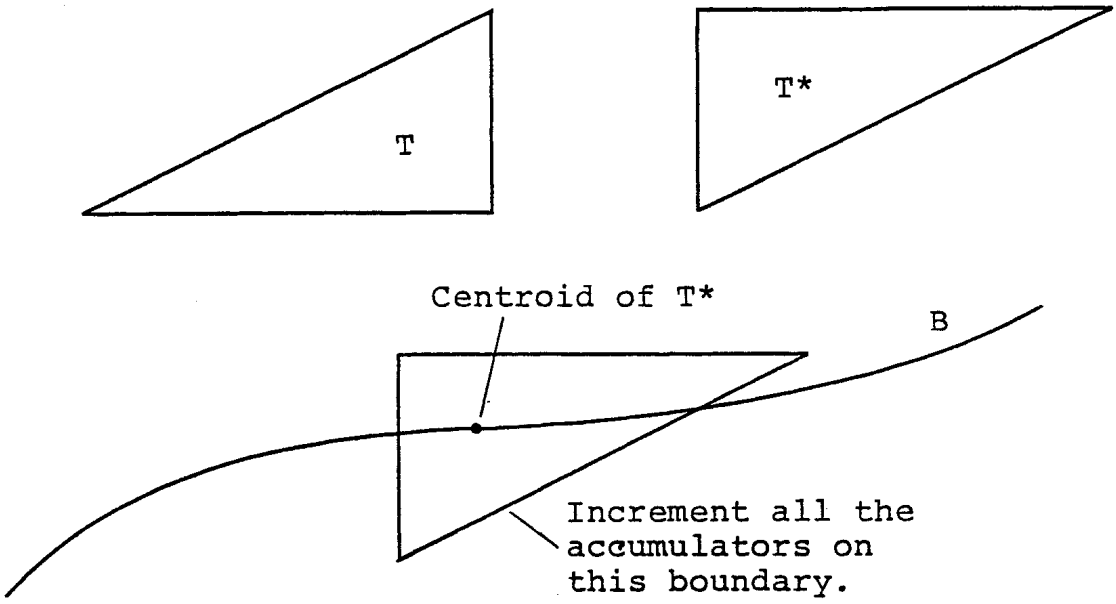
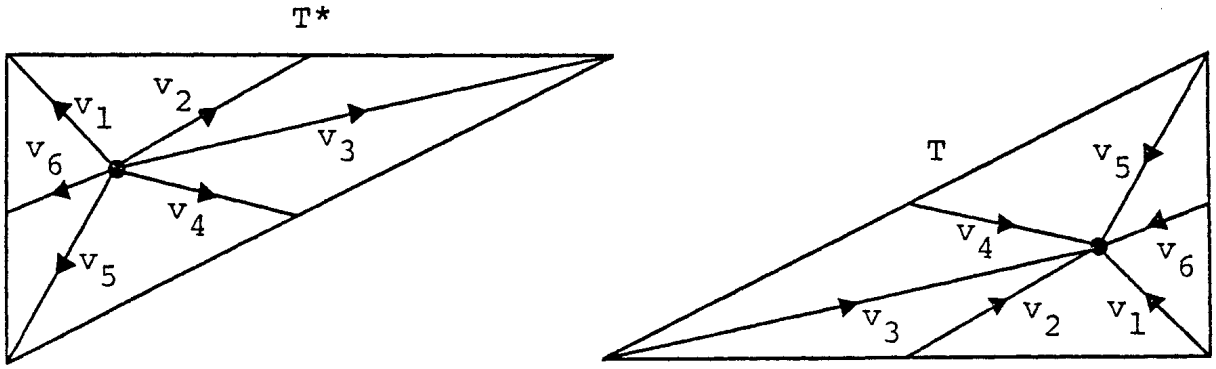


Figure 2. Hough Transform.

correlation in Rosenfeld and Kak [RoK76]), suggesting that template matching is by no means sufficient for recognizing partially occluded parts.

In [Skl78] it is also suggested that the Hough transform could be improved if local properties of the boundary pixels were used to restrict the set of accumulators that are incremented. Less computation would be needed and fewer false matches would occur, i.e., the "Q" of the filter would be increased. Ballard in [Bal81] presents an improved version of the Hough transform in which the slope of the boundary pixels is used to restrict the set of accumulators that are incremented. This is achieved by performing the Hough transform in the following way. Create a set of vectors, $V = \{\mathbf{v}_i, i=1,...,|T|\}$, from each pixel t_i on the boundary of T to T's centroid. When the vectors in V are all originated at the same point, they trace the rotated template T^* (see Figure 3). Therefore, the above procedure of Merlin and Farber can be performed equivalently by originating the vectors at a boundary image pixel and incrementing the accumulators pointed to by each of the vectors; this step is then repeated



$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$

Figure 3. Creating T' from V.

for every pixel on the boundary image. The equivalence between the Hough transform and template matching can be seen if it is considered that originating a vector, $\mathbf{v}_i$, at a boundary pixel, b_j, is, in effect, proposing b_j as a match for the template pixel t_i and voting for that match by incrementing the accumulator pointed to by $\mathbf{v}_i$. Restrictions can be applied by simply using subsets of V at each boundary point. In Ballard's version of the Hough transform the slope of the boundary pixel is used to select only those $\mathbf{v}_i$ whose corresponding t_i have the same slope[2] (see Figure 4). This approach has already been experimented with some success in an industrial application [ACM83]. In the next section the idea of local restriction is taken one stage further by requiring that a pixel's neighboring pixels satisfy a certain configuration, or subtemplate.

The computational complexity of the Hough transform (and hence template matching) can increase rapidly if it is necessary to deal with variations in orientation

[2] In the original description of the algorithm the perpendicular to the slope—the gradient—was used.

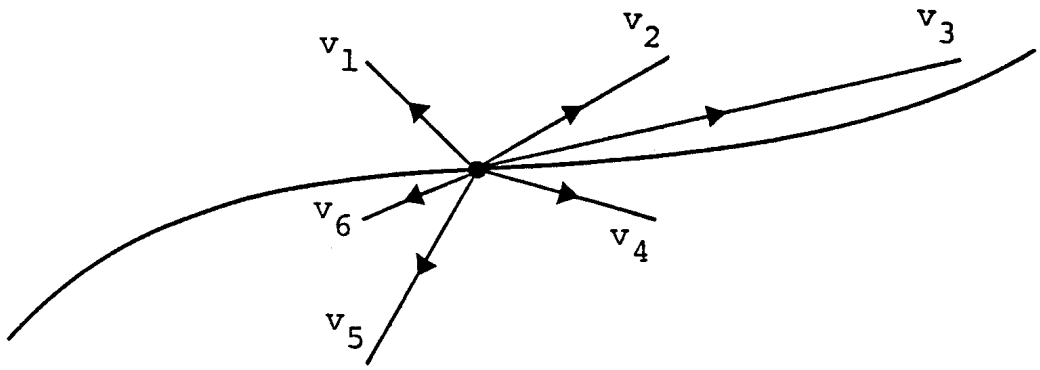


Figure 4. Slope Restricted Hough Transform.

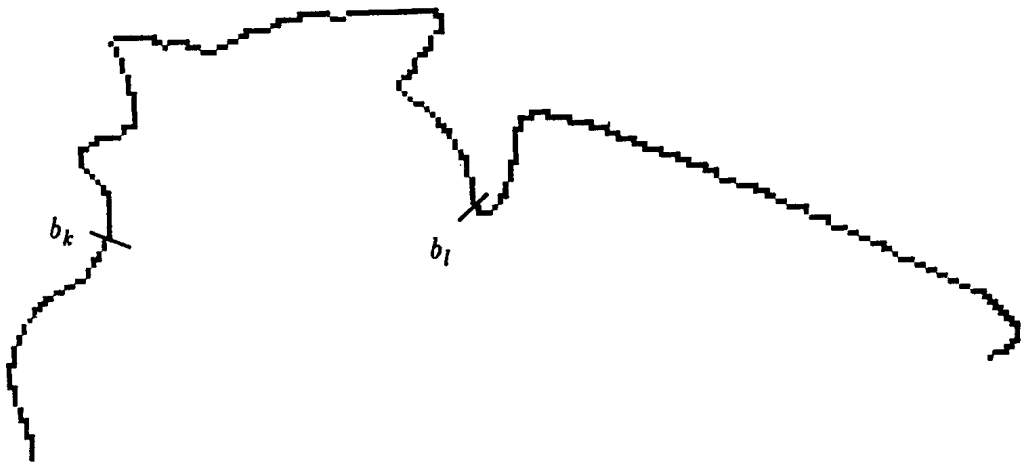
and scale. For example, the above procedure for the Hough transform assumes a fixed orientation of T' (or T). To account for variations in orientation the complete procedure must be repeated for every orientation to be distinguished. In other words, if it is required to distinguish between orientations that are 1^{0} apart, the procedure must be repeated 360 times, resulting in 360 accumulator arrays. The best match would then be identified by the accumulator with the largest value in all of the 360 arrays. A similar complexity arises if the scale of the parts are not known in advance--all sizes of T to be distinguished must be tried.

The next section presents a technique based on an extension of the Hough transform which uses subtemplates. This technique reduces the complexity that arises from the need to consider variations in orientation. In addition, and more importantly, subtemplates provide convenient boundary features that can receive salient weightings (see Figure 1).

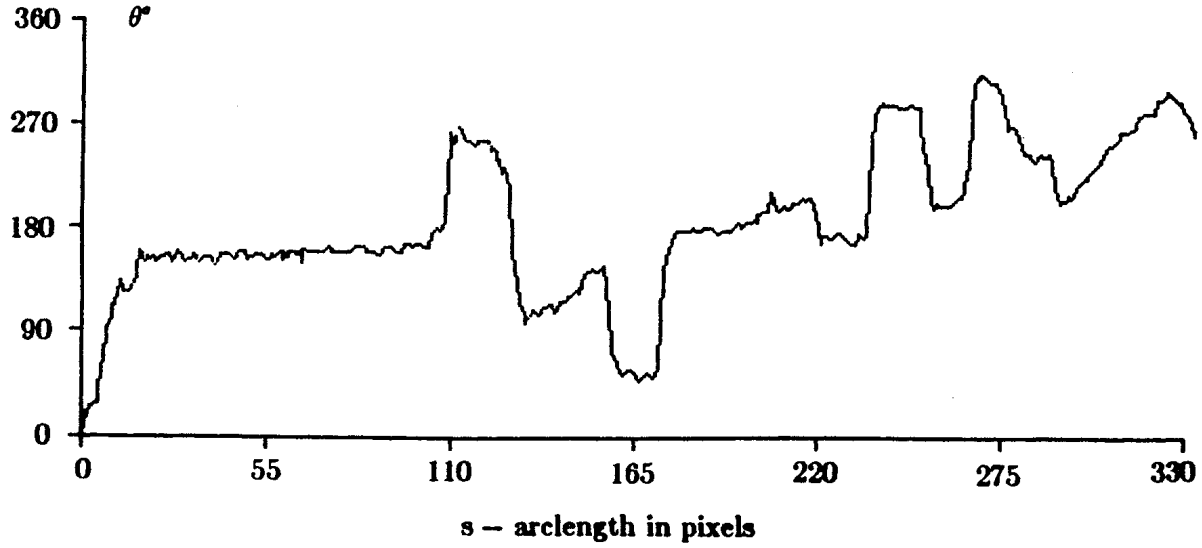
3. SUBTEMPLATE MATCHING

Let T be a template of the boundary of a part. A set, τ, of overlapping boundary subtemplates, τ_i, are created from template T. Each subtemplate τ_i has an associated vector, $\mathbf{v}_i$, that points to the location of the centroid of T. The Hough transform can now be modified by requiring that a segment of the boundary image centered on a particular pixel, b_j, match τ_i before $\mathbf{v}_i$ can be originated at b_j. In addition, one can introduce a coefficient of match between τ_i and the boundary segment (see below) and use this to allow the accumulator pointed to by $\mathbf{v}_i$ to be incremented by a fractional value. In our experiments each τ_i was 20 pixels in length and $|T|/2$ subtemplates were created for each template T, i.e., each template pixel was part of 10 subtemplates. The choice of the length and the number of subtemplates is application dependent.

In the process of matching the subtemplates with the boundary it is important to place the subtemplate in the correct orientation. This can be achieved efficiently if the subtemplate and the boundary image segments are represented in their *intrinsic* coordinate system in which the angle of slope, θ, and the arclength along the boundary, s, act as coordinates for pixels [BaP71, Per78, Per80]. This will be referred to as the θ–s representation of a boundary. The θ–s representation allows subtemplates and boundary image segments to be characterized by functions of the form $\theta(s)$. An important property of these intrinsic functions is that a change in orientation, θ_c, of a boundary in x–y space corresponds to simply adding θ_c to $\theta(s)$ in θ–s space. The slope θ is determined as mentioned before during boundary extraction, while s is determined by measuring the distance in pixels along the boundary image after thinning (see Section 2). Figure 5(a) shows a boundary image containing a partially occluded part between pixels b_k and b_l. Figure 5(b) displays its θ–s representation. Figure 6(a) shows the complete template of the partially occluded part in Figure 5; a subtemplate is shown with a heavy line. Figure 6(b) shows the θ–s representation of the template.

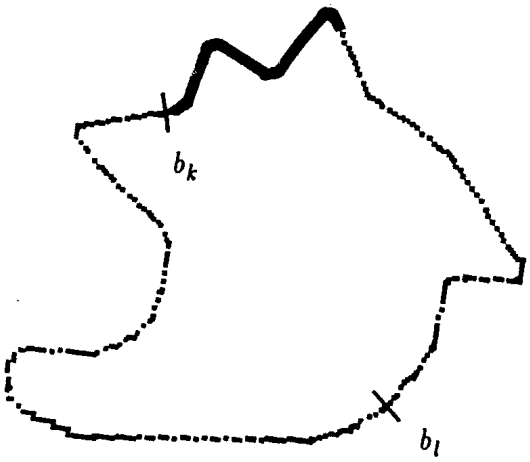


(a)

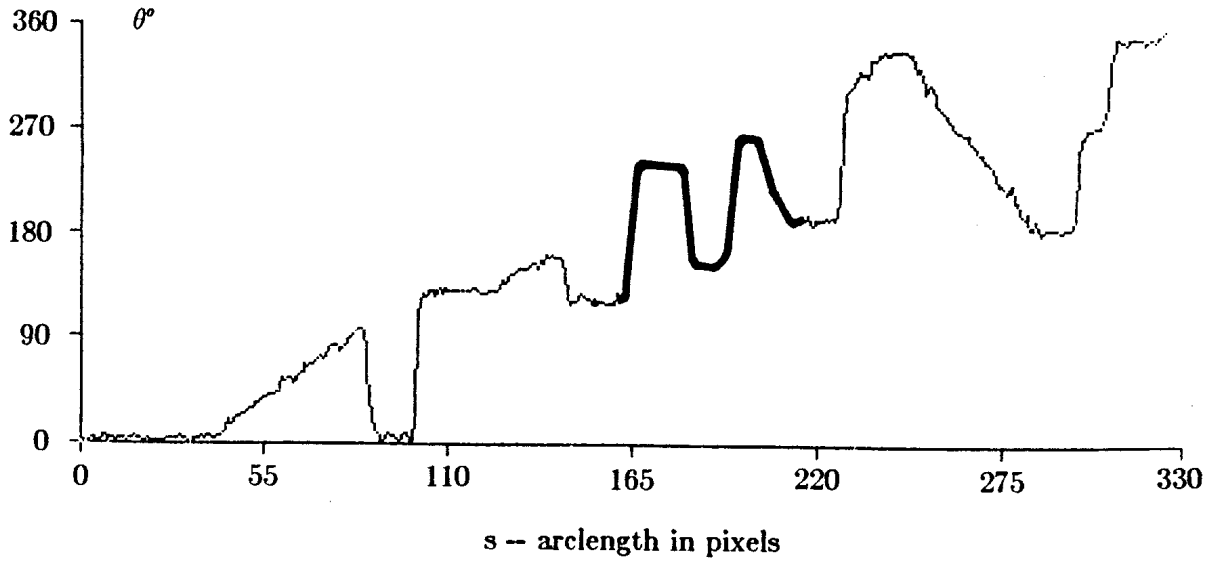


(b)

Figure 5. A Boundary Image and Its θ–s Representation.



(a)



(b)

Figure 6. A Template and Its θ–s Representation.

Matching starts by choosing a subtemplate, τ_i, from the template boundary and attempting to match it against an equal length segment of the boundary image, β_j,

centered at pixel b_j. The attempted match is performed in θ–s space at regularly spaced boundary image pixels, b_j, j=1,...,$|\beta|$, where $|\beta|$ is the number of boundary segments. In our experiments the spacing was every other boundary image pixel; as with subtemplate length and number, spacing is application independent. Strictly speaking, since the matching is performed in θ–s space, $\theta_{\tau_i}(s)$ is matched against $\theta_{\beta_j}(s)$. Conceptually this can be thought of as moving $\theta_{\tau_i}(s)$ along the s-axis direction in the θ–s graph of the boundary image, $\theta_B(s)$, and performing a match at regularly spaced points (see Figure 7). The matching operation is repeated for a "representative" set of subtemplates for the part. In the experiments reported in Section 5 the representative set is taken to be all the subtemplates. However, preliminary results mentioned in the Conclusion suggest various strategies for selecting this representative set that significantly improve recognition times.

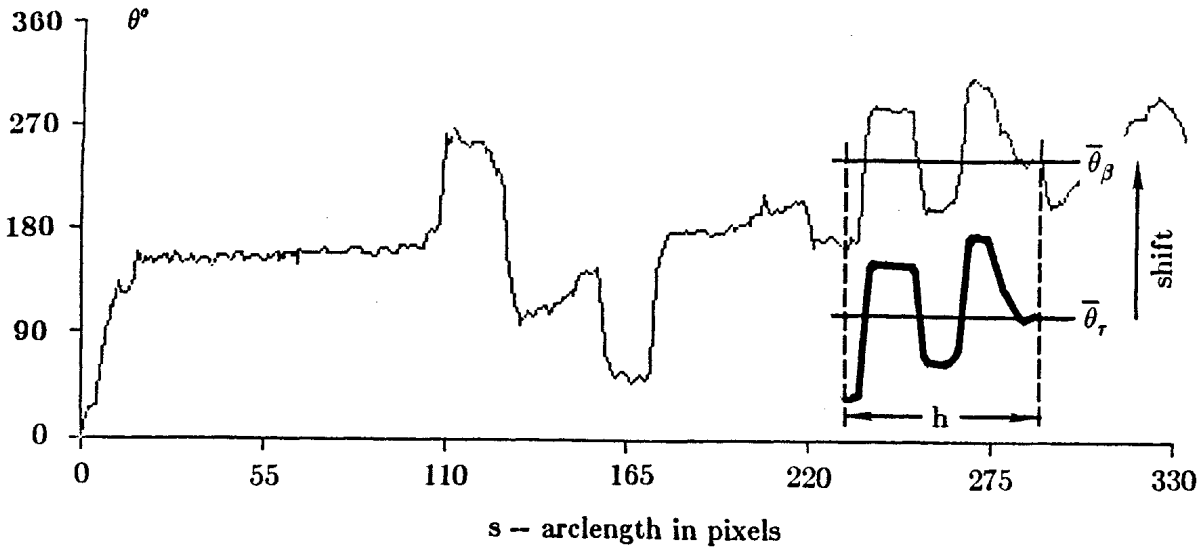


Figure 7. Matching in θ–s Space.

As noted earlier it is important to get the subtemplates in the correct orientation before matching. This is done as follows. The average angle of orientation, $\bar{\theta}_{r_i}$, for the subtemplate r_i is determined by averaging the slope angle of the pixels on the subtemplate, and the average angle of orientation, $\bar{\theta}_{\beta_j}$, for the boundary image segment β_j is determined by averaging the slope angles of the pixels on β_j. The difference between the average slope angles, $\bar{\theta}_{\beta_j} - \bar{\theta}_{r_i}$, is a measure of the difference in orientation between the boundary image segment and the subtemplate. The subtemplate is then shifted by $\bar{\theta}_{\beta_j} - \bar{\theta}_{r_i}$ to the same average angle as the boundary image segment before the two are compared (see Figure 7). The shift in θ corresponds to an angular rotation of the subtemplate to the average orientation of the boundary image segment. This correspondence between simply shifting in θ–s space and rotating in x–y space is a feature of the intrinsic coordinate functions, $\theta(s)$, noted earlier; in conjunction with subtemplates it reduces the complexity that arises from the need to consider variations in orientation. Figure 8 shows the corresponding match between the subtemplate and the boundary image segment in x–y space.

A measure of the mismatch between the rotated version of subtemplate, r_i, and the boundary image segment, β_j, is given by the quantity γ_{ij}, such that:

$$\gamma_{ij} = \sum_{p=1}^{h} [\theta_{\beta_j}(s_p) - \hat{\theta}_{r_i}(s_p)]^2$$

Where h is the number of pixels in r_i and β_j (recall, in our experiments h=20), and the term $\hat{\theta}_{r_i}$ is the θ–s representation of the subtemplate shifted in the θ direction by $\bar{\theta}_{\beta_j} - \bar{\theta}_{r_i}$.

Using γ_{ij}, we can define a *matching coefficient* which gives a measure of the match between a subtemplate and a boundary image segment. Let:

$$c_{ij} = \frac{1}{1+\gamma_{ij}}$$

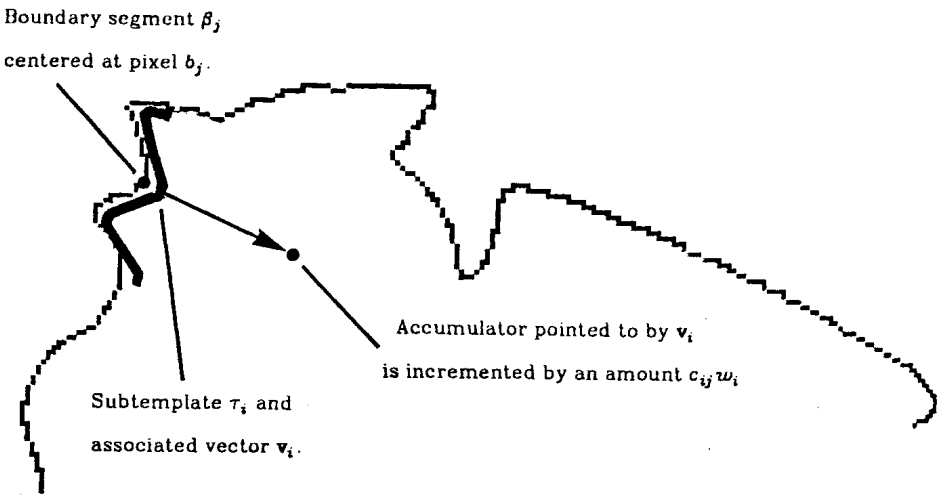


Figure 8. Matching in x–y Space.

The matching coefficient, c_{ij}, is used to adjust the contribution added to the accumulator during subtemplate matching, and it is defined in the above manner so that values close to 0 imply a very poor match and values close to 1 imply a very good match ($0 < c_{ij} \leq 1$).

In order to locate the potential centroid for template T (and hence the part that T represents), the vector $\mathbf{v}_i$ associated with subtemplate τ_i is rotated by the angle $\overline{\theta}_{\beta_j} - \overline{\theta}_{\tau_i}$ and is originated at the center of the boundary image segment, as shown in Figure 8. The accumulator pointed to by $\mathbf{v}_i$ is then incremented by the quantity $c_{ij} w_i$, where w_i is the saliency of τ_i mentioned in the Introduction (see also the next section). The result of matching a subtemplate τ_i with a boundary segment β_j is thus characterized by two factors: the similarity of τ_i and β_j (measured by c_{ij}), and the significance of finding a match for τ_i in recognizing the part from which τ_i is derived (measured by w_i).

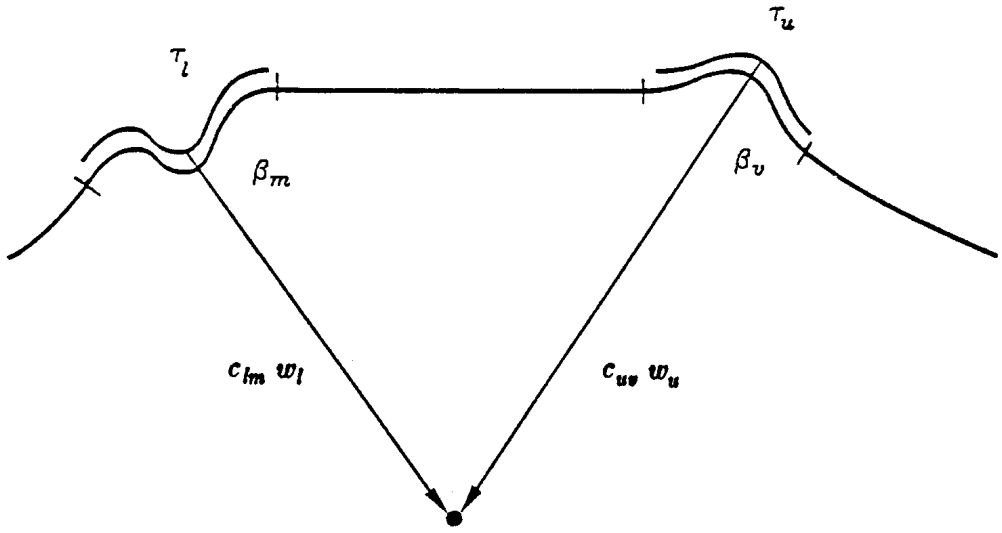
In addition, the angular difference, $\overline{\theta}_{\beta_j} - \overline{\theta}_{r_i}$, is appended to a linked list associated with the same pixel as the accumulator pointed to by $\mathbf{v}_i$. At the end of all the subtemplate matching, the list of angles associated with the pixel which is judged to be at T's centroid is averaged to estimate the orientation of T. As before, T's centroid is judged to be the pixel whose associated accumulator contains the greatest number.

4. DETERMINING THE SALIENCY OF SUBTEMPLATES

As mentioned before in Section 1, by properly weighting subtemplates emphasis can be given to those features which distinguish an object--the salient features. Thus, partially occluded parts can be recognized if features with enough saliency are exposed. This section develops an algorithm by which salient weights can be automatically assigned to the subtemplates.

The algorithm works off-line with the database of part-geometries, specifically the subtemplates derived from the part-geometries. It has the advantage that a new part-geometry can be introduced into the database and new weights can readily be assigned to all the subtemplates affected by the addition of the new part. The algorithm optimally adjusts the weights of subtemplates so that different templates correlate poorly with one another.

First, for simplicity, only two templates, A and B, will be considered. They may be derived from different parts or from different stable positions of the same part. Assume that template A consists of $|\tau|$ subtemplates, τ_i, and that each subtemplate is assigned a positive weight w_i. Assume, furthermore, that each τ_i is matched with each of the $|\beta|$ boundary segments, β_j, of B. Although A and B are both templates it is convenient for this explanation to consider A to be a template and B a boundary image derived from a (synthetic) application scene. If subtemplate τ_i of A matches a boundary segment β_j of B, a symbolic record of the match $c_{ij}\, w_i$ is stored at accumulator s pointed to by $\mathbf{v}_i$. The number s is a unique index into the accumulator array (for example, s might be defined by $s = N*row\_index + column\_index$ for an $M \times N$ array). The symbolic value stored at s is a variable w_i (to be determined) and a numerical value c_{ij}, the matching coefficient, derived through the subtemplate matching process described in the previous section. Figure 9 illustrates the contributions that accumulator s might received from matching subtemplates with boundary segments. Since template A has $|\tau|$ subtemplates, and B has $|\beta|$ subtemplates, $|\tau||\beta|$ matches are



Accumulator s contains

$$\cdots + c_{lm}\ w_l + c_{uv}\ w_u + \cdots$$

Figure 9. Computing Saliency.

attempted.

After all the subtemplate matching is completed s contains a linear function in $|\tau|$ variables, w_i, $i=1,\ldots,|\tau|$. Let this function be a_s, then:

$$a_s = \sum_{i=1}^{|\tau|} \sum_{j=1}^{|\beta|} \delta_{ij}^{s}\, c_{ij}\, w_i$$

Where δ_{ij}^{s} is a selector function that takes the value 1 if s receives a contribution from a match between subtemplate τ_i and boundary segment β_j; δ_{ij}^{s} takes the value 0 otherwise. The above equation can be written in matrix form as:

$$\mathbf{a} = D\mathbf{w}$$

Where the vector $\mathbf{w}$ has the elements w_i, and D is a $(M \times N) \times |\tau|$ matrix with nonnegative elements given by:

$$d_{si} = \sum_{j=1}^{|\beta|} \delta_{ij}^{s} c_{ij}$$

In practice D is very sparse.

As noted above, the objective of the weighting algorithm is to adjust the weights w_i so as to minimize the correlation of template A with boundary B. The approach taken is to minimize the sum of the squares of all of the a_s's. This results in the sub-templates that are most unlike others--the most salient ones--receiving the greatest weights. The problem may be stated as follows:

$$\text{Minimize:} \quad \sum_{s=1}^{M \times N} a_s^2 = \mathbf{a}^t \mathbf{a} = \mathbf{w}^t D^t D \mathbf{w}$$

$$\text{Subject to:} \quad \underline{1}^t \mathbf{w} = 1 \quad \text{and} \quad \mathbf{w} \geq \underline{0}$$

Where $\underline{1}$ is a $|\pi|$-dimensional vector of 1's and $\underline{0}$ is a $|\pi|$-dimensional vector of 0's. Given $\mathbf{w}$, a nonzero vector of weights, $k\mathbf{w}$ ($k>0$) will produce the same ordering in the accumulator array and hence the same solution to the part location problem. Accordingly, the constraint $\underline{1}^t \mathbf{w} = 1$ is used to provide a normalization of the weights.

The constraint $\underline{1}^t \mathbf{w} = 1$ is a hyperplane normal to the vector $\underline{1}$ and at a distance of $\sqrt{|\pi|}$ from the origin. The surfaces of constant value of the function $\mathbf{w}^t D^t D \mathbf{w}$ are hyperellipsoids. The constrained solution occurs for the value which makes the corresponding hyperellipse tangent to the constraining hyperplane. When the ratio of maximum to minimum eigenvalues of $D^t D$ is large (as is often the case in this application) and the major axis of the hyperellipsoid is skewed so as to be nearly parallel to the hyperplane, large negative and large positive weights can occur. Figure 10 shows a two-dimensional example in which positive and negative weights result--S is the solution point. In terms of the matched filter interpretation of Section 2 this amounts to creating a very high "Q" filter for the part. In other words, using the weights to recognize the part in an application scene results in the contents of the array of

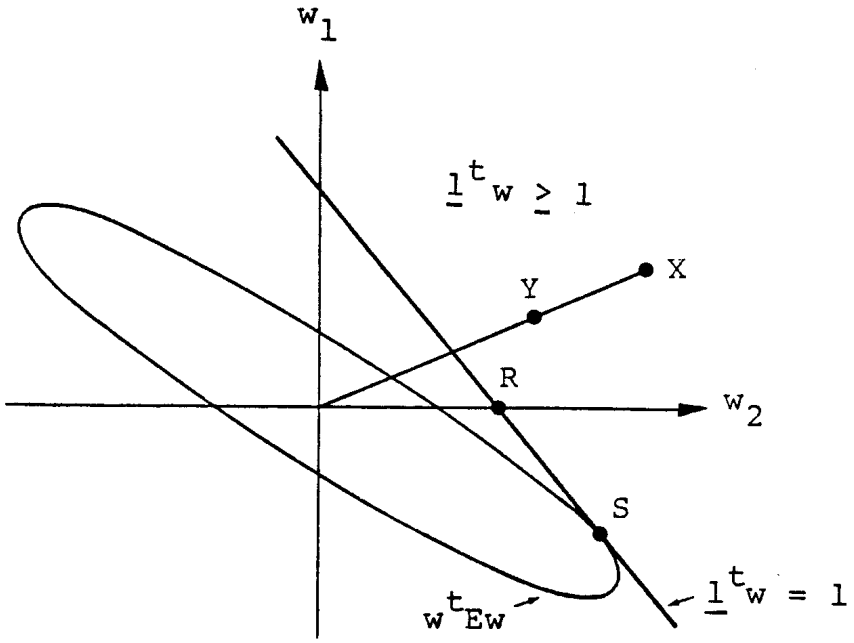


Figure 10. Minimal Solutions in Two Dimensions.

accumulators forming a sharp peak at the location of the part's centroid, if it is present. This narrow bandwidth is obtained in part by the weighted contributions to the accumulators canceling one another everywhere except close to the centroid. If the bandwidth of this peak is on the order of a single pixel there is a danger of missing the peak in practice due to noise and quantization (the weights are determined from noiseless images derived from the database of part-geometries). The constraint $\mathbf{w} \geq \underline{0}$ "detunes" the filter and prevents this from occurring. In terms of Figure 10, the solution point is moved to R.

It is convenient to restate the minimization problem in a slightly different form by letting E denote the product $D^t D$. E is then symmetric, and, although the D matrix is typically an extremely large sparse $(M \times N) \times |\tau|$ matrix (consider $M=N=256$), E is a much smaller $|\tau| \times |\tau|$ matrix. The above minimization problem becomes:

$$\text{Minimize:} \quad \sum_{s=1}^{M \times N} a_s^2 = \mathbf{a}^t \mathbf{a} = \mathbf{w}^t E \mathbf{w}$$

$$\text{Subject to: } \underline{1}^t \mathbf{w} = 1 \text{ and } \mathbf{w} \geq \underline{0}$$

This can be transformed to a standard least squares problem and solved accordingly (see [LaH74]). Details can be found in the Appendix.

In the above discussion, weights were chosen for template A, so that its correlation with the boundary of B would be minimized everywhere. More than two parts can be taken into consideration. For example, if parts A, B and C are to appear in a scene and it is required to distinguish part A, then it is necessary to minimize both $\sum_{s \in \{B\}} a_s^2$ and $\sum_{s \in \{C\}} a_s^2$, where the sets {B} and {C} denote the accumulators that receive contributions from parts B and C respectively. Assume that {B} and {C} are disjoint (*disjoint accumulators* assumption), then the correlation matrix for A correlated with both B and C, E_{BC}^A, is related to the E matrices for A correlated with B and A correlated with C by the relation:

$$\mathbf{w}^t E_{BC}^A \mathbf{w} = \sum_s a_s^2 = \sum_{s \in \{B\}} a_s^2 + \sum_{s \in \{C\}} a_s^2 = \mathbf{w}^t (E_B^A + E_C^A)\mathbf{w}$$

If it is required to recognize A in the presence of one or more objects, then the autocorrelation of A should also be minimized. (The constraint $1^t \mathbf{w} = 1$ insures that the accumulator corresponding to the correct match always has a value of one.) The associated autocorrelation matrix is denoted by E_A^A. The correlation matrix for identifying A in the presence of B and C is, given the disjoint accumulators assumption, computed as follows:

$$E_{ABC}^A = E_A^A + E_B^A + E_C^A$$

The correlation matrix E_{ABC}^A can then be used in the weighting algorithm to obtain values for the weights w_i of the subtemplates of A.

The solution to the weights gives preciseness to the term "saliency." The weights may be interpreted as the saliency of the subtemplate of A with respect to parts B and

C. Furthermore, the weights optimally (in a least squares sense) decorrelate each subtemplate of A from B, C and other subtemplates of A, allowing the recognition of A in the presence of B and/or C even if A is partially occluded by B and/or C. The degree of occlusion possible depends on how much saliency is visible.

In summary, the pairwise correlation matrices, E_j^i, are determined from the database of part-geometries by computing the D matrices using the subtemplate matching process of Section 3. They can then be added together according to the particular part(s) required to be recognized. An $E^i_{set\ of\ parts}$ matrix will result for each part, i, to be recognized. A separate set of weights can then be computed, as outlined above, for use in the recognition of each part. This can amount to a considerable amount of computation. However, in the type of environment envisioned for this recognition algorithm--a CAD driven manufacturing cell, for instance--the computation is performed as part of a set-up phase and reused over a much longer period of time.

Adding a new part to the database amounts to finding the weighting matrix E_j^{new} between the new part and every other member, j, of the database and adding them to the $E^i_{set\ of\ parts}$ then recomputing the sets of salient weights. Removing an old part from consideration can be done in a similar fashion by first subtracting E_j^{old} from the $E^i_{set\ of\ parts}$.

Finally, it should be noted that the $E^i_{set\ of\ parts}$ matrices are constructed on the disjoint accumulator assumption. This is equivalent to ignoring the possibility of accidental contributions to an accumulator that can occur in practice and result in faulty recognition. In fact, this rarely occurs while the assumption greatly simplifies the calculation of saliency.

5. EXPERIMENTAL RESULTS

Figure 11 illustrates the results of an experiment to test the recognition technique. Figure 11(a) shows three keys, two of which, key A and key C, are very similar in appearance. In Figure 11(b) key B has been placed on key A, occluding key A; key C remains unoccluded. An attempt was made to locate key A in the application scene depicted in Figure 11(b) using the slope restricted Hough transform [Bal81]. Figure 11(c) illustrates how the location of key A in Figure 11(b) was incorrectly determined to be at the location of key C. Table 1 shows the largest peaks in the accumulator array and their location after the above experiment. They are ranked according to the value of their contents--the value shown is actually the true value normalized by the total weight of all the subtemplates from key A scaled by 1000. The accumulators form a 256 $\times$ 256 array, and the accumulator locations shown are of the form (*abscissa, ordinate*).

The peak that corresponds to the centroid of key A--shown next to a bullet--is at (56,118) and is ranked 65 places below the maximum which is the centroid of key C at

Rank	Value	Accumulator Coordinates
100	101	(188,145)
99	82	(188,143)
98	82	(188,144)
97	82	(188,146)
96	74	(188,133)
95	72	(188,128)
	● ● ●	
36	46	(188,165)
35	44	(56,118) ●
34	44	(187,144)

Table 1. Accumulator Array for Slope Restricted Hough Transform.

(188,145).

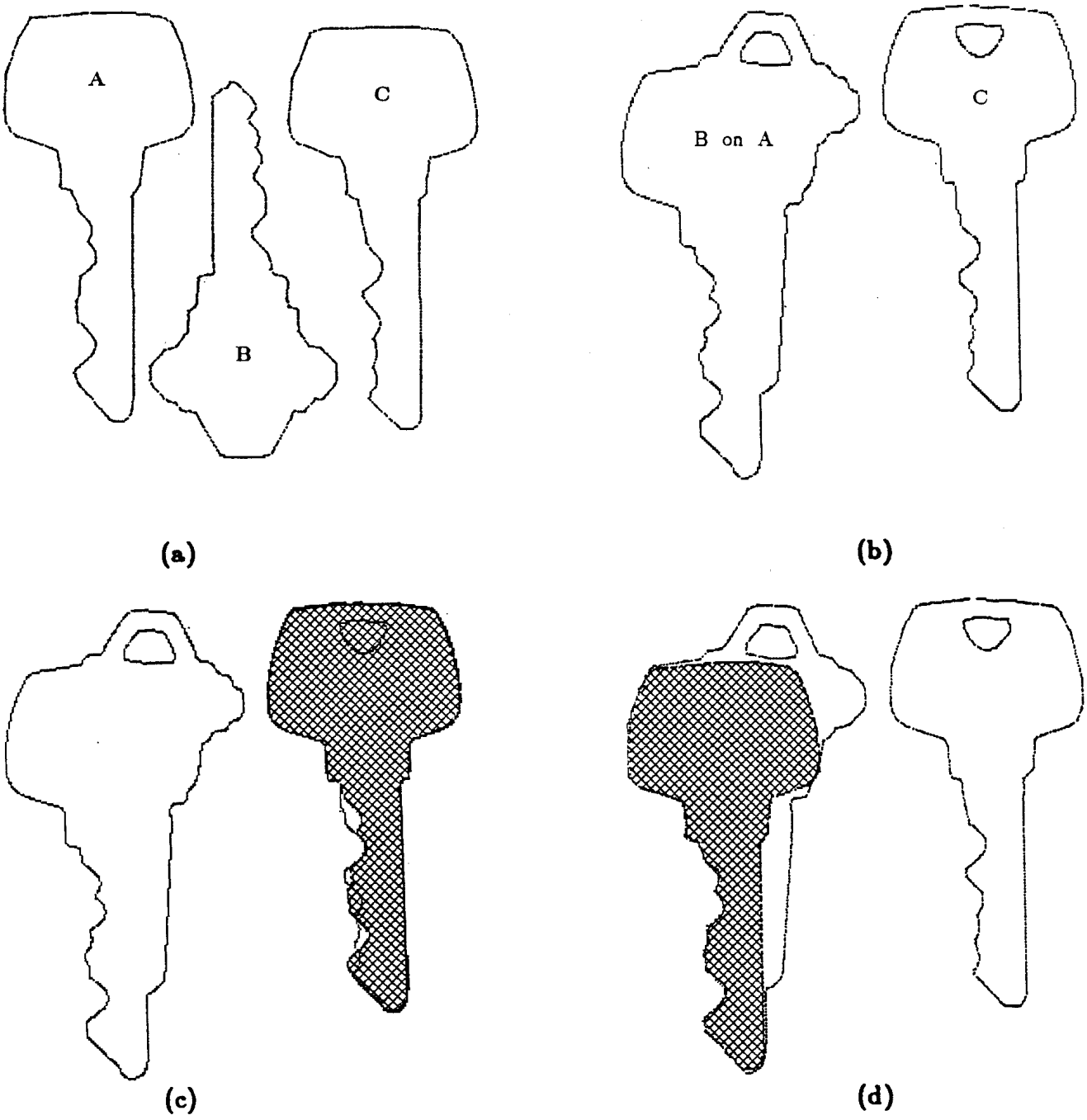


Figure 11. The Three Keys Experiment.

Next an attempt was made to locate key A in the application scene depicted in Figure 11(b) using the weighted subtemplate method. This attempt was successful and Figure 11(d) illustrates how key A was correctly located. Table 2 shows the contents of the accumulators; the maximum peak is at the location of A's centroid, (56,118).

A more complex experiment was performed with a set of nine parts. Figure 12(a) shows templates derived from the parts. Figure 12(b) shows the boundary extracted from an application scene in which the parts were piled in an arbitrary manner. Figure 12(c) shows the result of trying to locate the shaded part using the slope restricted Hough transform. Its position and orientation were incorrectly identified. Figure 12(d) shows the result of using the weighted subtemplate method to locate the part.

In other experiments with the pile in Figure 12(b) it was found that typical times for locating individual parts using the slope restricted Hough transform were in the range 4 - 4.5 CPU minutes using a VAX 11/780. In several cases an incorrect match was made. The typical times for the weighted subtemplate method were in the range 3 - 3.5 CPU minutes. In contrast, no incorrect matches were made. Preliminary results mentioned in the Conclusion suggest recognition time can be considerably shortened if

Rank	Value	Accumulator Coordinates
100	89	(56,118)
99	84	(57,118)
98	80	(56,117)
97	65	(57,117)
96	62	(58,118)
95	59	(187,147)
94	50	(59,118)
93	42	(55,118)
92	40	(58,116)

Table 2. Accumulator Array for the Weighted Subtemplate Method.

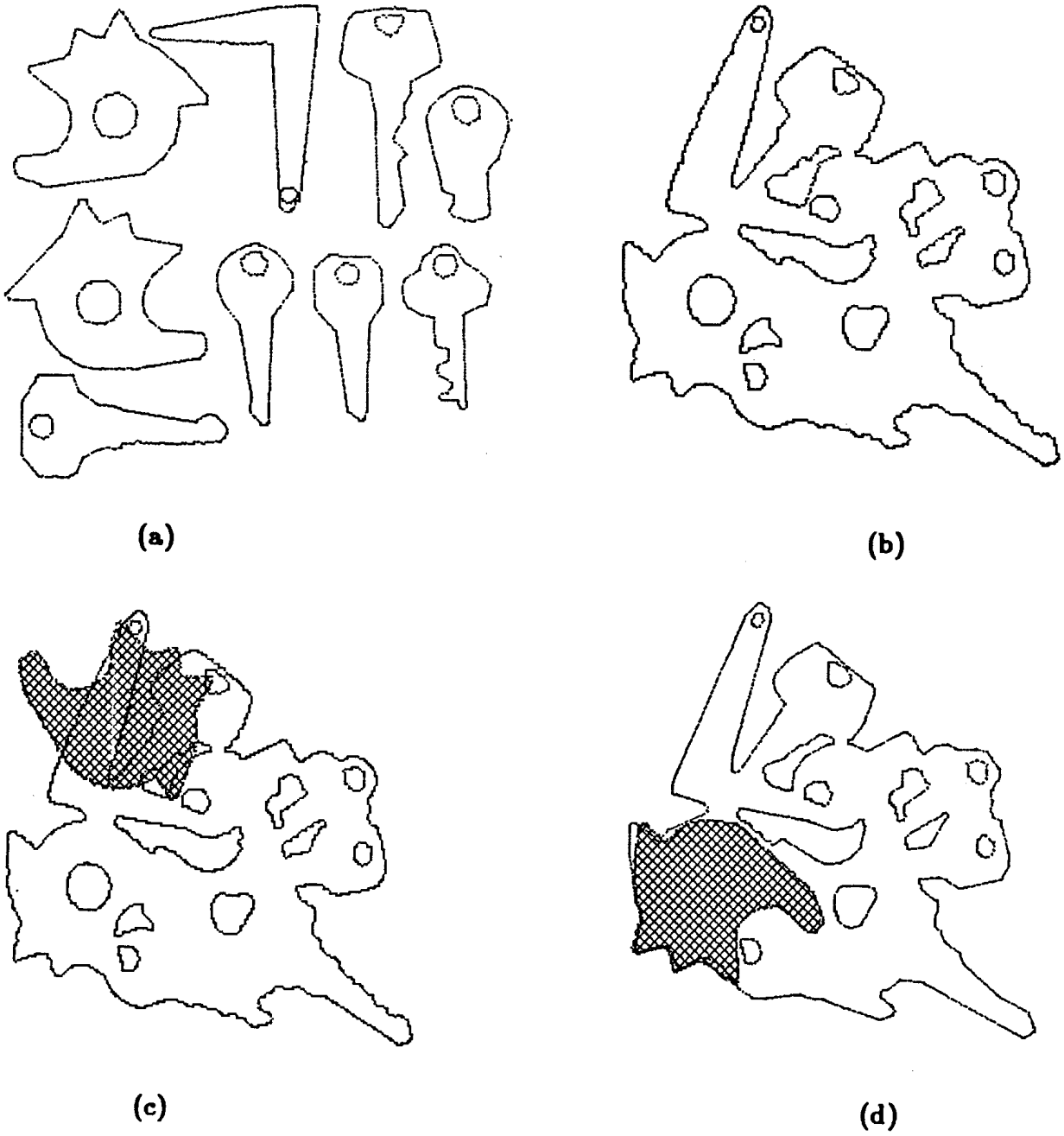


Figure 12. Experiments with a Set of Nine Parts.

the recognition process is based primarily on the more salient subtemplates. Both algorithms were written in C. Not much was attempted in the way of optimizing the implementation of the algorithms, apart from representing the accumulator arrays--very

sparse arrays--as hash tables. This significantly reduced storage requirements and made possible a speed-up in the search for the location of the maximum peak.

It is possible to get false matches using the weighted subtemplate method. They typically occur if the part to be located is almost totally occluded, or if the exposed portion does not distinguish the part from other parts. The latter situation is not unlike the confusion a human observer would experience when presented with views of occluded parts in which two or more parts had all their distinguishing features hidden from view.

6. CONCLUSION

A method for recognizing partially occluded parts has been presented. There are two principal components to the method. The first is a subtemplate matching scheme derived from earlier Hough transform techniques. This provides the features, subtemplates, for the second component which is a weighting scheme. The weighting scheme derives weights for the subtemplates that are a measure of their distinctiveness or saliency. Since the saliency of a part's subtemplate is entirely dependent on the parts with which it can appear (refer again to Figure 1), deriving the weights requires *a priori* knowledge about the set of parts that can appear in an application scene. The saliency weights are an embodiment of this knowledge and account for the added recognition capability of the method over earlier template matching methods. It is this important reliance on *a priori* information that distinguishes the weighted subtemplate method from other template matching methods that use weighting schemes (see Section 5 of [Bal81] for a good summary of these methods). However, the added recognition capability is bought at the expense of having to know ahead of time what the application scene may contain. Clearly, there are applications where this may not be possible. However, in our principal area of concern, recognizing manufactured parts, the set of parts likely to appear can usually be determined.

There are several techniques which may reduce recognition times that warrant further investigation. An obvious one, when several parts are being sought, is to eliminate segments of the boundary as soon as they are determined to belong to a part.

Another is to implement a coarse to fine search in matching the subtemplates to the boundary segments. The subtemplate matching is performed between subtemplates and boundary images that have been resampled at coarse intervals in θ–s space, i.e., less frequently than every pixel. A coarsely sampled subtemplate is matched to a coarsely sampled boundary segment and, if the match is poor, the matching coefficient is set to zero. On the other hand, if the match is satisfactory, matching is attempted

again but with subtemplates and boundaries more finely sampled. A sequence of increasingly more finely resampled templates and boundaries is used in matching, and, if they continue to show satisfactory matches, a situation is reached where the normal subtemplate is matched to the normal boundary and the matching coefficient can be determined. The potential for speed-up lies in the fact that most matches are very poor and that this can be recognized early in the above sequence when the number of points in the matching is relatively small. This technique requires a suitable definition of satisfactory and poor matches.

A technique which has perhaps the greatest potential to reduce recognition times takes advantage of the saliency of the subtemplates. A strategy is adopted, where, after matching only a few of the more salient subtemplates, the contents of affected accumulators are compared to a prescribed "detection" threshold. If the contents of one of these accumulators exceeds the threshold it is taken as a candidate for the associated template's centroid; the entire template is rotated to the angle of match and an attempt is made to match the entire template to the boundary. Work on this technique is still preliminary, but it has shown to provide a considerable speed-up--approximately 40 CPU seconds for a match. A quantitative method of choosing a "detection" threshold has yet to be determined. Such a threshold is needed also to determine if a part is absent. The idea is based on the observation that a large part of the recognition time is spent on matching the more common low weight subtemplates.

Finally, recognition times can be reduced if more than one processor were employed in running the algorithm. For example, each processor could be assigned to match the subtemplates for different parts. Strategies for multiprocessors are discussed further in [MuA83].

The experimental results presented suggest that saliency can be applied with advantage in the occluded parts problem. The concept is not intimately linked with boundary features; other features may be used, for example, internal edges are an obvi-

ous addition. Less obviously, preliminary studies suggest that future research consider working with subregions of surfaces obtained from a three-dimensional sensor. The three-dimensional template is supplied by a solid model database; scaling is possible from the range information.

7. APPENDIX

Techniques in the numerical solution of least squares problems include efficient methods for the solution of the following problem:

$$\text{Minimize:} \quad \|F\mathbf{x}-\mathbf{f}\|_2$$
$$\text{Subject to:} \quad G\mathbf{x} \geq \mathbf{g}$$

Where F is an $n_1 \times n$ matrix, G is an $n_2 \times n$ matrix, $\mathbf{x}$ is an n-vector (to be determined), $\mathbf{f}$ is an n_1-vector, and $\mathbf{g}$ is an n_2-vector. The problem is referred to as a "least squares with linear inequality constraints" (LSI) in [LaH74] which provides a detailed description of the algorithm.

To take advantage of these solution techniques it is first necessary to show that E is symmetric positive definite.

Theorem: E is symmetric positive definite.

Proof: Since E denotes the product $D^t D$, E is symmetric. If there exists an element of $\mathbf{w}$, w_i, that is nonzero, then the product $c_{ij} w_i$ is also nonzero because $c_{ij} > 0$ for all i and j (every r_i and β_j match to some extent). This product must contribute to some accumulator. Thus, there exists an s such that $\delta_{ij}^s = 1$ and, consequently, such that $a_s > 0$. Therefore, $\mathbf{a}^t \mathbf{a} > 0$ provided $\mathbf{w} \neq \underline{0}$. But from the minimization problem statement, $\mathbf{w}^t E\mathbf{w} = \mathbf{a}^t \mathbf{a}$. Thus E satisfies the definition of a symmetric positive definite matrix •

Since E is symmetric positive definite, Cholesky factorization can be used to factor E into the form $U^t U$, where U is a real upper triangular $|r| \times |r|$ matrix. Consequently, the minimization problem of Section 5 may be rewritten as the following least squares problem:

$$\text{Minimize:} \quad \|U\mathbf{w}\|_2$$

$$\text{Subject to:} \quad \underline{1}^t \mathbf{w} = 1 \quad \text{and} \quad \mathbf{w} \geq \underline{0}$$

This is similar to a special case of the LSI problem, as can be seen by setting:

$$F = U,\ \mathbf{x} = \mathbf{w},\ \mathbf{f} = \underline{0},\ G = \begin{matrix} I & \underline{0} \\ \underline{1}^t & 1 \end{matrix},\ \text{and}\quad \mathbf{g} = \begin{matrix} \underline{0} \\ 1 \end{matrix}$$

To yield the following:

$$\text{Minimize:}\quad \|U\mathbf{w}\|_2$$

$$\text{Subject to:}\quad \underline{1}^t\mathbf{w} \geq 1 \ \text{ and } \ \mathbf{w} \geq \underline{0}$$

This is equivalent to the minimization problem, if the inequality, $\underline{1}^t\mathbf{w} \geq 1$, constraint is replaced with the equality, $\underline{1}^t\mathbf{w} = 1$. The following theorem shows that the solution is unaffected by changing the inequality constraint to an equality. The theorem will be stated in terms of minimizing $\mathbf{w}^t E\mathbf{w}$ rather than in terms of the least squares form involving U.

Theorem: Given that E is symmetric positive definite, the problem:

$$\text{Minimize:}\quad \mathbf{w}^t E\mathbf{w}$$

$$\text{Subject to:}\quad \underline{1}^t\mathbf{w} \geq 1 \ \text{ and } \ \mathbf{w} \geq \underline{0}$$

has the same solution if $\underline{1}^t\mathbf{w} \geq 1$ is replaced by $\underline{1}^t\mathbf{w} = 1$.

Proof: A complete proof is omitted for brevity. Instead, the basis for the validity of the theorem is sketched by referring to the two-dimensional case illustrated in Figure 10.

Clearly, the inequality $\underline{1}^t\mathbf{w} \geq 1$ allows the possibility of solutions above the line given by $w_1 + w_2 = 1$, i.e., when strict inequality holds. Suppose the point X satisfies the constraints $\underline{1}^t\mathbf{w} > 1$ and $\mathbf{w} \geq \underline{0}$ and represents the minimum solution, then there is a hyperellipse of constant value passing through X. Draw a line from the origin through X. Since $\underline{1}^t\mathbf{w} > 1$ there exists a point Y on the line between X and the line $w_1 + w_2 = 1$ which also satisfies all constraints. This can be expressed algebraically by: $\mathbf{w}_Y = k\mathbf{w}_X$, where $0 < k < 1$. Thus:

$$\mathbf{w}_Y^t E\mathbf{w}_Y = k^2\mathbf{w}_X^t E\mathbf{w}_X < \mathbf{w}_X^t E\mathbf{w}_X$$

contradicting the minimality of X. Therefore, the solution point lies on the line $\underline{1}^t \mathbf{w} > 1$ •

Thus, the minimization problem of Section 5 can be solved using the LSI solution technique with the above substitutions for F, $\mathbf{x}$, $\mathbf{f}$, G, and $\mathbf{g}$.

8. REFERENCES

[ACM83] A. Arbuschi, V. Cantoni and G. Musso, "Recognition and Location of Mechanical Parts Using the Hough Technique," *Image Analysis*, Ed. S. Levialdi, London: Pittman Books, 1983.

[Bal81] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, No. 2, 1981, pp. 111-122.

[BaP71] H. G. Barrow and R. J. Popplestone, "Relational Descriptions in Picture Processing," *Machine Intelligence*, Vol. 6, 1971, pp. 377-396.

[Bau81] E. W. Baumann, "Model Based Vision and the MCL Language," *IEEE Systems, Man and Cybernetics Conference*, Oct. 1981, pp. 433-438.

[Bau82] E. W. Baumann, "CAD Model Input for Robotic Sensory Systems," *Proceedings of Autofact IV*, Nov. 1982.

[BlZ76] H. J. Blinchikoff and A. I. Zverev, *Filtering in the Time and Frequency Domains*, New York: Wiley, 1976.

[DeC83] E. J. Delp and C.-H. Chu, *Edge Detection Using Contour Tracing*, CRIM Report No. RSD-TR-12-83, Department of Electrical and Computer Engineering, University of Michigan, Jul. 1983, 47 pp.

[Hou62] P. V. C. Hough, *Methods and Means for Recognizing Complex Patterns*, U. S. Patent 3069654, 1962.

[LaH74] C. L. Lawson, R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974.

[MeF75] P. M. Merlin and D. J. Farber, "A Parallel Mechanism for Detecting Curves in Pictures," *IEEE Transactions on Computers*, Vol. C-24, No. 1, Jan. 1975, pp. 96-98.

[MuA83] T. N. Mudge and T. S. Abdel-Rahman, "Case Study of a Program for the Recognition of Occluded Parts," *Proceedings of the 2nd Annual IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Data Base Management*, Pasadena, CA, Oct. 1983, pp. 56-60.

[Per78] W. A. Perkins, "A Model-based Vision System for Industrial Parts," *IEEE Transactions on Computers*, Vol. C-27, No. 2, Feb. 1978, pp. 923-926.

[Per80] W. A. Perkins, "Simplified Model-based Part Locator," *Proceedings of the 5th International Conference on Pattern Recognition*, Dec. 1980.

[RoK76] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, New York, NY: Academic Press, 1976

[Skl78] J. Sklansky, "On the Hough Technique for Curve Detection," *IEEE Transactions on Computers*, Vol. C-27, No. 10, Oct. 1978, pp. 923-926.

[TMV83a] J. L. Turney, T. N. Mudge, R. A. Volz and M. D. Diamond, "Experiments in Occluded Parts Recognition Using the Generalized Hough Transform," *Proceedings of the Conference on Artificial Intelligence,* Oakland University, Rochester, MI, April 1983 (to appear).

[TMV83b] J. L. Turney, T. N. Mudge and R. A. Volz, "Experiments in Occluded Parts Recognition," *Proceedings of the Society of Photo-optical Instrumentation Engineers Cambridge Symposium on Optical and Electro-optical Engineering,* Cambridge, MA, Nov. 1983, (to appear).

[VMG83] R. A. Volz, T. N. Mudge and D. A. 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 through 12-57.

[WWV82] J. Wolter, T. C. Woo and R. A. Volz, "Gripping Position for 3D Objects," *Proceedings of the 1982 Meeting of the Industry Applications Society,* Oct. 1982, pp. 1309-1314.