Proceedings of the International Symposium on Computer Vision, pages 449-454, 1995.

On the Speed and Accuracy of Object Recognition When Using Imperfect Grouping

Clark F. Olson
Computer Science Department
Cornell University
Ithaca, NY 14853
clarko@cs.cornell.edu

Abstract

This paper analyzes the improvements that can be gained in object recognition through the use of simple, imperfect grouping techniques. We consider, in particular, the pose clustering method of object recognition. Simple grouping techniques are described that determine pairs of points that are connected in the image edge map. We show that such grouping techniques can considerably improve both the speed and accuracy of object recognition. Experiments are described that demonstrate the improvements in performance.

1 Introduction

Object recognition methods for complex problems have been plagued by poor speed and accuracy. The primary cause of both of these problems is image clutter. Such clutter requires considerable computation to process and causes false positives to be found. Solutions to these problems have been only partially successful. Two methods that have been useful are grouping and indexing. Grouping methods attempt to determine which features in an image are part of a single object. Indexing methods determine which sets of model features may match these sets of image features. These methods can be powerful when they are used together [2].

Many studies have analyzed the power of indexing, by itself, as a means to reduce the search space for object recognition [2, 6, 9, 11, 15]. It has been demonstrated that, when we attempt to recognize objects in the presence of noise, feature indexing systems index, on average, a constant fraction of all of the possible matches for a particular set of image features. Indexing thus does not reduce the computational complexity of recognition in the presence of noise.

Grouping, on the other hand, can reduce the computational complexity of object recognition. Grimson [5] has shown that the performance of a particular constrained search system is exponential in the problem size if we have spurious features, but it is low-order polynomial if the data is known to have all come from a single object. Of course, the requirement that all of the data comes from a single object implies the need for a perfect grouping system, unless we limit ourselves to very simple problems. This paper shows that even simple, imperfect grouping techniques can improve the computational complexity of object recognition. In addition, such grouping can considerably improve the accuracy of recognition by decreasing the false alarm rate.

Many cues have been used to perform grouping of image features. Some of the examples include parallelism, proximity, colinearity, connectivity, convexity, symmetry, and closure. This paper will consider only simple grouping mechanisms that find pairs of points that are likely to belong the same object. While proximity, or even color or texture, can provide such information, we concentrate on using connectivity of feature points in the image edge map.

We use these grouping techniques to improve the pose clustering method described in [13], which recognizes three-dimensional objects from a single view in two-dimensional images. A detailed analysis is given to determine the complexity and accuracy of the system when these grouping techniques are used. A comparison against the analysis for the case where grouping is not used indicates that grouping substantially improves the system in both regards.

The following section will describe the pose clustering method of object recognition and the framework that will be used in this paper. Section 3 will discuss the methods used to perform grouping. We will then discuss the computational complexity and accu-

```
Function recognize(input: model-points, image-points)
  Repeat:
    Choose two random image points, ν1 and ν2.
    For all pairs of model points, μ1 and μ2:
      For all point matches, (μ3, ν3):
        Determine the poses aligning the group
        match, γ = {(μ1, ν1), (μ2, ν2), (μ3, ν3)}.
      Find and output clusters among these poses.
End
```

Figure 1: Efficient pose clustering algorithm.

racy achieved when using grouping in Sections 4 and 5, respectively. Section 6 describes experiments that were performed using this system on real images. Finally, Section 7 summarizes the paper.

2 Recognition framework

Pose clustering is an object recognition technique based on the generalized Hough transform [1]. The key is that the pose of an object can be determined (almost) uniquely from a small set of feature matches between the model and the image. For the case of three-dimensional object feature points and two-dimensional image feature points, the number of matches that are required is three [3, 10]. The correct matches should yield poses close to the correct pose of the object. If we determine the poses corresponding to all possible matches (since we don't know which are correct in advance), then a large cluster in the pose space indicates the likely position of the object. But, for the problem of recognizing three-dimensional objects from intensity images, there are $O(m^3n^3)$ such matches, where m is the number of model points and n is the number of image points.

We have previously shown that if the clustering operation finds exactly those poses that bring some number of matches between sets of model and image points into alignment up to some error criterion, then pose clustering has optimal accuracy for point matching [13]. Approximate algorithms that do not achieve optimality are used for efficiency reasons. We have also shown that the computational complexity of pose clustering can be reduced to $O(mn^3)$ and that the space efficiency can be improved through the use of decomposition and randomization techniques. A summary of this system follows.

Figure 1 gives the algorithm that is used in [13]. This algorithm considers subproblems where a pair of *distinguished points*, (ν_1, ν_2), are selected that must be correct model points for the algorithm to succeed. If we consider each possible pair of distinguished points, this algorithm would perform essentially equivalently to the conventional pose clustering method. This would require $O(m^3n^3)$ time. Randomization is used to reduce the number of pairs of distinguished points that we need to examine to approximately $\frac{n^2}{(fm)^2} \ln \frac{1}{\delta}$, where f is the minimum fraction of the object that we require to appear in the image to recognize it and δ is the error rate that we allow. We then consider each of the $2m(m-1)$ permutations of possibly matching model points, (μ_1, μ_2). For each of these, we determine the poses aligning the matches (μ_1, ν_1), (μ_2, ν_2) and each of the $(m-2)(n-2)$ additional point matches, (μ_3, ν_3). Approximately $\frac{n^2(n-2)(m-2)}{f^2} \ln \frac{1}{\delta}$ poses are computed overall. Since the clustering step is performed in linear time using recursive histograming techniques, the overall time required by this algorithm is $O(mn^3)$.

An error analysis [12] showed that the probability that a bin in the pose space yields a false positive of size K through the random accumulation of feature points is approximately:

$$p \approx \left(\frac{bmn}{1+bmn} \right)^K$$

where b is the average fraction of pose space that brings a model point into alignment with an image point up to some error criterion. If we set our recognition threshold to $K = fm$, then the most image points that we can tolerate with false alarm rate γ is:

$$n \approx \frac{f}{b \ln \frac{1}{\gamma}}$$

3 A simple grouping mechanism

Grouping image features into sets that are likely to come from the same object can considerably improve the speed and quality of object recognition. It is possible for such techniques to not only distinguish points that may come from the same object, but also to reduce the search within a single object, by only producing certain subsets of features among all possibilities.

This work uses a simple grouping mechanism to determine which pairs of feature points are likely to belong to the same object. The criterion used to determine whether two feature points should be grouped

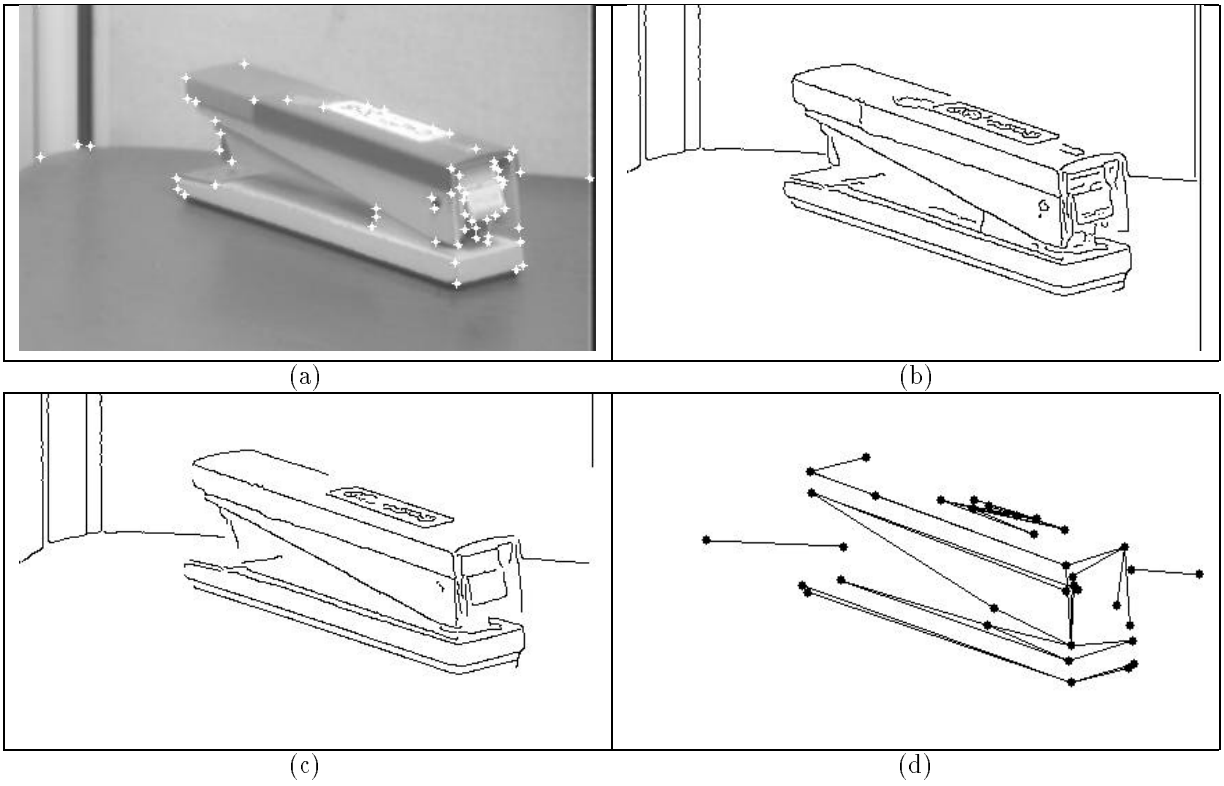

Figure 2: The grouping process. (a) The corners in the original image. (b) The edges of the original image. (c) The lines detected in the edge image using Hough transform techniques. (d) The groups detected in the image.

is whether they are connected[1] in the image edge map. Since the objects used in these experiments were largely polyhedral, we used the heuristic that the edges connecting the feature points should be straight. Of course, this is not necessary for correct grouping operation. A second heuristic, requiring that the points be some minimum distance apart to form a group, was used for two reasons. First, this excluded many unmodeled groups that were formed due to areas of high image texture. Second, groups that are close together produce unstable results in the pose estimation method and are thus unlikely to be useful in any case.

The first step in the grouping process is to determine the feature points in the image. A fast interest operator [4] is used detect corner points in the image. Next, the edges are detected. Straight lines in the edge map are determined using the Hough transform techniques described in [14]. Finally, the corners that lay close to the straight lines are detected and groups formed.

Figure 2 shows an example of the grouping process on an image of a stapler. In this case, 38 groups were found involving 34 of the 63 image feature points.

4 Computational Complexity

Let's now consider the computational complexity of object recognition using pose clustering when grouping is used to determine which pairs of distinguished points to use.

The number of groups found by grouping systems is typically linear in the number of image points. For grouping based on connectivity in the edge map, we are guaranteed a linear number of such groups if we have polyhedral objects. Let's thus assume we have $\alpha_i n$ image groups and $\alpha_m m$ model groups. (For trihedral objects, we have $\frac{1}{2} \leq \alpha_i \leq \frac{3}{2}$ and $\frac{1}{2} \leq \alpha_m \leq \frac{3}{2}$, if we eliminate points from consideration if they are not

[1] We use this term loosely. A method is used that allows small gaps between colinear edges to be bridged.

part of one the groups.)

If f is the fraction of the model points appearing in the image, then the expected fraction of correct groups appearing in the image is at least f^2. The actual fraction should be larger, since pairs of points that are grouped are more likely to be either both occluded or unoccluded than random pairs of points. The probability that k trials will be unsuccessful is then:

$$p \leq \left(1 - \frac{f^2 \alpha_m m}{\alpha_i n}\right)^k$$

since there are expected to be $f^2 \alpha_m m$ correct model groups among the $\alpha_i n$ image groups.

We can set this probability to be less than some arbitrarily small constant δ and solve for the number of trials necessary to achieve this accuracy:

$$\left(1 - \frac{f^2 \alpha_m m}{\alpha_i n}\right)^k \leq \delta$$

$$k \geq \frac{\ln \delta}{\ln 1 - \frac{f^2 \alpha_m m}{\alpha_i n}} \approx \frac{\alpha_i n}{f^2 \alpha_m m} \ln \frac{1}{\delta}$$

For each image group, we consider each of the model groups as a possible match. For each such match, we then determine and cluster the poses corresponding to all of the matches between three image points and three model points that match the image group to the model group. This yields an $O(mn^2)$ algorithm. The number of poses computed is approximately $\frac{\alpha_i n(n-2)(m-2)}{f^2} \ln \frac{1}{\delta}$, a speedup of approximately $\frac{n}{\alpha_i}$ over the version that did not use grouping to select good distinguished points.

5 Accuracy

This section will consider the probability that a false positive will be found when using these imperfect grouping techniques and compare it with the probability when no grouping techniques are used.

Previous work [7, 8, 12] has used the Bose-Einstein occupancy model to estimate the probability of finding a false positive at some point in pose space in various recognition problems. This analysis can be modified for the case at hand by considering the probability that points in pose space are both consistent with a match between grouped features and consistent with sufficient additional matches to result in a false positive. If the locations of the individual model and image features are independent then these two probabilities are also independent. So, we can compute the probability, P_{fp_K}, of a false positive of size K at some point in pose space as the product of the probability, P_g, that the pose is compatible with one of the group matches of size 2 and the probability, $P_{r_{K-2}}$, that the remaining matches accumulate to a random match of size $K-2$.

$$P_{fp_K} = P_g P_{r_{K-2}}$$

First consider P_g. Let's approximate the probability distribution of poses such that the distribution of transformed model features is uniform in the image. This implies that the probability, P_1, that a pose will bring a pair of model features into alignment with a pair of image features to within an error of ϵ is:

$$P_1 = \left(\frac{\pi \epsilon^2}{WH}\right)^2$$

where W and H are the width and height of the image in pixels.

If there are $\alpha_i n$ image groups and $\alpha_m m$ model groups then we have:

$$P_g \approx (1 - (1 - P_1)^{\alpha_m \alpha_i mn}) \approx \frac{\alpha_m \alpha_i mn \pi^2 \epsilon^4}{W^2 H^2}$$

Now we must consider whether $K-2$ additional matches are brought into alignment by the pose. Previous work [12] has indicated that this can be approximated by

$$p \approx \left(\frac{bmn}{1 + bmn}\right)^x$$

where $b \approx \frac{\pi \epsilon^2}{WH}$ is the fraction of pose space aligning a single model and image point and $x = K - 2$ is the size of the false positive. We thus have:

$$P_{r_{K-2}} \approx \left(\frac{mn}{\frac{WH}{\pi \epsilon^2} + mn}\right)^{K-2}$$

And:

$$P_{fp_K} \approx \left(\frac{mn}{\frac{WH}{\pi \epsilon^2} + mn}\right)^{K-2} \frac{\alpha_m \alpha_i mn \pi^2 \epsilon^4}{W^2 H^2}$$

Table 1 gives a comparison of the estimated probability of a false positive for a problem with $m = 30$, $n = 150$ and $WH = 65536$ for the case with no grouping and the case with grouping. The probability of a false positive is much lower when grouping is used. False positives should occur approximately 230 times more frequently when grouping is not used for $\epsilon = 3$ and approximately 49 times more frequently when $\epsilon = 5$.

	No grouping		Grouping	
f	$\epsilon = 3$	$\epsilon = 5$	$\epsilon = 3$	$\epsilon = 5$
1.0	$3.86 \cdot 10^{-6}$	$6.08 \cdot 10^{-3}$	$1.67 \cdot 10^{-8}$	$1.24 \cdot 10^{-4}$
.75	$8.74 \cdot 10^{-5}$	$2.18 \cdot 10^{-2}$	$3.77 \cdot 10^{-7}$	$4.45 \cdot 10^{-4}$
.50	$1.97 \cdot 10^{-3}$	$7.80 \cdot 10^{-2}$	$8.50 \cdot 10^{-6}$	$1.59 \cdot 10^{-3}$

Table 1: The estimated probability of a false positive for the case with no grouping and with grouping. For this problem: $m = 30$, $n = 150$, $WH = 65536$, $\alpha_i = \alpha_m = 1.5$.

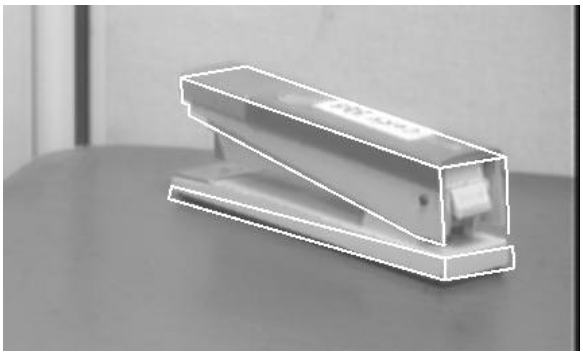

Figure 3: The recognized position of the stapler.

6 Experiments

This system has been tested on several of the same images as the original pose clustering system [13] to verify the improved performance. Figure 3 shows the recognized position of the stapler from Figure 2. Figure 4 gives an additional example of the recognition process.

The running time of the recognition algorithm on these examples was between 6 and 10 minutes per object on a Sparc-5, depending on the complexity of the object model and image. The previous implementation required several hours to run on a Sparc-10. This improvement is not solely due to grouping, an improvement in the implementation also yielded a speedup. A better comparison to determine the speedup gained is the number of poses that were computed and clustered in each case. Table 1 gives these numbers for the objects recognized in Figure 4. When using grouping, the system examines less than 1% of the poses examined when grouping is not used in these cases.

No significant false positives were found when the grouping techniques were used. Some instances occurred where a correct group match produced hypotheses that included several correct and incorrect feature matches, but the same groups produced better matches involving predominantly correct matches.

object	N_{ng}	N_g	$\frac{N_{ng}}{N_g}$
widget	$2.56 \cdot 10^8$	$2.52 \cdot 10^6$	101.5
plane	$3.70 \cdot 10^8$	$3.54 \cdot 10^6$	104.7
hammer	$2.56 \cdot 10^8$	$2.52 \cdot 10^6$	101.5
person	$2.11 \cdot 10^8$	$2.09 \cdot 10^6$	100.9

Table 2: Number of poses computed with no grouping (N_{ng}) and with grouping (N_g).

7 Summary

This paper has shown that even simple, imperfect grouping techniques can yield a considerable improvement in both the speed and accuracy of object recognition. Grouping techniques based on connectivity in the edge map were used to improve the pose clustering method of object recognition. The grouping techniques reduced the computational complexity of the algorithm and experiments validated that the running time was reduced considerably in practice. Furthermore, the analysis indicates that the rate of false positives is considerably reduced through the use of these techniques.

References

[1] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] D. T. Clemens and D. W. Jacobs. Space and time bounds on indexing 3-d models from 2-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007–1017, October 1991.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–396, June 1981.

[4] W. Förstner. Image matching. Chapter 16 of *Computer and Robot Vision*, Vol. II, by R. Haralick and L. Shapiro, Addison-Wesley, 1993.

[5] W. E. L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. *Artificial Intelligence*, 44(1-2):121–165, 1990.

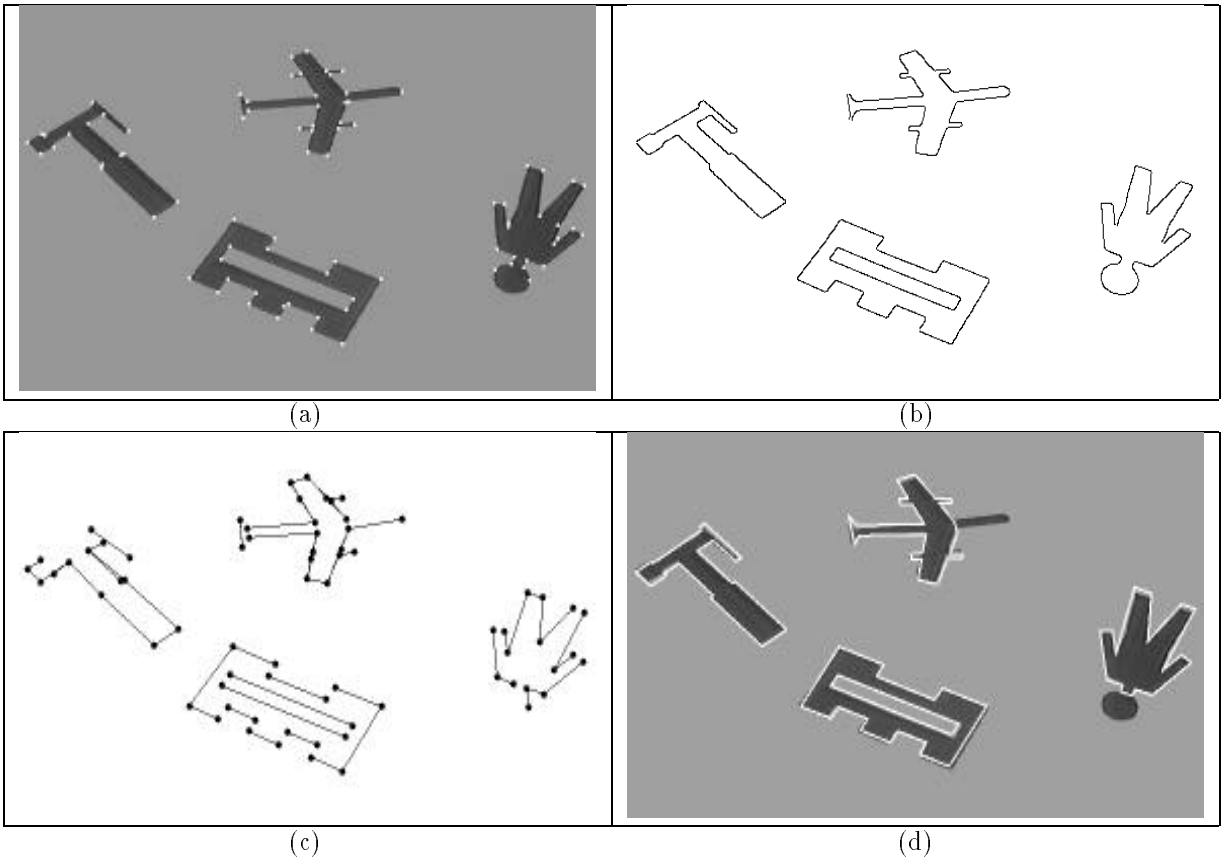

Figure 4: The recognition of two-dimensional figures. (a) The original image with corners highlighted. (b) The edges found in the image. (c) The groups found in the image. (d) The recognized positions of the figures.

[6] 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.

[7] W. E. L. Grimson and D. P. Huttenlocher. On the sensitivity of the Hough transform for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):255–274, March 1990.

[8] W. E. L. Grimson and D. P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1201–1213, December 1991.

[9] 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.

[10] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

[11] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Affine invariant model-based object recognition. *IEEE Transactions on Robotics and Automation*, 6(5):578–589, October 1990.

[12] 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.

[13] C. F. Olson. Time and space efficient pose clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 251–258, 1994.

[14] C. F. Olson. Improved curve detection through decomposition of the Hough transform. Technical Report 95-1516, Department of Computer Science, Cornell University, 1995.

[15] C. F. Olson. Probabilistic indexing for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):518–522, May 1995.