

Automatic Target Recognition by Matching Oriented Edge Pixels

Clark F. Olson and Daniel P. Huttenlocher

Abstract—This paper describes techniques to perform efficient and accurate target recognition in difficult domains. In order to accurately model small, irregularly shaped targets, the target objects and images are represented by their edge maps, with a local orientation associated with each edge pixel. Three-dimensional objects are modeled by a set of two-dimensional (2-D) views of the object. Translation, rotation, and scaling of the views are allowed to approximate full three-dimensional (3-D) motion of the object. A version of the Hausdorff measure that incorporates both location and orientation information is used to determine which positions of each object model are reported as possible target locations. These positions are determined efficiently through the examination of a hierarchical cell decomposition of the transformation space. This allows large volumes of the space to be pruned quickly. Additional techniques are used to decrease the computation time required by the method when matching is performed against a catalog of object models. The probability that this measure will yield a false alarm and efficient methods for estimating this probability at run time are considered in detail. This information can be used to maintain a low false alarm rate or to rank competing hypotheses based on their likelihood of being a false alarm. Finally, results of the system recognizing objects in infrared and intensity images are given.

I. INTRODUCTION

THIS PAPER considers methods to perform automatic target recognition by representing target models and images as sets of oriented edge pixels and performing matching in this domain. While the use of edge maps implies matching 2-D models to the image, 3-D objects can be recognized by representing each object as a set of 2-D views of the object. Explicitly modeling translation, rotation in the plane, and scaling of the object (i.e. similarity transformations), combined with considering the appearance of an object from the possible viewing directions, approximates the full, six-dimensional (6-D), transformation space.

This representation provides a number of benefits. Edges are robust to changes in sensing conditions, and edge-based techniques can be used with many imaging modalities. The use of the complete edge map to model targets rather than approximating the target shape as straight edge segments allows small, irregularly shaped targets to be modeled accurately. Furthermore, matching techniques have been developed for

edge maps that can handle occlusion, image noise, and clutter and that can search the space of possible object positions efficiently through the use of intelligent search strategies that are able to rule out much of the search space with little work.

One problem that edge matching techniques can have is that images with considerable clutter can lead to a significant rate of false alarms. This problem can be reduced by considering not only the location of each edge pixel but, in addition, their orientations when performing matching. Our analysis and experiments indicate that this greatly reduces the rate at which false alarms are found. An additional benefit of this information is that it helps to prune the search space and thus leads to improved running times.

We must have some decision process that determines which positions of each object model are output as hypothetical target locations. To this end, Section II describes a modified Hausdorff measure that uses both the location and orientation of the model and image pixels in determining how well a target model matches the image at each position. Section III then describes an efficient search strategy for determining the image locations that satisfy this modified Hausdorff measure and are thus hypothetical target locations. Pruning techniques that are implemented using a hierarchical cell decomposition of the transformation space allow a large search space to be examined quickly without missing any hypotheses that satisfy the matching measure. Additional techniques to reduce the search time when multiple target models are considered in the same image are also discussed.

In Section IV, the probability that a false alarm will be found when using the new matching measure is discussed, and a method to estimate this probability efficiently at run time is given. This analysis allows the use of an adaptive algorithm, where the matching threshold is set such that the probability of a false alarm is low. In very complex imagery, where the probability of a false alarm cannot be reduced to a small value without the risk of missing objects that we wish to find, this estimate can be used to rank the competing hypotheses based on their likelihood of being a false alarm. Section V demonstrates the use of these techniques in infrared and intensity imagery. The accuracy with which we estimate the probability of a false alarm is tested, and the performance of these techniques is compared against a similar system that does not use orientation information. Finally, a summary of the paper is given.

Due to the volume of research that has been performed on automatic target recognition, this paper discusses only the previous research that is directly relevant to the ideas

Manuscript received November 1, 1995; revised June 13, 1996. This work was supported in part by ARPA under ARO Contract DAAH04-93-C-0052 and by National Science Foundation PYI Grant IRI-9 057 928.

C. F. Olson was with the Department of Computer Science, Cornell University, Ithaca, NY 14853 USA. He is now with the Jet Propulsion Laboratory, Pasadena, CA 91109 USA.

D. P. Huttenlocher is with the Department of Computer Science, Cornell University, Ithaca, NY 14853 USA.

Publisher Item Identifier S 1057-7149(97)00657-X.

described here. The interested reader can find overviews of automatic target recognition from a variety of perspectives in [2], [3], [6], [9], and [22]. Alternative methods of using object edges or silhouettes to perform automatic target recognition have been previously examined, for example, in [7], [20], and [21]. Portions of this work have been previously reported in [13]–[15].

II. MATCHING ORIENTED EDGE PIXELS

This section first reviews the definition of the Hausdorff measure and how a generalization of this measure can be used to decide which object model positions are good matches to an image. This generalization of the Hausdorff measure yields a method for comparing edge maps that is robust to object occlusion, image noise, and clutter. A further generalization of the Hausdorff measure that can be applied to sets of oriented points is then described.

A. The Hausdorff Measure

The directed Hausdorff measure from M to I , where M and I are point sets, is

$$h(M, I) = \max_{m \in M} \min_{i \in I} \|m - i\|$$

where $\|\cdot\|$ is any norm. This yields the maximum distance of a point in set M from its nearest point in set I . In the context of recognition, the Hausdorff measure is used to determine the quality of a match between an object model and an image. If M is the set of (transformed) object model pixels and I is the set of image edge pixels, the directed Hausdorff measure determines the distance of the worst matching object pixel to its closest image pixel. Of course, due to occlusion, it cannot be assumed that each object pixel appears in the image. The partial Hausdorff measure [11] between these sets is thus often used. It is given by

$$h_K(M, I) = K \max_{m \in M} \min_{i \in I} \|m - i\|. \quad (1)$$

This determines the Hausdorff measure among the K object pixels that are closest to image pixels. K can be set to the minimum number of object pixels that are expected to be found in the image if the object model is present or K can be set such that the probability of a false alarm occurring is small. Since this measure does not require that all of the pixels in the object model match the image closely, it is robust to partial occlusion. Furthermore, noise can be withstood by accepting models for which this measure is nonzero, and this measure is robust to clutter that may appear in the image since it measures only the quality of the match from the model to the image and not vice versa.

Typically, we are interested in whether a match with a size of at least K exists with Hausdorff measure below some threshold δ . It is useful to conceptualize this as a set containment problem. Let $S_1 \oplus S_2$ denote the Minkowski sum of sets S_1 and S_2 (or dilation of S_1 by S_2). The statement $h(M, I) < \delta$ is equivalent to $M \subset (I \oplus E_\delta)$, where E_δ is a disk of radius δ centered at the origin in the appropriate L_p

norm:

$$E_\delta = \{x \mid \|x\| \leq \delta\}.$$

Similarly, $h_K(M, I) < \delta$ and $|M \cap (I \oplus E_\delta)| \geq K$ are equivalent, where $|\cdot|$ denotes cardinality.

One method of determining whether a match of size K exists is to dilate the image pixels I by E_δ and probe the result at the location of each of the model pixels in M . Each time a probe hits a pixel in the dilated image, a match for a pixel in the object model has been found. A count on the number of these matches is kept. If the count surpasses K , then a match with a size of at least K has been found at this position of the object model.

When there is a combination of a small object model and a complex image, this measure can yield a significant number of false alarms, particularly when the transformation space is large [13]. This problem can be solved, in part, by using orientation information in addition to location information in determining the proximity between pixels in the transformed object model and the image.

B. The Generalization to Oriented Points

The Hausdorff measure can be generalized to incorporate oriented pixels by considering each edge pixel in both the object model and the image to be a vector in \mathbb{R}^3 :

$$p = \begin{bmatrix} p_x \\ p_y \\ p_o \end{bmatrix}$$

where (p_x, p_y) is the location of the point, and p_o is the local orientation of the point (e.g., the direction of the gradient, edge normal, or tangent). Typically, we are concerned with edge points on a pixel grid, and the x and y values thus fall into discrete sets. The orientations can be mapped into a discrete set in a similar manner. Let us call a set of image points that have been extended in this fashion an *oriented image edge map* I_o , and similarly, let us call such an extended set of points in the object model an *oriented model edge map* M_o .

We now need a measure to determine how well these oriented edge maps match. Among pixels with the same orientation, we would like the measure to reduce to the previous Hausdorff measure. Furthermore, the previous measure should be a lower bound on the new measure. One measure that fulfills these conditions is

$$h_\alpha(M, I) = \max_{m \in M} \min_{i \in I} \max \left\{ \left\| \begin{bmatrix} m_x - i_x \\ m_y - i_y \end{bmatrix} \right\|, \frac{|m_o - i_o|}{\alpha} \right\}.$$

This has the same general form as the previous Hausdorff measure, but the distance between two points is now measured by taking the maximum of the distances in translation and orientation. In this measure, α is a normalization factor that makes the orientation values implicitly comparable with the location values. In practice, this allows the specification of a maximum deviation in translation and in orientation for two pixels to match, and thus, a count of the number of model pixels that match image pixels according to both conditions can be kept. The parameters α and δ can be set arbitrarily to

adjust the required proximities. A partial measure for oriented points that is robust to occlusion can also be formulated similar to (1).

Our system discretizes the orientations such that $\alpha = 1$ and uses the L_∞ norm. In this case, the measure for oriented points simplifies to

$$h(M, I) = \max_{m \in M} \min_{i \in I} \|m - i\|_\infty.$$

III. SEARCH STRATEGY

Recent work [11]–[13], [17], [19] has shown that efficient methods can be formulated to search the space of possible transformations of the model to find the position with the minimum Hausdorff measure or all positions where the measure is below some threshold. This section discusses how such methods operate in general and how they can be extended to oriented points. In addition, we describe techniques that are used to reduce the running time of the system when there are multiple object models that may appear in the image.

A. Matching Edge Pixels

Chamfer matching [1], [5] is an edge matching technique that minimizes the sum of the distances from each object edge pixel to its closest image edge pixel over the space of possible transformations. This technique is closely related to minimizing the generalized Hausdorff measure, which instead minimizes the K th largest of these distances. Since the chamfer measure sums the distances over all of the object pixels, it is not robust to occlusion. In the original formulation of chamfer matching, Barrow *et al.* [1] used a starting hypothesis and an optimization procedure to determine a position of the model that is a local minimum with respect to the chamfer measure. This method requires a good starting hypothesis to converge to the global minimum.

Borgefors [5] proposed a hierarchical method that examines an edge pyramid of the model and image. A number of initial positions are considered at some level of the pyramid, where a Gauss–Seidel optimization procedure is used to find a local minima for each initial position. Poor local minima are rejected. The remaining positions are considered at the next lower level of the pyramid, and the procedure is repeated until local minima are found at the lowest level of the pyramid. This technique performs a search of the image for good local minima, but it still cannot guarantee that the best transformation is found.

Paglieroni *et al.* [16], [17] have considered methods to speed up the search over all possible transformations in chamfer matching by probing a distance transform of the image at the locations of the transformed object edge pixels. This distance transform measures the distance of each pixel in the image from an edge pixel and can be computed efficiently using a two-pass algorithm [18], [4], [16]. If the sum of the distance transform probes at each of the object pixels at some transformation is large enough, then we can rule out not only this transformation but also many transformations close to it since we know that the close transformations will yield a similar distance transform value for each pixel in the object

model. This method is able to search an entire image efficiently and is able to guarantee that the best match (or all matches that surpass some threshold) according to the chamfer measure are found.

Similar techniques have been developed to perform efficient matching using the generalized Hausdorff measure [11], [12], [19], which is robust to partial occlusions of the object. First, the image is dilated by E_δ (as described in the previous section), and the distance transform of this dilated image is determined. If the K th largest probe into this distance transform is 0, then a match of size (at least) K has been found. Otherwise, the K th largest probe yields the distance to the closest possible position of the object model that could produce a match of size K . We can thus rule out any transformation that does not move any object pixel more than this distance. To improve efficiency, the transformation space is discretized, but to ensure that no good matches are missed, this discretization is such that adjacent transformations do not map any object pixel more than one pixel (Euclidean distance) apart in the image. Now, if d is the value of the K th largest probe, we can rule out at least those transformations with a city-block distance (L_1 norm) less than d from the current transformation in the discretized transformation space since such transformations are guaranteed to move each object pixel less than d pixels from the current location.

B. Using Oriented Pixels

Since the oriented object and image pixels have three degrees of freedom, a 3-D distance transform is now required. Before this can be computed, we must consider how rotations of object models will be treated since such rotations change the orientations of the object pixels. If we wish to rule out nearby transformations that may change the orientations of object pixels, then this must be accounted for the distance transform, but this is problematic since the discretization of the rotations in the transformation space will, in general, be very different from the discretization of the orientations of the edge pixels. To avoid this problem, each rotation of an object model is treated independently (essentially as a separate object model). This allows each orientation plane of the distance transform to be treated independently.

It must also be decided how the models will be rotated and scaled to compare them to the image. If a CAD model is available from which the edges of our targets can be determined, these models can be rotated before performing the edge detection stage since different rotations of the model are treated as (essentially) separate models. On the other hand, if the original model consists only of a set of edge points, each point is simply rotated around the center of the model. Similarly, scaling of the model is performed by scaling each point with respect to the center of the model.

It is now possible to use Hausdorff matching techniques similar to those for unoriented points to perform efficient recognition. This is accomplished by considering a hierarchical cell decomposition of the transformation space [12], [19]. The transformation space is first discretized as above and divided into a set of rectilinear cells on the discrete grid of trans-

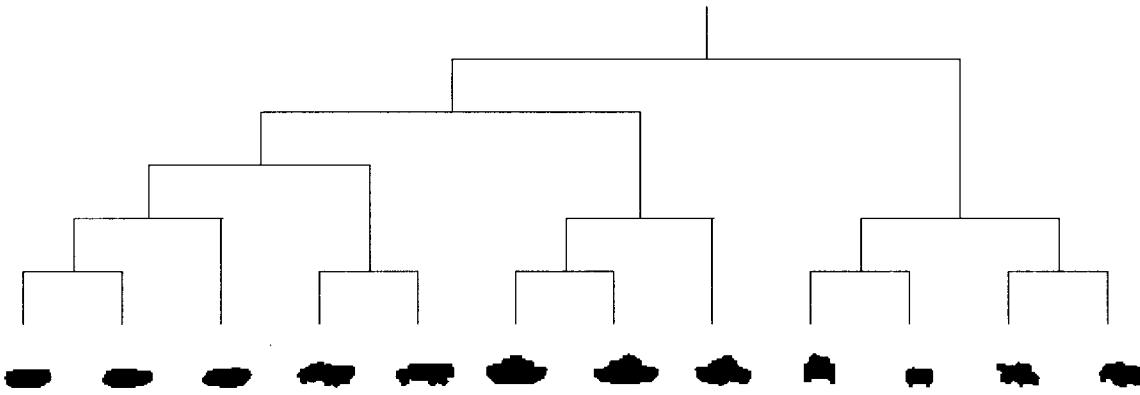


Fig. 1. Hierarchical clustering of the models is performed as the canonical positions of the models relative to each other are determined. This figure shows an example of the hierarchy produced by these techniques for 12 model views. The full silhouettes are shown rather than the edge maps for visual purposes.

formations. Since the orientations are treated independently, these cells have three dimensions: scale and translation in x and y . For each such cell, the discrete transformation that is closest to the center of the cell is considered. If the match at this transformation is poor enough that the entire cell can be ruled out using the techniques described above, then the cell is pruned. Otherwise, the cell is divided into subcells, and each of the subcells is considered recursively. If a cell is reached that contains only one transformation, then the transformation is tested explicitly. This search strategy corresponds to a depth-first tree search of the cells in the transformation space where pruning is applied when possible.

To process a single cell, the following steps are performed. First, a discrete transformation close to the center of the cell is chosen, and the maximum difference in the transformed location of a model pixel between the center transformation and any other transformation in the cell must be computed. This is bounded by the sum of the distance in the scale direction (by counting the number of discrete scales) between the transformations and the maximum of the distances in the x and y directions since we use the L_∞ norm in the image space. The distance transform is then probed at the locations of the model pixels after transforming them by the transformation at the center of the cell. If the K th largest probe into the distance transform is greater than the maximum distance any other transformation in the cell can move an object pixel from its current position, then the entire cell can be pruned. This is determined simply by counting the number of probes that yield a greater value than the computed distance. Otherwise, the cell is divided into either two subcells by cutting at the midpoint of the range of scales in the cell or into four subcells by cutting in both the x and y translations based on whether the distance in scale is greater than the distance in translation in both x and y .

The examination of a single cell in the transformation space can be performed very quickly if some preprocessing is performed. The index into the array storing the distance transform for each pixel of each model at every rotation and scale can be computed in advance. For a particular translation, these pointers into the distance transform array need only be offset by a constant amount, and these indexes can be used directly to probe at the locations of pixels of the object model.

C. Considering Multiple Models

When there are multiple object models that may appear in a single image, there are methods by which the search can be made faster than examining each object model sequentially. This section describes one such method. Note that these object models need not come from separate objects; they may be alternate views of the same object.

The first step is to determine a canonical position for each model with respect to the other models and to construct a hierarchical representation of the model set. This step is performed off line, prior to recognition. For our multiple model search strategy, it is desirable to maximize the number of pixels between the edge maps of various models that overlap in their canonical position in both position and orientation.

The best relative position between each pair of individual models according to the chamfer measure [1] is determined using search techniques similar those described above. The chamfer measure sums the distances from each pixel in one image to their closest neighbors in the other. This measure is asymmetric since the chamfer measure from some model M_i to another M_j is not necessarily the same as the reverse measure from M_j to M_i . A symmetric version is used that takes the maximum of the two measures. This measure is used as a score indicating how well each pair of models match.

The method builds a tree of models using hierarchical clustering techniques [8]. At each step, the two closest models are determined and clustered. This yields a canonical position for these models with respect to each other and a new set of model points replacing the two previous models. The new "model" is then compared with the remaining models as above, and the process is repeated until all of the models belong to a single hierarchically constructed model tree. At this point, canonical positions for each model with respect to the others have been computed, and a model hierarchy represented by a binary tree has been determined, where the leaves of the tree are individual models, and the remaining nodes correspond to the set of models below them in the tree. Fig. 1 shows a small example.

It should be noted that this procedure can be time consuming if there are a large number of models since the clustering procedure requires $O(M^2 \log M)$ time with a significant constant factor, where M is the number of model views. Since this step

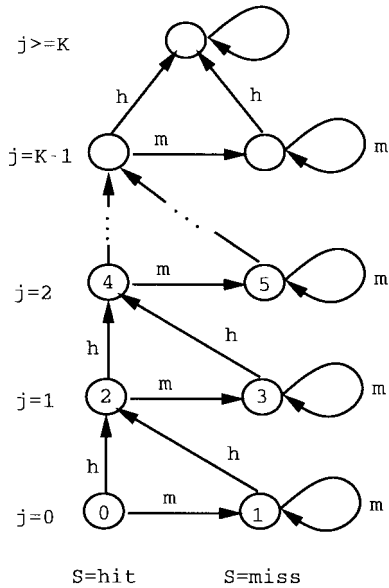


Fig. 2. Markov chain that counts the number of object pixels that match image pixels.

is performed off line, it is usually acceptable to expend a lot of computation here. For very large model sets, there are a number of heuristics that can be used to reduce the time that this process requires.

For each node in the tree, the model points that overlap at the canonical positions of all of the models below the node in the tree are stored, except for those that are stored at ancestors of the node. The amount of repeated computation among the object models can now be reduced using the computed model hierarchy. At each transformation considered, the hierarchy is searched starting at the top, and the probes are performed for the model points that are stored at each node. A count on the number of probes that yield a distance greater than the distance to the edge of the cell in the transformation space is kept for each node, and this count is propagated to the children of the node. If this count reaches a large enough value, the subtree of the model hierarchy for this cell of the transformation space and all of its subcells can be pruned. This is continued until all of the object models have been pruned or it is determined that not all of the object models can be pruned, and thus, the cell must be subdivided. If a cell that contains only a single transformation cannot be pruned, then a hypothetical target location is output.

IV. PROBABILITY OF A FALSE ALARM

This section discusses the probability that a false alarm will occur when matching is performed using the matching measure described in Section II. Methods by which this probability can be estimated efficiently during run time and how this estimate can be used to improve the performance of the recognition system are examined in detail.

A. A Simple Model for Matching Oriented Pixels

Let us consider matching a single connected chain of oriented object pixels to the image at some specified location.

For some pixel in the object chain, we will say that it results in a hit if the transformed object pixel matches an image pixel in both location and orientation according to our measure, and otherwise, we will say that it results in a miss. If the object chain is mapped to a sequence of such hits and misses, then this yields a stochastic process.

Note that if some pixel in the object chain maps to a hit, this means that locally, the object chain aligns with an image chain very closely in both location and orientation. It is thus very likely that the next pixel will also map to a hit since the chains are expected to continue in the direction specified by the local orientation with little change in this orientation. Let S_i be a random variable describing whether the i th object pixel is a hit or a miss, and let s_i be the value taken by this variable for a specific object chain. If the probability of being in each state at each pixel is dependent only on i and the previous state

$$\begin{aligned} \Pr[S_i = s | (S_{i-1} = s_{i-1}) \wedge \dots \wedge (S_0 = s_0)] \\ = \Pr[S_i = s | S_{i-1} = s_{i-1}] \end{aligned}$$

then the process is said to be a *Markov process*. If, furthermore, the probability does not depend on i , then the process is a *Markov chain*. To determine the probability distribution of the number of hits over the entire object model, the number of hits so far in our chain j must be counted explicitly. A separate state in the chain is thus used for each member of

$$\{hit, miss\} \times \{j \mid 0 \leq j \leq m\}$$

where m is the number of object pixels. If we are only interested in whether a false alarm of size K occurs, a Markov chain with $2K + 1$ states can be used (see Fig. 2). If the final state of this chain is reached due to matches with random edge chains in the image, then a false alarm has occurred.

Let us number the states in the Markov chain as follows:

$$\begin{aligned} 0 : & (S_i = h) \wedge (j = 0) \\ 1 : & (S_i = m) \wedge (j = 0) \\ 2 : & (S_i = h) \wedge (j = 1) \\ 3 : & (S_i = m) \wedge (j = 1) \\ & \vdots \\ 2k - 2 : & (S_i = h) \wedge (j = K - 1) \\ 2k - 1 : & (S_i = m) \wedge (j = K - 1) \\ 2k : & (j \geq K). \end{aligned}$$

Abbreviate $P(S_i = h | S_{i-1} = m)$ as P_{mh} . We now have the following state transition matrix for the Markov chain in Fig. 2:

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_{hm} & P_{mm} & 0 & 0 & 0 & 0 & 0 \\ P_{hh} & P_{mh} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{hm} & P_{mm} & 0 & 0 & 0 \\ 0 & 0 & P_{hh} & P_{mh} & \dots & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & & P_{hm} & P_{mm} & 0 \\ 0 & 0 & 0 & 0 & & P_{hh} & P_{mh} & 1 \end{bmatrix}.$$

Let p_0 be a vector containing the probability of the chain starting in each state. The probability distribution among the

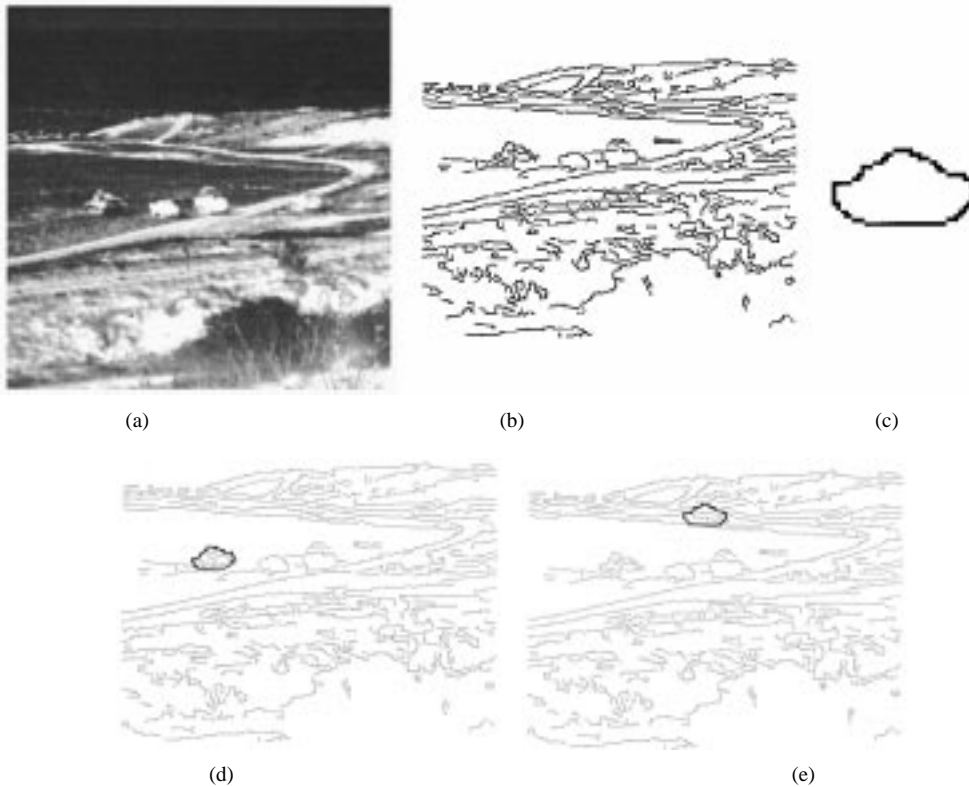


Fig. 3. Automatic target recognition example. (a) FLIR image after histogram equalization. (b) Edges found in the image. (c) Smoothed edges of a tank model. (d) Detected position of the tank. (e) False alarm.

states after examining the entire object chain is

$$p_m = T^m p_0.$$

The last element of p_m is the probability that a false alarm of size K will occur at this position of the model. The probability that a false alarm of any other size $K' \leq K$ will occur can be determined by summing the appropriate elements of p_m .

B. An Accurate Model for Matching

To model the matching process accurately, it is not correct to treat the state transition probabilities as independent of which pixel in the chain is examined. Consider the probability of a hit following another hit for two cases. In the first case, the two object pixels have the same orientation and lie along the line perpendicular to the gradient. In the second case, there is a significant change in the orientation and/or the segment between the pixels is not perpendicular to the gradient. The first case has a significantly higher probability of the second pixel being a hit given that the first pixel was a hit since the chain of image pixels is expected to continue in the direction perpendicular to the gradient with approximately the same gradient direction.

This means that the stochastic process of pixel hits and misses is not a Markov chain, but it is still a Markov process. Let T_i be the state transition matrix for the i th object pixel in such a process. The state probability vector p_m is now given by

$$p_m = \left(\prod_{i=0}^{m-1} T_i \right) p_0. \quad (2)$$

Furthermore, not all hits should be treated the same. In the Hausdorff measure, an image pixel may match more than one pixel in an object chain since the image is dilated prior to matching. This causes an effect such that after a pixel in the object chain first hits a pixel in the oriented image edge map, the following pixels in the object chain are likely to hit the same image pixel, especially if there is no orientation change between the object pixels. This effect dies off after a few pixels, but it means that the probability of an object pixel resulting in a hit is not dependent on only the previous state. A Markov process can still be used if the necessary information is encoded in the states of the process. When $\delta = 1$ is used (which is sufficient for most applications), the following states can be used:

- m : The object pixel did not hit an image pixel.
- n : The object pixel hit a new pixel in the oriented image edge map.
- o : The object pixel hit the same pixel in the oriented image edge map as the previous object pixel.
- p : The object pixel hit the same pixel in the oriented image edge map as the previous two object pixels.

It is possible for an object pixel to hit both a new pixel and an old pixel. In this case, state n takes precedence. To determine the probability distribution of the number of hits, a Markov process that consists of the cross product of these states with the count of the number of hits so far is used:

$$\{m, n, o, p\} \times \{j \mid 0 \leq j \leq K\}.$$

Experiments indicate that this model of the matching process is sufficient to achieve accurate results in determining

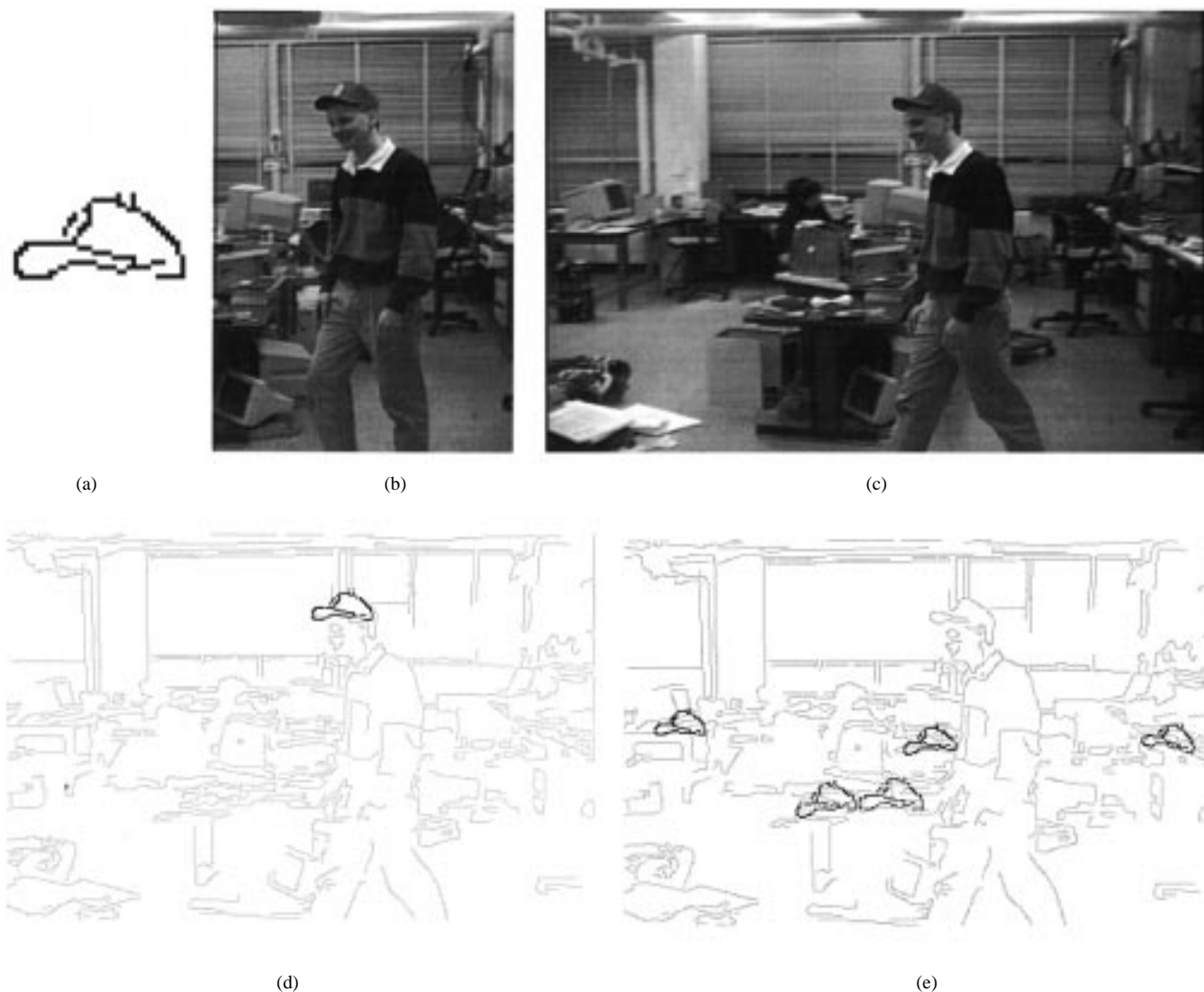


Fig. 4. Image sequence example. (a) Object model. (b) Part of the image frame from which the model was extracted. (c) Image frame in which we are searching for the model. (d) Position of the model located using orientation information. No false alarms were found for this case. (e) Several false alarms that were found when orientation information was not used. These each yielded a higher score than the correct position of the model.

the probability of a false alarm at a single specified position of the object in the image if accurate estimates for the transition probabilities are used.

C. State Transition Probabilities

The state transition probabilities must now be determined. These probabilities will be different in locations of the image that have different densities of edge pixels. Consider, for example, the probability of hitting a new pixel following a miss. The probability will be much higher if the window is dense with edge pixels rather than having few edge pixels. To model this, let us consider the window of the image that the object model overlays at some position. This is simply the rectangular subimage covered by the object model at this position. Each of these windows in the image will enclose some number d of image pixels. We call this the density of the image window. The state transition probabilities are closely approximated by linear functions of the number of edge pixels present in the image window and belong to one of two classes:

- 1) Probabilities that are linear functions passing through the origin (i.e., $\text{Pr} = k_i d$): The probability that an object model pixel hits a new image pixel, when the previous object model pixel did not hit a new pixel, is approximated by such a linear function of the density of image edge pixels in the image window. The following state transition probabilities are thus modeled in this manner: $P_{mn}(i)$, $P_{on}(i)$ and $P_{pn}(i)$. Note that each has a different constant k_i .
- 2) Probabilities that are constant (i.e., $\text{Pr} = c_i$): When the previous object model pixel hit an image pixel, the probability that the current object model pixel will hit the same image pixel is essentially constant. In addition, when the object model chain is following an image chain (i.e., the previous object model pixel hit a new image pixel), the probability that the object model chain continues to follow the image chain is approximately constant. The state transitions that are modeled in this manner are thus $P_{op}(i)$, $P_{no}(i)$, and $P_{nn}(i)$.

These probabilities are determined by sampling possible positions of the object model and comparing the object model to the image at these positions. This is performed by examining the pixels of the object model chain, in order, and determining whether each object model pixel hits an image pixel or not and, if so, whether the previous object model pixel(s) hit the same image pixel. In addition, for each case, the next state is recorded. The appropriate constant, given by $c_i = \Pr(i)$ or $k_i = \frac{\Pr(i)}{d}$, is then averaged over each of the sampled positions to estimate the correct value.

The remaining probabilities can be determined as a function of these probabilities as follows:

$$\begin{aligned} P_{nm}(i) &= 1 - P_{no}(i) - P_{nn}(i) \\ P_{om}(i) &= 1 - P_{on}(i) - P_{op}(i) \\ P_{pm}(i) &= 1 - P_{pn}(i) \\ P_{mm}(i) &= 1 - P_{mn}(i). \end{aligned}$$

If the state at $i = 0$ is considered to be m , this will yield the correct result for the first pixel in the object chain (i.e., $i = 1$). In this case, there are no previous object model pixels to compare against, and the probability of an object pixel resulting in a hit at random is desired. Similarly, if the object model consists of more than one chain of pixels, the state is reset to m when a new chain is started.

D. Probability of a False Alarm Over a Set of Transformations

Let us now consider the probability that there exists a false alarm at any translation of the object model. As with the search strategy, only translations on the integer grid are considered. While this may miss the optimal translation for our matching measure, this can increase the size of the minimum Hausdorff measure over the space of possible translations by at most $\frac{1}{2}$ when using the L_∞ norm.

While the probability that a false alarm occurs at some translation is not independent of whether a false alarm occurs at a close translation, previous work [10] has indicated that approximating these events as independent yields accurate results. These events will thus be treated as if they are independent here, and the performance of the model will be checked on real data to ensure that this assumption is realistic.

We do not assume that a target model will always appear either brighter or darker than the background in an image, but we do assume that individual targets will be either entirely brighter or entirely darker than the background, although this restriction can be easily removed. This means that each translation must be considered twice: once for the case when the target is brighter than the background and once for the case when the target is darker since the orientation of the point in these two cases will be shifted by π . If $P_K(t)$ is the probability of a false alarm of size K at translation t , the probability of a false alarm existing over all translations can be determined by computing

$$1 - \prod_t (1 - P_K(t)).$$

This can be computed more efficiently if we have a histogram of the number of edge pixels contained in the image

windows. Let d_i be the number of image windows containing i edge pixels for $0 \leq i \leq W$, where W is the size of the window in pixels. The probability of a false alarm in two image windows containing the same number of image pixels is the same in this estimation model. Let $P_K(i)$ be the probability of a false alarm of size K in a window containing i edge pixels. The probability of a false alarm is now given by

$$1 - \prod_{i=0}^W (1 - P_K(i))^{d_i}. \quad (3)$$

To estimate the probability of a false alarm when scaled and rotated versions of the target models are allowed in the matching process, the discretization of the transformation space must be considered. Rotating and scaling the object model does not move every pixel a uniform distance as translation does, but discrete rotations and scales can be considered such that two adjacent transformations move the farthest moving object pixel by no more than one pixel in the image (Euclidean distance), as in the search strategy. If these transformations are treated as being independent, an estimate of the probability of a false alarm can be obtained over discretized space of similarity transformations by sampling over the possible translations, scales, and rotations of the object model and following the above equations.

The overall steps in the estimation of the probability of a false alarm are as follows. First, possible locations of the object model in the image are sampled to estimate the probabilities in the state transition matrices T_i as a function of the density of the image window. A histogram of the number of edge pixels the image windows is also determined using dynamic programming. For each density, the probability that a false alarm occurs at a window with that density is estimated by computing (2). Equation (3) is used to estimate the probability of a false alarm occurring over the entire image. To improve the speed of this process, we consider only every 10th density value in the histogram and perform interpolation to estimate the remaining values.

The expected number of false alarms can also be estimated, if desired, as follows:

$$E(N_F) = \sum_{i=0}^W d_i P_K(i).$$

In addition, the *a priori* probability that any particular image window yields a false alarm can be estimated by examining the result of (2) for the density of that image window.

E. Using the False Alarm Rate Estimate

Now that we have a method to estimate the probability of a false alarm for any particular matching threshold, we can use the estimate to improve the performance of a recognition system that matches oriented edge pixels.

One method by which we could use the estimate is to set the matching threshold such that the probability of a false alarm is below some predetermined probability. However, this can be problematic in very cluttered images since it can cause correct instances of targets that are sought to be missed.

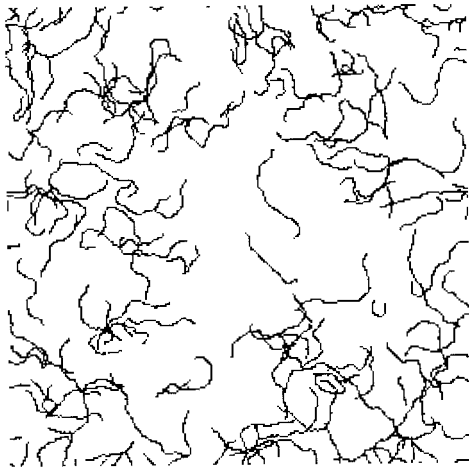


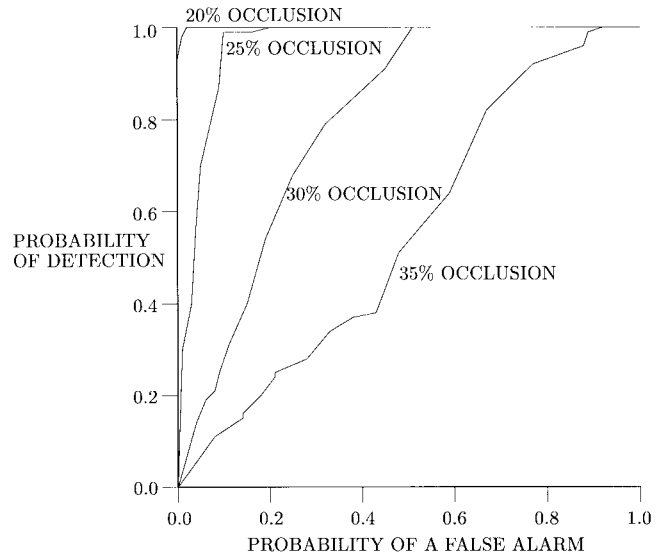
Fig. 5. One of the synthetic images used to generate ROC curves.

Alternatively, the matching threshold can be set such that it is expected that most or all of the correct target instances that are present in the image are detected. The techniques that have been described here yield an estimate on the probability that a false alarm will be found for this threshold as well as an estimate on the expected number of such false alarms, which will be useful when the probability is not small. More importantly, the likelihood that each hypothesis that we find is a false alarm can be determined by considering the *a priori* probability that the image window of the hypothesis yields a false alarm of the appropriate size as described above. These likelihoods can be used to rank the hypotheses by likelihood and the hypotheses for which the likelihood of being a false alarm is too high can be eliminated.

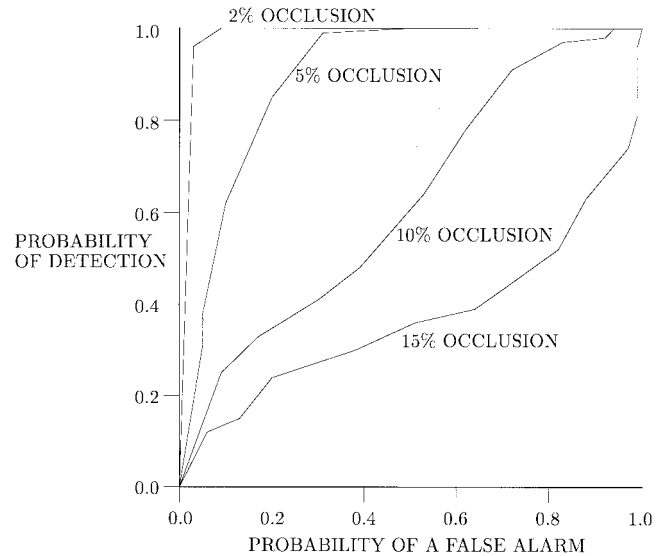
V. PERFORMANCE

Fig. 3 shows an example of the use of these techniques. The image is a low contrast infrared image of an outdoor terrain scene. After histogram equalization, a tank can be seen in the left-center of the image, although due to the low contrast, the edges of the tank are not clearly detected. Despite the mediocre edge image and the fact that the object model does not well fit the image target, a large match was found at the correct location of the tank. It should be noted, however, that this was not the only match reported. Fig. 3 also shows a false alarm that was found. Note that the image window for this false alarm is more dense with edge pixels than the correct location. The false alarm rate estimation techniques can be used to rank these hypotheses based on their likelihood of being a false alarm, although, in this case, the false alarm is a sufficiently good match that these techniques indicate that it is less likely to be a false alarm than the correct location of the target.

The current implementation of these techniques uses 16 discrete orientations and $\delta = \alpha = 1$ (each discrete orientation thus corresponds to $\frac{\pi}{8}$ rad, but matches are also allowed with neighboring orientations). In these experiments, the allowable orientation and scale change of the object views was limited to $\pm\frac{\pi}{18}$ and $\pm 10\%$, respectively, since we expect to have prior knowledge of the approximate range and orientation of the target.



(a)



(b)

Fig. 6. Receiver operating characteristic (ROC) curves generated using synthetic data. (a) ROC curves when using orientation information. (b) ROC curves when not using orientation information.

These techniques are not limited to automatic target recognition. Fig. 4 shows an example of the use of these techniques in a complex indoor scene. In this case, the object model was extracted from a frame in an image sequence, and it is matched to a later frame in the sequence (as in tracking applications). Since little time has passed between these frames, it is assumed that the model has not undergone much rotation out of the image plane, and thus, a four-dimensional (4-D) transformation space is used, consisting of translation, rotation in the plane, and scale. The position of the object was correctly located when orientation information was used. No false alarms were found for this case. When orientation information was not used, several positions of the object were found that yielded a better score than the correct position of the object.

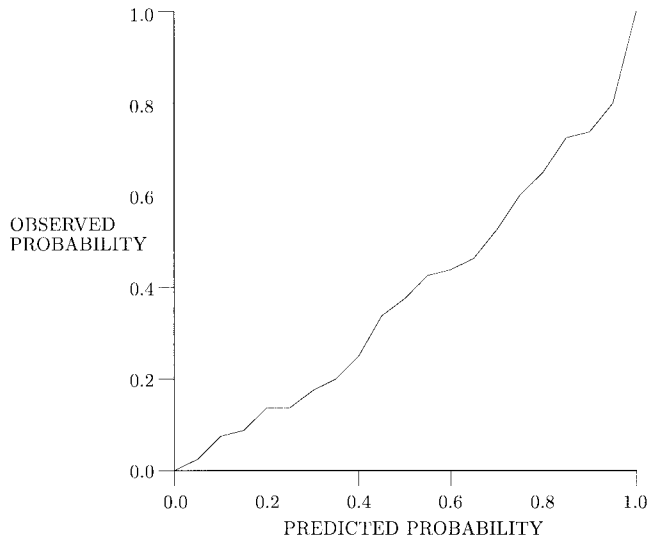


Fig. 7. Predicted probability of a false alarm versus observed probability of a false alarm in trials using real images.

We have generated ROC curves for this system using synthetic edge images. Each synthetic edge image was generated with 10% of the pixels filled with random image clutter (curved chains of connected pixels). An instance of a target was placed in each image with varying levels of occlusion generated by removing a connected segment of the target boundary. Random Gaussian noise was added to the locations of the pixels corresponding to the target. An example of such a synthetic image can be found in Fig. 5. Fig. 6 shows ROC curves generated for cases when orientation information was used and when it was not. These ROC curves show the probability that the target was located versus the probability that a false alarm of this target model was reported for varying levels of the matching threshold. When orientation information was used, the performance of the system was very good in these images up to 25% occlusion of the target. On the other hand, when orientation information was not used, the performance degraded significantly before 10% occlusion of the object was reached.

The false alarm rate (FAR) estimation techniques were tested on real imagery. In these tests, the largest threshold at which a false alarm was found was determined for each object model and image in a test set. In addition, the FAR estimation techniques were used to determine the probability that a false alarm of at least this size would be determined in each case. From this information, we can obtain the observed probability of a false alarm when the matching threshold is set to yield any predicted false alarm rate by determining the fraction of tests that yielded a false alarm with the matching threshold set to yield the predicted rate (see Fig. 7). In the ideal case, this would yield a straight line between (0.0,0.0) and (1.0,1.0). Since the plot that was produced by these tests lies slightly below this line for the most part, the FAR estimation techniques described here predict false alarms that are slightly larger than those observed in these tests, but the prediction performance is otherwise quite good.

TABLE I

PERFORMANCE COMPARISON. *Points* IS THE NUMBER OF POINTS IN THE MODEL. *Thresh* IS THE THRESHOLD USED TO DETERMINE HYPOTHESES. *Probes* IS THE NUMBER OF TRANSFORMATIONS OF THE OBJECT MODEL THAT WERE PROBED IN THE DISTANCE TRANSFORMS AND IS IN THOUSANDS. THE TIME GIVEN IS FOR MATCHING A SINGLE OBJECT MODEL AND NEGLECTS THE IMAGE PREPROCESSING TIME. *Biggest* IS THE SIZE OF THE LARGEST FALSE ALARM FOUND

	<i>Points</i>	<i>Thresh</i>	Using orientations			No orientations		
			<i>Probes</i>	<i>Time</i>	<i>Biggest</i>	<i>Probes</i>	<i>Time</i>	<i>Biggest</i>
Sample	67	53	122K	1.1s	63	2263K	11.0s	67
FLIR	67	60	49K	0.5s	62	1367K	5.9s	67
images	95	60	318K	4.5s	65	4396K	34.6s	95
	95	76	83K	1.1s	†	2383K	17.3s	95
Int. Image	123	98	78K	1.3s	99	1832K	17.2s	120

† No match was found surpassing the threshold for this case.

The computation time required by the system is low. The preprocessing stage requires approximately 7 s on a Sparc-5 for a 256×256 image. This stage performs the edge detection on the image, creates and dilates the oriented image edge map, and computes the distance transform on each orientation plane of the oriented image edge map. This step is performed only once per image. The running time per object view varies with the size of the object model and the matching threshold used, but we have observed times ranging from 0.5 to 4.5 s. See Table I for example times and counts on the number of transformations that were probed in each case. The prediction stage required approximately an additional 1.0 s per model to estimate the false alarm rate.

In addition to reducing the false alarm rate, the use of orientation information has significantly improved the speed of matching. Table I indicates that in a small sample of the trials, the search time is reduced by approximately a factor of 10 when everything else is held constant. The techniques to reduce the search time when multiple models were considered in a single image also helped to speed the search. When 27 different object models were considered in the same image using the multimodel techniques, 0.86 s were necessary per model to perform the matching when 80% of the model edge pixels were required to match the image closely, and 0.34 s were necessary per model with when 90% of the model edge pixels were required to match closely.

VI. SUMMARY

This paper has discussed techniques to perform automatic target recognition by matching sets of oriented edge pixels. A generalization of the Hausdorff measure that allows the determination of good matches between an oriented model edge map and an oriented image edge map was first proposed. A search strategy that allowed the full space of possible transformations to be examined quickly in practice using a hierarchical cell decomposition of the transformation space was then given. This method allows large volumes of the transformation space to be efficiently eliminated from consideration. Additional techniques for reducing the overall time necessary when any of several target models may appear in an image were also described. The probability that this method would yield false alarms due to random chains of edge pixels in the image was discussed in detail, and a method

to estimate the probability of a false alarm efficiently at run time was given. This allows automatic target recognition to be performed adaptively by maintaining the false alarm rate at a specified value or to rank the competing hypotheses that are found on their likelihood of being a false alarm. Experiments confirmed that the use of orientation information at each edge pixel, in addition to the pixel locations, considerably reduces the size and number of false alarms found. The experiments also indicated that the use of orientation information resulted in faster recognition.

The techniques described here yield a very general method to perform automatic target recognition that is robust to changes in lighting and contrast, occlusion, and image noise and that can be applied to a wide range of imaging modalities. Since efficient techniques exist to determine good matches, even when a large space of transformations are considered, and to determine the likelihood that a false alarm will be found or that any particular hypothesis is a false alarm, these methods are useful and practical in identifying targets in images.

REFERENCES

- [1] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *Proc. Int. Joint Conf. Artificial Intell.*, 1977, pp. 659–663.
- [2] B. Bhanu, "Automatic target recognition: State of the art survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, pp. 364–379, July 1986.
- [3] B. Bhanu and T. L. Jones, "Image understanding research for automatic target recognition," *IEEE Aerospace Electron. Syst. Mag.*, vol. 8, pp. 15–23, Oct. 1993.
- [4] G. Borgefors, "Distance transformations in digital images," *Comput. Vision, Graphics, Image Processing*, vol. 34, pp. 344–371, 1986.
- [5] ———, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 849–865, Nov. 1988.
- [6] W. M. Brown and C. W. Swonger, "A prospectus for automatic target recognition," *IEEE Trans. Aerospace Electron. Syst.*, vol. 25, no. 3, pp. 401–410, May 1989.
- [7] C. E. Daniell, D. H. Kemsley, W. P. Lincoln, W. A. Tackett, and G. A. Baraghimian, "Artificial neural networks for automatic target recognition," *Opt. Eng.*, vol. 31, no. 12, pp. 2521–2531, Dec. 1992.
- [8] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *J. Classification*, vol. 1, no. 1, pp. 7–24, 1984.
- [9] D. E. Dudgeon and R. T. Lacoss, "An overview of automatic target recognition," *Lincoln Lab. J.*, vol. 6, no. 1, pp. 3–9, 1993.
- [10] W. E. L. Grimson and D. P. Huttenlocher, "Analyzing the probability of a false alarm for the Hausdorff distance under translation," in *Proc. Workshop Performance versus Methodology Computer Vision*, 1994, pp. 199–205.
- [11] D. P. Huttenlocher, G. A. Klanderma, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 850–863, Sept. 1993.
- [12] D. P. Huttenlocher and W. J. Rucklidge, "A multi-resolution technique for comparing images using the Hausdorff distance," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1993, pp. 705–706.
- [13] C. F. Olson and D. P. Huttenlocher, "Recognition by matching dense, oriented edge pixels," in *Proc. Int. Symp. Comput. Vision*, 1995, pp. 91–96.
- [14] ———, "Determining the probability of a false positive when matching chains of oriented pixels," in *Proc. ARPA Image Understanding Workshop*, 1996, pp. 1175–1180.
- [15] C. F. Olson, D. P. Huttenlocher, and D. M. Doria, "Recognition by matching with edge location and orientation," in *Proc. ARPA Image Understanding Workshop*, 1996, pp. 1167–1174.
- [16] D. W. Paglieroni, "Distance transforms: Properties and machine vision applications," *CVGIP: Graphical Models Image Processing*, vol. 54, no. 1, pp. 56–74, Jan. 1992.
- [17] D. W. Paglieroni, G. E. Ford, and E. M. Tsujimoto, "The position-orientation masking approach to parametric search for template matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 740–747, July 1994.
- [18] A. Rosenfeld and J. Pfaltz, "Sequential operations in digital picture processing," *J. Assoc. Comput. Mach.*, vol. 13, pp. 471–494, 1966.
- [19] W. J. Rucklidge, "Locating objects using the Hausdorff distance," in *Proc. Int. Conf. Comput. Vision*, 1995, pp. 457–464.
- [20] F. Sadjadi, "Object recognition using coding schemes," *Opt. Eng.*, vol. 31, no. 12, pp. 2580–2583, Dec. 1992.
- [21] J. G. Verly, R. L. Delaney, and D. E. Dudgeon, "Model-based system for automatic target recognition from forward-looking laser-radar imagery," *Opt. Eng.*, vol. 31, no. 12, pp. 2540–2552, Dec. 1992.
- [22] E. G. Zelnio, "ATR paradigm comparison with emphasis on model-based vision," in *Proc. SPIE, Model-Based Vision Development Tools*, vol. 1609, 1992, pp. 2–15.



Clark F. Olson received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the University of Washington, Seattle, in 1989 and 1990, respectively. He received the Ph.D. degree in computer science from the University of California, Berkeley, in 1994.

He is currently a member of the technical staff in the Robotic Vehicles Group at the Jet Propulsion Laboratory, Pasadena, CA. From 1989 to 1990, he was a research assistant in the Intelligent Systems Laboratory at the University of Washington, where he worked on a translator for mapping machine vision programs onto a reconfigurable computational network architecture. From 1991 to 1994, he was a graduate student researcher in the Robotics and Intelligent Machines Laboratory at the University of California, Berkeley, where he examined efficient methods for performing model-based object recognition. From 1994 to 1996, he was a post-doctoral associate at Cornell University, Ithaca, NY, where he worked on automatic target recognition, curve detection, and the application of subspace methods to object recognition. His current research interests include computer vision, object recognition, mobile robot navigation, and content-based image retrieval.



Daniel Huttenlocher received the B.S. degree from the University of Michigan, Ann Arbor, in 1980 and the M.S. degree in 1984 and the Ph.D. degree in 1988 from the Massachusetts Institute of Technology, Cambridge.

He has served as a consultant or visiting scientist at several companies, including Schlumberger, Hewlett-Packard, and Hughes Aircraft. He is currently an associate professor in the Department of Computer Science at Cornell University, Ithaca, NY, and a Principal Scientist at Xerox PARC. His research interests are in computer vision, image analysis, document processing, and computational geometry. He has published over 50 articles in professional journals and conferences and holds 10 US patents. Dr. Huttenlocher received a Presidential Young Investigator Award from the National Science Foundation. He has also received recognition for his commitment to undergraduate education, including being named the 1993 Professor of the Year in New York State by the Washington, DC-based Council for the Advancement and Support of Education, and receiving Cornell's top teaching honor, the Weiss Presidential Fellowship in 1996. He was associate editor of the *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* from 1991 to 1995 and is program co-chair of the 1997 *IEEE Conference on Computer Vision and Pattern Recognition*.