Proceedings of the International Symposium on Computer Vision, pages 91-96, 1995.

Recognition by Matching Dense, Oriented Edge Pixels

Clark F. Olson and Daniel P. Huttenlocher
Computer Science Department
Cornell University
Ithaca, NY 14853
clarko@cs.cornell.edu,dph@cs.cornell.edu

Abstract

This paper describes techniques to perform efficient and accurate recognition in difficult domains by matching dense, oriented edge pixels. We model three-dimensional objects as the set of two-dimensional views of the object. Translation, rotation, and scaling of the views are allowed to approximate full three-dimensional motion. A modified Hausdorff measure is used to determine which transformations of each object model are reported as matches. The use of dense, oriented edge pixels allows us to achieve a low rate of false positives. Techniques to prune the search space are used to obtain a system that is efficient in practice. We give results of the system recognizing object views in intensity and infrared images.

1 Introduction

Much recent work on object recognition has focused on matching sparse feature points in the object model and in the image. Analysis has shown that relying on such feature points implies that false positives will occur in images with moderate complexity [4, 5, 6, 9]. In addition, many objects are difficult to model by sparse feature points and these points can be difficult to locate robustly in images. Recent work has investigated matching dense edge maps using the Hausdorff distance [7, 8]. Since this representation retains more information from the object, false positives are less likely.

For some applications, we have small object models and/or complex images, and even the use of dense edge maps is not sufficient to rule out false positives reliably. This paper discusses techniques to improve matching performance by using additional local information. In particular, we consider matching dense pixels with associated orientations. The use of such information reduces the number and size of false positives considerably. Analysis that determines the probability that a false positive will be found when using these techniques can be found in [10]. This information can also improve the speed of such recognition tasks by helping to prune the search space.

We use matching techniques that find close matches between sets of dense, two-dimensional points, yet we are interested in detecting objects with considerable three-dimensional structure. To accomplish this, we consider two-dimensional views of the three-dimensional objects from multiple viewpoints and match the object views to the images. Our search strategy explicitly models two-dimensional translation and rotation, and scaling of the object.

In the next section, we will describe how we measure the quality of a match between the object model and the image. This measure is a modification of the Hausdorff distance that allows orientation information to be included in a natural manner. We will then describe, in Section 3, our search strategy to efficiently find positions of object models composed of dense, oriented edge pixels that align closely with the image. Section 4 discusses the performance of the system, in terms of both speed and accuracy. Finally, Section 5 summarizes the paper.

2 Matching dense, oriented edge pixels

The directed Hausdorff distance between two point sets, M and I, 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 to its nearest point in set I. In the context of object recognition, the Hausdorff distance is used to measure the quality of a match between an object model and an image. If M is the set of (transformed) object model points and I is the set

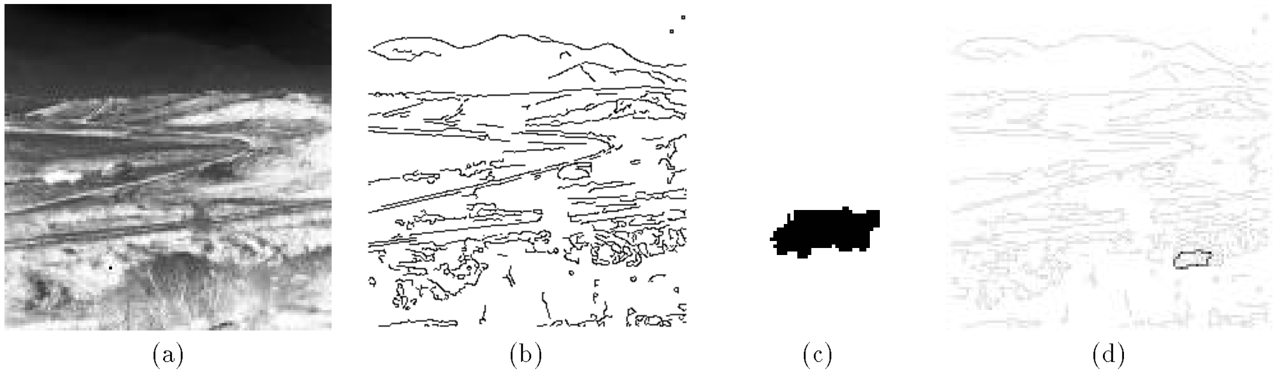

Figure 1: An example showing where false positives occur in practice due to dense edge pixels. (a) A FLIR image after histogram equalization. (b) The edges found in the original image. (c) An object view. (d) A false positive matching the entire object model to image pixels with $\delta = 1$.

of image edge points, the directed Hausdorff distance measures the distance of the worst matching object point to its closest image point. Of course, due to occlusion, we cannot assume that each object point appears in the image. We are thus interested in the partial distance between these sets, given by:

$$h_K(M, I) = \underset{m \in M}{K^{\text{th}}} \min_{i \in I} ||m - i|| \qquad (1)$$

This measures the Hausdorff distance among the K object points that are closest to image points. We can set K to be the minimum number of object points that we expect to find in the image if the object model is present or we can set K such that the probability of a false positive match occurring at random is small.

Typically, we are interested in whether there exists a match of size K with Hausdorff distance 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. The statement $h(M, I) < \delta$ is equivalent to $M \subset (I \oplus E_\delta)$, where E_δ is the area in the image that could match an object point at the origin.

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

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 I by E_δ and probe the result at each of the points in M. Each time the probe hits a point in the dilated image, a match has been found. We simply sum these matches and determine if the result surpasses K.

When we have a combination of a small set of object features and a complex image, this measure can still yield a considerable number of false positives, particularly when the transformation space is large. Figure 1 shows a case where examining the directed Hausdorff distance might lead us to believe that a match is present where it is not in FLIR (Forward Looking Infra-Red) imagery. The dense edges caused by the texture at some locations in the image can lead to false positive object model matches. These problems can be solved, in part, by using orientation information in addition to spatial information in determining the proximity between points in the transformed object model and the image.

We can generalize the Hausdorff distance to use orientations by considering each edge point, 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 direction of the gradient at the point. Typically, we are concerned with edge points on a pixel grid and, thus, the x and y values fall into discrete sets. We can map the orientations into a discrete set in a similar manner. Let's call a set of image points that have been extended in this fashion an *oriented image edge map*, I_o, and similarly, let's call a such an extended set of points in the object model an *oriented model edge map*, M_o.

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

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

This has the same general form as the previous Hausdorff measure, but we now measure the distance between two points by taking the maximum of the distances in translation and orientation. In this measure, α is a normalization factor that makes the orientation values comparable with the spatial values.

Requiring a match to have $h_\alpha(M,I) \leq \delta$ constrains the matching points to be close both spatially and in orientation. We can set the parameters α and δ arbitrarily to adjust the required proximities. Partial distances can also be computed as in Equation 1.

Considering this as a set containment problem yields the following as the error volume with which we must dilate I_o prior to probing:

$$E_{\alpha,\delta} = \left\{ t \mid \left|\left| \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right|\right| \leq \delta, |t_o| \leq \alpha\delta \right\}.$$

If we choose the discretized orientations such that $\alpha = 1$ and use the L_∞ norm, this simplifies to:

$$E_\delta = \{t \mid ||t||_\infty \leq \delta\}.$$

We can now probe $I_o \oplus E$ at the transformed locations of the pixels in the oriented model edge map to determine if a match is present.

3 Search Strategy

Recent work has shown that efficient approximate methods can be formulated to compute the minimum Hausdorff distance between sets of points by discretizing the space of possible transformations of the model points. We'll discuss how such methods work in general before describing the extension to oriented points.

Chamfer matching [1, 3] is an edge matching technique that minimizes the sum of the distances from each object edge point to its closest image edge point. This technique is closely related to minimizing the generalized Hausdorff measure, which instead selects the Kth largest of these minimum distances. Paglieroni [11, 12] considered methods to speed up chamfer matching by probing a distance transform of the image at the transformed object edge points. The distance transform measures the distance of each location in the image from an edge point and can be computed efficiently using a two-pass algorithm [13, 2, 11]. If the sum of the distance transform probes of each of the object points at some transformation is large enough, then we can rule out not only the current transformation, but also many transformations close to it.

Huttenlocher and Rucklidge [7, 8, 14] use similar techniques in conjunction with Hausdorff matching. We now consider the distance transform of the dilated image. If the Kth largest probe into this distance transform is 0, then we have found a match of size K. Otherwise, the Kth largest probe yields the distance to the closest possible position of the object model that could produce a match of size K.

Let's consider which transformations this allows us to rule out. To do this we must first decide how the transformation space will be discretized. To ensure that we do not miss any good matches, we discretize the transformation space such that two adjacent transformations move any object pixel by at most 1 pixel (Euclidean distance). Note that if we want the coarsest possible discretization that maintains this property, the discretization will be dependent upon the object model being considered. Now, if d is the value of the Kth largest probe, we can rule out any transformation that does not map any object pixel to an image location that is at least distance d from where the current transformation maps it[1].

Now, since our oriented object and image points are three-dimensional vectors, we require a three-dimensional distance transform, but there is a question as to how the distance should be measured in the orientation direction. We would like treat each orientation plane independently, but since rotations of the object model may change the orientation of some object pixels, we cannot do this if we wish to rule out neighboring transformations that vary in rotation. To avoid this problem, we treat each rotation of the object model independently (essentially as separate object models). Now, since none of the transformations we would like to rule out change the orientation of the object model, we can treat each of the orientation planes of the distance transform independently.

Using the techniques described above, we can now formulate an algorithm to perform efficient recognition. We simply start at some location in the dis-

[1] This certainly includes all transformations within a city-block distance of $d-1$ of the current transformation in the discretized transformation space, but usually includes more transformations depending on the L_p norm used and the transformation space.

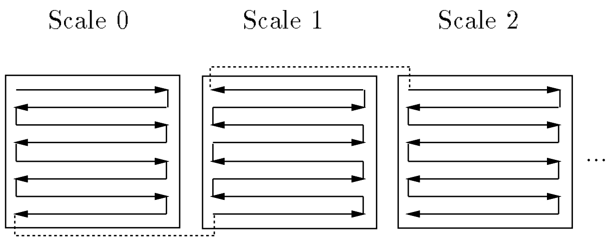

Figure 2: A snake-like search strategy is used.

cretized transformation space and probe the distance transform at the locations of the transformed object pixels. The results of these probes allow us to rule out some subset of the transformation space. We then proceed to the next transformation (in some ordering) that has not been ruled out and repeat this process, until each transformation has either been probed or ruled out.

The ordering that we use to examine transformations in the translation+scale space is snake-like, in an effort to do few probes. See Figure 2. Each time a discrete transformation is examined, irrespective of whether it is probed or it is ruled out by a previous probe, the local bound on the distance to closest possible transformation that could produce a match is decremented and propagated to all of the neighboring transformation cells in both translation and scale. At each of these neighbors, the propagated value is compared against the bound already present and the larger of the two is saved. This method of propagation requires a map to be kept in memory of the current bounds of all of the translations for the current scale and the next scale. The snake-like ordering guarantees that when we use this propagation technique, no transformation is probed when it should have been ruled out by the probe of a previous transformation.

It may possible to achieve further speedup by selecting the transformations to probe in some nonconnecting fashion to allow less probes to be performed. (Paglieroni [11], for example, discusses a scheme to do this in conjunction with chamfer matching.) The disadvantages to this method are that the full propagation of the bounds resulting from a probe must be performed immediately and more storage space is required since a map of the full transformation space must be kept in memory. Alternately, a hierarchical cell decomposition method could be used [8, 14], but this technique has not been explored for this system.

To reduce overhead, we use a two-level hierarchy of transformations. For each scale and rotation, the transformations on the translation axes are divided into 3×3 blocks. At first, only the center of each of these blocks is examined (in the snake-like ordering). As long as the distance at the center of the block is greater than one, we do not need to examine any of the neighboring transformations. Any time the distance is less than or equal to one, we subsequently probe the neighbors of the transformation. We could use larger blocks or a taller hierarchy to further reduce overhead.

4 Performance

Figure 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, 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. The largest match found, in fact, was a false positive.

The current implementation of these techniques uses 16 discrete orientations and $\delta = \alpha = 1$ (each orientation bucket thus corresponds to $\frac{2\pi}{16}$ radians, but matches are allowed in neighboring buckets, also). In these experiments, we limited the allowable orientation and scale change of the object views to 10° and 10%, respectively.

Figure 4 shows another example in a complex indoor scene. In this case, the object model was extracted from a frame in an image sequence and we are matching it to a later frame in the sequence. Figure 4(d) shows the position of the object detected when orientation information was used. No false positives were found in this case. When orientation information was not used, several false positives were found. Figure 4(e) shows some of the false positives that yielded larger matches than the correct position of the object.

The use of orientation information has lowered the size of the maximum false positive found considerably. When orientation information was not used in 40 trials with object models and images similar to those found in Figure 3, a false positive accounting for the entire object model was found in 36 of the 40 cases, with an average of 99.7% of the object model accounted for by the image. When orientation information was used in these same trials, the largest false positive accounted for, on average, 79.9% of the object model.

The actual running time of the system is good. The preprocessing stage requires approximately 7 seconds

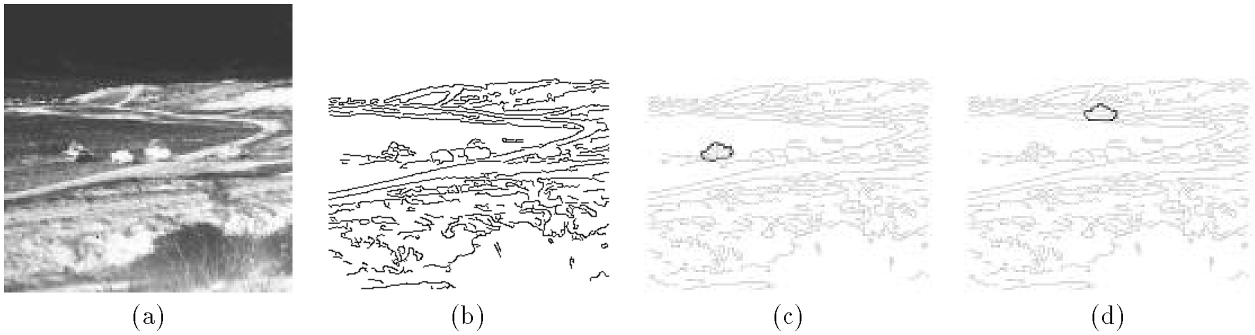

Figure 3: Automatic target recognition example. (a) The FLIR image after histogram equalization. (b) The edges found in the image. (c) The largest scoring position near the correct location of the target. (d) The largest scoring position over the entire image.

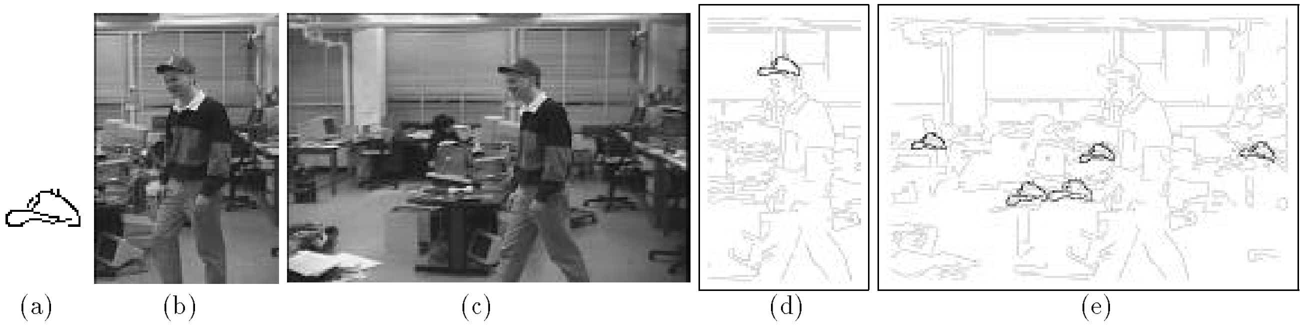

Figure 4: Image sequence example. (a) The object model. (b) Part of the image frame that the object model was extracted from. (c) The image frame in which we are matching. (d) The position of the object located using orientation information. No false positives were found for this case. (e) Several false positives that were found when orientation information was not used.

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 performs the distance transform on each orientation plane of the oriented image edge map. This step needs to be 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 2 seconds to 6 seconds. See Table 1 for some examples. Note that the largest value observed was for a case with a large object model (95 pixels) and with a small threshold (60 pixel matches). Most trials required less than 4 seconds per object view.

In addition to reducing the false alarm rate, the use of orientation information has significantly improved the speed of matching. We can see in Table 1, that, in a small sample of the trials, the number of transformations that are probed is reduced by approximately an order of magnitude. The running times per model reported in Table 1 are reduced less due to overhead.

Overall, these techniques have considerably reduced the size and number of false positives found and at the same time reduced the number of transformations that must be considered.

5 Summary

This paper has discussed matching techniques for sets of dense, oriented edge pixels. The use of such techniques allows recognition in domains that would previously have yielded too many false positives. We have given an extension of the Hausdorff distance that allows the matching of sets of oriented points. Using this measure, we have formulated a search strategy that allows us to find the transformations (translation, rotation, and scale) that match some minimum

			Using orientations			No orientations		
	Points	*Threshold*	*Probes*	*Time*	*Biggest*	*Probes*	*Time*	*Biggest*
Sample	67	53	63K	3.0s	63	709K	7.8s	67
FLIR	67	60	25K	2.2s	62	350K	4.5s	67
images	95	60	160K	6.4s	65	1411K	17.9s	95
	95	76	45K	3.7s	-†	703K	9.7s	95
Intensity Image	123	98	39K	5.1s	99	588K	11.2s	120

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

Table 1: Performance comparison. *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 positive found.

number of the object points closely to the image in this new measure. Experiments have confirmed that this strategy not only produces far fewer false positive matches, but is considerably faster than when orientations are not considered.

Acknowledgments

This work was supported in part by ARPA under ARO contract DAAH04-93-C-0052 and by National Science Foundation PYI grant IRI-9057928 .

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 *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.

[2] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[3] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1998.

[4] W. E. L. Grimson and D. P. Huttenlocher. On the sensitivity of geometric hashing. In *Proceedings of the International Conference on Computer Vision*, pages 334–338, 1990.

[5] W. E. L. Grimson, D. P. Huttenlocher, and T. D. Alter. Recognizing 3d objects from 2d images: An error analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 316–321, 1992.

[6] W. E. L. Grimson, D. P. Huttenlocher, and D. W. Jacobs. A study of affine matching with bounded sensor error. *International Journal of Computer Vision*, 13(1):7–32, 1994.

[7] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[8] D. P. Huttenlocher and W. J. Rucklidge. A multiresolution technique for comparing images using the Hausdorff distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 705–706, 1993.

[9] C. F. Olson. Time and space efficient pose clustering. Technical Report UCB//CSD-93-755, Computer Science Division, University of California at Berkeley, July 1993.

[10] C. F. Olson and D. P. Huttenlocher. Determining the probability of a false positive when matching chains of oriented pixels. In *Proceedings of the ARPA Image Understanding Workshop*, pages 1175–1180, 1996.

[11] D. W. Paglieroni. Distance transforms: Properties and machine vision applications. *CVGIP: Graphical Models and Image Processing*, 54(1):56–74, January 1992.

[12] D. W. Paglieroni, G. E. Ford, and E. M. Tsujimoto. The position-orientation masking approach to parametric search for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:740–747, July 1994.

[13] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13:471–494, 1966.

[14] W. J. Rucklidge. *Efficient Computation of the Minimum Hausdorff Distance for Visual Recognition.* PhD thesis, Cornell University, 1995.