International Journal of Computer Vision 23(2), 131–147 (1997)

Efficient Pose Clustering Using a Randomized Algorithm*

CLARK F. OLSON
Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
clarko@cs.cornell.edu

Received February 23, 1995; Revised July 10, 1995; Accepted December 5, 1995

Abstract. Pose clustering is a method to perform object recognition by determining hypothetical object poses and finding clusters of the poses in the space of legal object positions. An object that appears in an image will yield a large cluster of such poses close to the correct position of the object. If there are m model features and n image features, then there are $O(m^3n^3)$ hypothetical poses that can be determined from minimal information for the case of recognition of three-dimensional objects from feature points in two-dimensional images. Rather than clustering all of these poses, we show that pose clustering can have equivalent performance for this case when examining only $O(mn)$ poses, due to correlation between the poses, if we are given two correct matches between model features and image features. Since we do not usually know two correct matches in advance, this property is used with randomization to decompose the pose clustering problem into $O(n^2)$ problems, each of which clusters $O(mn)$ poses, for a total complexity of $O(mn^3)$. Further speedup can be achieved through the use of grouping techniques. This method also requires little memory and makes the use of accurate clustering algorithms less costly. We use recursive histograming techniques to perform clustering in time and space that is guaranteed to be linear in the number of poses. Finally, we present results demonstrating the recognition of objects in the presence of noise, clutter, and occlusion.

1. Introduction

The recognition of objects in digital image data is an important and difficult problem in computer vision (Besl and Jain, 1985; Chin and Dyer, 1986; Grimson, 1990). Interesting applications of object recognition include navigation of mobile robots, indexing image databases, automatic target recognition, and inspection of industrial parts. In this paper, we investigate techniques to perform object recognition efficiently through pose clustering.

Pose clustering (also known as the generalized Hough transform) is a method to recognize objects from hypothesized matches between feature sets in the object model and feature sets in the image (Ballard, 1981; Stockman et al., 1982; Silberberg et al., 1984; Turney et al., 1985; Silberberg et al., 1986; Dhome and Kasvand, 1987; Stockman, 1987; Thompson and Mundy, 1987; Linnainmaa et al., 1988). In this method, the transformation parameters that bring the sets of features into alignment are determined. Under a rigid-body assumption, the correct matches will yield transformations close to the correct pose of the object. Objects can thus be recognized by finding clusters among these transformations in the pose space. Since we do not know which of the hypothesized matches are correct in advance, pose clustering methods typically examine the poses from all possible matches of some cardinality, k, where k is the minimum number of feature matches necessary to constrain the pose of the object to a finite set of possibilities, assuming non-degeneracy.

*This research has been supported by a National Science Foundation Graduate Fellowship, NSF Presidential Young Investigator Grant IRI-8957274 to Jitendra Malik, and NSF Materials Handling Grant IRI-9114446. A preliminary version of this work appears in (Olson, 1994).

We will focus on the recognition of general three-dimensional objects undergoing unrestricted rotation and translation from single two-dimensional images. To simplify matters, the only features used for recognition are feature points in the model and the image. It should be noted, however, that these results can be generalized to any problem for which we have a method to estimate the pose of the object from a set of feature matches.

If m is the number of model feature points and n is the number of image feature points, then there are $O(m^3n^3)$ transformations to consider for this problem, assuming that we generate transformations using the minimal amount of information. We demonstrate that, if we are given two correct matches, performing pose clustering on only the $O(mn)$ transformations that can be determined from these correct matches using minimal information yields equivalent performance to clustering all $O(m^3n^3)$ transformations, due to correlation between the transformations. Since we do not know two correct matches in advance, we must examine $O(n^2)$ such initial matches to ensure an insignificant probability of missing a correct object, yielding an algorithm that requires $O(mn^3)$ total time. This is the best complexity that has been achieved for the recognition of three-dimensional objects from feature points in single intensity images. When additional information is present, as is typical in computer vision applications, additional speedup can be achieved by using grouping to generate likely initial matches and to reduce the number of additional matches that must be examined (Olson, 1995).

An additional problem with previous pose clustering methods is that they have required a large amount of memory and/or time to find clusters, due to the large number of transformations and the size of pose space. Since we now examine only $O(mn)$ transformations at a time, we can perform clustering quickly using little memory through the use of recursive histograming techniques.

The remainder of this paper is structured as follows. Section 2 discusses some previous techniques used to perform pose clustering. Section 3 proves that examining small subsets of the possible transformations is adequate to determine if a cluster exists and discusses the implications of this result on pose clustering algorithms. Section 4 discusses the computational complexity of these techniques. Section 5 gives an analysis of the frequency of false positives, using the results on the correlation between transformations to achieve more accuracy than previous work. Section 6 describes methods by which clustering can be performed efficiently. Section 7 discusses the implementation of these ideas. Experiments that have been performed to demonstrate the utility of the system are presented in Section 8. Section 9 discusses several interesting issues pertaining to pose clustering. Finally, Section 10 describes previous work that has been done in this area and a summary of the paper is given in Section 11.

2. Recognizing Objects by Clustering Poses

As mentioned above, pose clustering is an object recognition technique where the poses that align hypothesized matches between sets of features are determined. Clusters of these poses indicate the possible presence of an object in the image. We will assume that we are considering the presence of a single object model in the image. Multiple objects can be processed sequentially.

To prevent a combinatorial explosion in the number of poses that are considered, we want to use as few as possible matches between image and model points to determine the hypothetical poses of the object. It is well known that matches between three model points and three image points is the smallest number of non-degenerate matches that yield a finite number of transformations that bring three-dimensional model points into alignment exactly with two-dimensional image points using the perspective projection or any of several approximations (Fischler and Bolles, 1981; Huttenlocher and Ullman, 1990; DeMenthon and Davis, 1992; Alter, 1994). See Fig. 1. If we know the center of projection and focal length of the camera, we can use the perspective projection to model the imaging process accurately. Otherwise, an approximation such

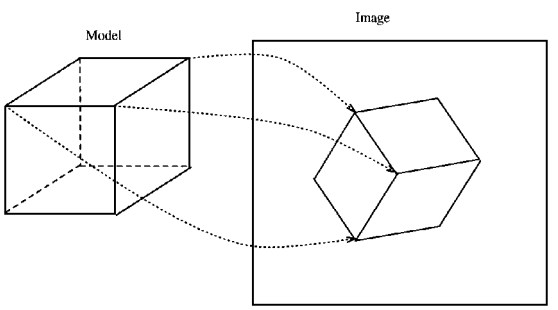


Figure 1. There exist a finite number of transformations that align three non-colinear model points with three image points.

as weak-perspective can be used. Weak-perspective is accurate only when the distance of the object from the camera is large compared to the depth variation within the object. In either case, pose clustering algorithms can use matches between three model points and three image points to determine hypothetical poses.

Let us call a set of three model features, $\{\mu_1, \mu_2, \mu_3\}$, a *model group* and a set of three image points, $\{\nu_1, \nu_2, \nu_3\}$, an *image group*. A hypothesized matching of a single model feature to an image feature, $\pi = (\mu, \nu)$, will be called a *point match* and three point matches of distinct image and model features, $\gamma = \{(\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3)\}$, will be called a *group match*.

If there are m model features and n image features, then there are $6\binom{m}{3}\binom{n}{3}$ distinct group matches (since each group of three model points may match any group of three image points in six different ways), each of which yields up to four transformations that bring them into alignment exactly. Most pose clustering algorithms find clusters by histograming the poses in the multi-dimensional transformation space (see Fig. 2). In this method, each pose is represented by a single point in the pose space. The pose space is discretized into bins and the poses are histogramed in these bins to find large clusters. Since pose space is six-dimensional for general rigid transformations, the discretized pose space is immense for the fineness of discretization necessary to perform accurate pose clustering.

Two techniques that have been proposed to reduce this problem are coarse-to-fine clustering (Stockman et al., 1982) and decomposing the pose space into orthogonal subspaces in which histograming can be performed sequentially (Dhome and Kasvand, 1987; Thompson and Mundy, 1987; Linnainmaa et al., 1988). In coarse-to-fine clustering (see Fig. 3), pose space is quantized in a coarse manner and the large clusters found in this quantization are then histogramed in a more finely quantized pose space. Pose space can also be decomposed such that clustering is performed in two or more steps, each of which examines a projection of the transformation parameters onto a subspace of the pose space (see Fig. 4). The clusters found in a projection of the pose space are subsequently examined with respect to the remaining transformation parameters.

These techniques can lead to additional problems. The largest clusters in the first clustering step do not necessarily correspond to the largest clusters in the entire pose space. We could examine all of the bins in the first space that contain some minimum number of transformations, but Grimson and Huttenlocher (1990) have shown that for cluttered images, an extremely large number of bins would need to be examined due to saturation of the coarse or projected histogram. In addition, we must either store the group matches that

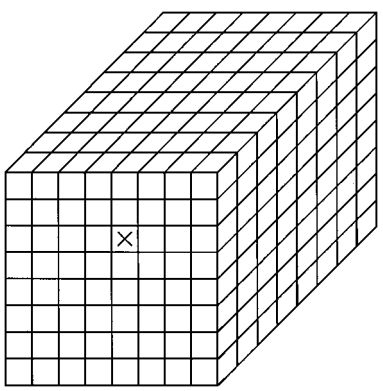


Figure 2. Clusters representing good hypotheses are found by performing multi-dimensional histograming on the poses. This figure represents a coarsely quantized three-dimensional pose space.

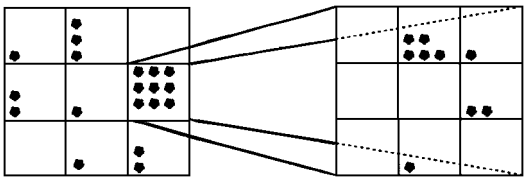


Figure 3. In coarse-to-fine histograming, the bins at a coarse scale that contain many transformations are examined at a finer scale.

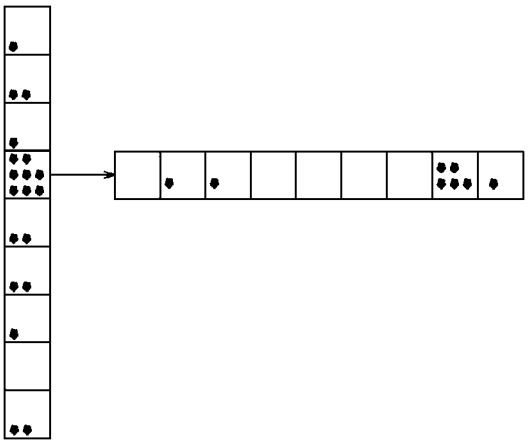


Figure 4. Pose space can be decomposed into orthogonal subspaces. Histograming is then performed in one of the decomposed subspaces. Bins that contain many transformations are examined with respect to the remaining subspaces.

contribute to a cluster in each bin (so that we can perform the subsequent histograming steps on them) or we must reexamine all of the group matches (and redetermine the transformations aligning them) for each subsequent histograming step. The first possibility requires much memory and the second requires considerable extra time.

We will see that these problems can be solved through a decomposition of the pose clustering problem. Furthermore, randomization can be used to achieve a low computational complexity with a low rate of failure. Similar techniques in the context of transformation equivalence analysis can be found in (Cass, 1993).

3. Decomposition of the Problem

Let Θ be the space of legal model positions. Each $p \in \Theta$ can be considered a function, $p : \mathcal{R}^3 \to \mathcal{R}^2$, that takes a model point to its corresponding image point. Each group match, $\gamma = \{(\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3)\}$, yields some subset of the pose space, $\theta(\gamma) \subset \Theta$, that brings each of the model points in the group match to within the error bounds of the corresponding image point. We will consider a generalization of this function, $\theta(\gamma)$, that applies to sets of point matches of any cardinality.

Let's assume that the feature points are localized with error bounded by a circle of radius ϵ (though the following analysis is not dependent on any choice of error boundary). We can then define $\theta(\gamma)$ as follows:

Definition.

$$\theta(\gamma) \equiv \{p \in \Theta : \| p(\mu_i) - \nu_i \|_2 \leq \epsilon, \text{for } 1 \leq i \leq |\gamma|\}$$

The following theorem is the key to showing that we can examine several small subproblems and achieve equivalent performance to examining the original pose clustering problem.

Theorem 1. *The following statements are equivalent for each* $p \in \Theta$:

1. *There exist* $g = \binom{x}{3}$ *distinct group matches that pose* p *brings into alignment up to the error bounds. Formally*,

$$\exists \gamma_1, \ldots, \gamma_g \text{ s.t. } p \in \theta(\gamma_i) \quad \text{for } 1 \leq i \leq g.$$

2. *There exist* x *distinct point matches,* $\pi_1, \ldots, \pi_x$, *that pose* p *brings into alignment up to the error bounds*:

$$\exists \pi_1, \ldots, \pi_x \text{ s.t. } p \in \theta(\{\pi_i\}) \quad \text{for } 1 \leq i \leq x.$$

3. *There exist* $x - 2$ *distinct group matches sharing some pair of point matches that pose* p *brings into alignment up to the error bounds*:

$$\exists \pi_1, .., \pi_x \text{ s.t. } p \in \theta(\{\pi_1, \pi_2, \pi_i\}) \quad \text{for } 3 \leq i \leq x.$$

Proof: The proof of this theorem has three steps. We will prove (a) Statement 1 implies Statement 2, (b) Statement 2 implies Statement 3, and (c) Statement 3 implies Statement 1. Therefore the three statements must be equivalent.

(a) Each of the group matches is composed of a set of three point matches. The fewest point matches from which we can choose $\binom{x}{3}$ group matches is x. The definition of $\theta(\gamma)$ guarantees that each of the individual point matches of any group match that is brought into alignment are also brought into alignment. Thus each of these x point matches must be brought into alignment up to the error bounds.

(b) Choose any two of the point matches that are brought into alignment. Form all of the $x-2$ group matches composed of these two point matches and each of the additional point matches. Since each of the point matches is brought into alignment, each of the group matches composed of them also must be from the definition of $\theta(\gamma)$.

(c) There are x distinct point matches that compose the $x - 2$ group matches, each of which must be brought into alignment. Any of the $\binom{x}{3}$ distinct group matches that can be formed from them must therefore also be brought into alignment. □

This theorem implies that we can achieve equivalent performance to the examining all of the group matches when we examine subproblems in which only those group matches that share some pair of correct point matches are considered. So, instead of finding a cluster of size $\binom{x}{3}$ among all of the group matches, we simply need to find a cluster of size $x - 2$ within any set of group matches that all share some pair of point matches. Furthermore, it is clear that any pair of correct point matches can be used. For each such pair, we

```
1.  Pose-Clustering(M, I):      /* M is the model point set.
                                   I is the image point set. */
2.      Repeat k times:
3.          Choose two random image points ν1 and ν2.
4.          For all pairs of model points μ1 and μ2:
5.              For all point matches (μ3, ν3):
6.                  Determine the poses aligning the group
                        match γ = {(μ1, ν1), (μ2, ν2), (μ3, ν3)}.
7.              End-for
8.              Find and output clusters among these poses.
9.          End-for
10.     End-repeat
11. End
```

Figure 5. The new pose clustering algorithm.

must examine $O(mn)$ group matches, since there are $(m-2)(n-2)$ group matches for a single pair of point matches such that no feature is used more than once. Of course, examining just one pair of image points will not be sufficient to rule out the appearance of an object in an image since there may be image clutter. We could simply examine all $2\binom{n}{2}\binom{m}{2}$ possible pairs of point matches, but we will see in the next section that we can examine $O(n^2)$ pairs of matches and achieve a low rate of failure.

Figure 5 gives the updated pose clustering algorithm.

4. Computational Complexity

This section discusses the computational complexity necessary to perform pose clustering using the techniques described above. We can use a randomization technique similar to that used in RANSAC (Fischler and Bolles, 1981) to limit the number of initial pairs of matches that must be examined. A random pair of image points is chosen to examine as the initial image points. All pairs of point matches that include these image points are examined, and, if one of them leads to recognition of the object, then we may stop. Otherwise, we continue choosing pairs of image points at random until we have reached a sufficient probability of recognizing the object if it is present in the image. Note that once we have examined this number of pairs of image points, we stop, regardless of whether we have found the object, since it may not be present in the image.

If we require fm model points to be present in the image to ensure recognition, we can determine an upper bound on the probability of not choosing a correct pair of image points in k trials, where each trial consists of examining a pair of image points at random. (We allow $(1-f)m$ model points to be absent as the result of occlusion by other objects, self-occlusion, or being missed by the feature detector; f is the fraction of model points that must appear.) Since the probability of a single image point being a correct model point is at least $\frac{fm}{n}$ in this case, the maximum probability of a pair being incorrect is approximately $1-(\frac{fm}{n})^2$. Thus, the probability that k random trials will all be unsuccessful is approximately:

$$p \leq \left(1-\left(\frac{fm}{n}\right)^2\right)^k$$

If we require the probability of a false negative to be less than δ we have:

$$\left(1-\left(\frac{fm}{n}\right)^2\right)^k \leq \delta$$

$$k \geq \frac{\ln \delta}{\ln\left(1-\left(\frac{fm}{n}\right)^2\right)}$$

Note that the minimum k that is necessary is $O(\frac{n^2}{m^2})$ since, $k_{\min}$ approaches $\frac{n^2}{(fm)^2}\ln\frac{1}{\delta}$ as $(fm/n)^2$ approaches zero[1].

For each pair of image points, we must examine each of the $2\binom{m}{2}$ permutations of model points which may match them. So, in total, we must examine $O(\frac{n^2}{m^2})\cdot O(m^2) = O(n^2)$ pairs of point matches to achieve the success rate $1-\delta$. Since we halt after k trials, regardless of whether we have found the object, this is the number of trials we examine in the worst-case, and is independent of whether the object appears in the image. The time bound varies with only the logarithm of the desired success rate, so very high success rates can be achieved without greatly increasing the running time of the algorithm. Since we must examine $O(mn)$ group matches for each pair of point matches, this method requires $O(mn^3)$ time per object in the database in the worst case, if we perform clustering in linear time, where previously $O(m^3n^3)$ time was required.

5. Frequency of False Positives

While the above analysis has been interpreted in terms of the "correct" clusters, so far, it also applies to false positive clusters. Let t be our threshold for the number

of model points that must be brought into alignment for us to output a hypothesis. If a pose clustering system that examines all of the poses finds a false positive cluster of size $\binom{t}{3}$, we would expect the new techniques to yield a false positive cluster of size $t-2$. We will thus find false positives with the same frequency as previous pose clustering systems.

Grimson et al. (1992) analyze the pose clustering approach to object recognition to estimate the probability of a false match having a large peak in transformation space for the case of recognition of three-dimensional objects from two-dimensional images. They use the Bose-Einstein occupancy model (see, for example, Feller, 1968) to estimate this probability. This analysis assumes independence in the locations of the transformations, which is not correct. Consider two group matches composed of a total of six distinct point matches. If there is some pose, $p \in \Theta$, that brings both group matches into alignment up to the error conditions, then any of the $\binom{6}{3}$ group matches that can be formed using the six point matches is also brought into alignment by this pose. The poses determined from these group matches are thus highly correlated.

Theorem 1 indicates that we will find a false positive only in the case where there is a pose that brings t model points into alignment with corresponding image points. This result allows us to perform a more accurate analysis of the likelihood of false positive hypotheses. We'll summarize the results of Grimson et al. before describing modifications to their analysis that account for the correlations between transformations and achieve more accuracy.

The Bose-Einstein occupancy model yields the following approximation of the probability that a bin will receive l or more votes due to random accumulation:

$$p_{\geq l} \approx \frac{\lambda^l}{(1+\lambda)^{-l}}$$

In this equation, λ is the average number of votes in a single bin (including redundancy due to uncertainty in the image). In the work of Grimson et al., $\lambda = 6\binom{m}{3}\binom{n}{3}b_g \approx \frac{m^3 n^3 b_g}{6}$, where b_g is the average fraction of bins that contain a pose bringing a particular group match into alignment (called the redundancy factor), m is the number of model features, and n is the number of image features. Each correct object is expected to have $\binom{fm}{3} \approx \frac{(fm)^3}{6}$ correct transformations, since each distinct group of model features will include the correct bin among those it votes for. The probability that an incorrect point match will have a cluster of at least this size is:

$$q \approx \left(\frac{\lambda}{1+\lambda}\right)^{\frac{(fm)^3}{6}}$$

Setting $q \leq \delta$ and solving for n, they find that the maximum number of image features that can be tolerated without surpassing the given error rate, δ, is:

$$n_{\max} \approx \frac{f}{\sqrt[3]{b_g \ln \frac{1}{\delta}}}$$

Grimson et al. have determined overestimates on the size of the redundancy factor, b_g, necessary for various noise levels to ensure that the correct bin is among those voted for by an image group using a bounded error model and they have used this to compute sample values of $n_{\max}$.

As noted above, this analysis can be made more accurate by considering the correlations between the transformations. Theorem 1 indicates that there exists some point, p, in transformation space that brings $\binom{fm}{3}$ group matches into alignment if and only if there are fm point matches that p brings into alignment. So, we must determine the likelihood that there exists a point in transformation space that brings into alignment fm of the nm point matches. We'll call the average fraction of transformation space that brings a single point match into alignment b_p.

If we otherwise follow the analysis of Grimson et al., we have $\lambda = b_p mn$ and we expect a correct pose to yield fm matches. Using the Bose-Einstein occupancy model we can estimate the probability of a false positive of this size:

$$p \approx \left(\frac{b_p mn}{1+b_p mn}\right)^{fm}$$

We can set $p \leq \delta$ and solve for n as follows:

$$\left(\frac{b_p mn}{1+b_p mn}\right)^{fm} \leq \delta$$

$$fm \ln\left(1+\frac{1}{b_p mn}\right) \geq \ln\frac{1}{\delta}$$

Using the approximation: $\ln(1+\alpha) \approx \alpha$, for small α, we have:

$$\frac{fm}{b_p mn} \geq \ln\frac{1}{\delta}$$

In fact, $\frac{1}{b_p mn}$ is not always small, but this approximation yields a conservative estimate for n.

$$n \leq \frac{f}{b_p \ln \frac{1}{\delta}}$$

Note that this is not very different from the result derived by Grimson et al. since $b_p \approx \sqrt[3]{b_g}$. The primary difference is a change from a factor of $\sqrt[3]{\ln \frac{1}{\delta}}$ to $\ln \frac{1}{\delta}$, which means that the new estimate of the allowable number of image features before a given rate of false positives is produced is lower than that obtained by Grimson et al.

It should be noted that this result is a fundamental limitation of all object recognition systems that use only point features to recognize objects, not of this system alone. Any time there exists a transformation that brings fm model points into alignment with image points, a system dealing only with feature points should recognize this as a possible instance of the object.

Some possible solutions to this problem are to use grouping or more descriptive features. The results presented here are easily generalized to encompass such information, if a method exists to estimate the pose from a set of matches between such features. This will increase the allowable clutter, but a similar result will still be applicable.

The primary implication of this result is that we should not assume that large clusters in the pose space necessarily imply the presence of the modeled object. We should use pose clustering as a method of finding likely hypotheses for further verification. As an additional verification step, we could, for example, verify the presence of edge information in the image as is done by Huttenlocher and Ullman (1990).

6. Efficient Clustering

This section discusses methods to perform clustering of the poses in time and space that is linear in the number of poses. This is accomplished through the use of recursive histograming techniques. Each hypothetical position of the model that is determined from a group match is represented by a single point in pose space. We use overlapping bins that are large enough to contain most, if not all, of the transformations consistent with the bounded error. This prevents clusters from being missed due to falling on a boundary between bins. This method is able to find clusters containing most of the correct transformations, but it does not have optimal accuracy.

An alternate method that could be used for complex or very noisy images, where false positives could prove problematic, is to sample carefully selected points in the pose space (see, for example, (Cass, 1988)) and determine which matches are brought into alignment by each sampled point. This alternative will find no cases where the matches in a cluster are not mutually consistent, but at a lower speed and at the risk of missing a cluster due to the sampling rate. Another alternative (Cass, 1992) determines regions of the pose space that are equivalent with respect to the matches they bring into alignment and that bring a large number of such matches into alignment. Such a method can achieve optimal accuracy in the sense that it can find all partitions of the pose space that bring some minimum number of matches into alignment. However, this appears difficult for the case of three-dimensional object undergoing rigid transformations since the legal poses do not form a vector space. Note that the analysis of the previous sections still applies to these methods.

When histograming is used to find clusters, either coarse-to-fine clustering or decomposition of the pose space should be used, since the six-dimensional pose space is immense. Let's consider the decomposition approach here. The pose space can be decomposed into the six orthogonal spaces corresponding to each of the transformation parameters. To solve the clustering problem, histograming can be performed recursively using a single transformation parameter at a time. In the first step, all of the transformations are histogramed in a one-dimensional array, using just the first parameter. Each bin that contains more than $fm - 2$ transformations is retained for further examination, where f is the predetermined fraction of model features that must be present in the image for us to recognize the object. (Let us for the moment neglect the possibility that not all of the correct poses may be found. In this case, if fm model points are present in the image, a correct pair of point matches will yield $fm - 2$ correct transformations.) For each bin with enough transformations, we recursively cluster the poses in that bin using the remaining parameters. Since this procedure continues until all six parameters have been examined, the bins in the final step contain transformations that agree closely in all six of the transformation parameters and thus form a cluster in the complete pose space.

```
1.  Find-Clusters(P, Π):        /* P is the set of poses. Π is
                                   the set of pose parameters. */
2.      If |Π| > 0 then
3.          Choose some π ∈ Π.
4.          Histogram poses in P by parameter π.
5.          For each bin, b, in the histogram:
6.              If |b| > fm − 2 then
7.                  Find-Clusters({p ∈ P : p ∈ b}, Π\π);
8.              End-if
9.          End-for
10.     Else
11.         Output the cluster location.
12.     End-if
13. End
```

Figure 6. The recursive clustering algorithm.

This method can be formulated as a depth-first tree search. The root of the tree corresponds to the entire pose space and each node corresponds to some subset of the pose space. The leaves correspond to individual bins in the six-dimensional pose space. At each level of the tree, the nodes from the previous level are expanded by histograming the poses in those nodes using a previously unexamined transformation parameter. The tree has height six, since there are six pose parameters to examine. At each level, we can prune every node of the tree that does not correspond to a volume of transformation space containing at least $fm-2$ transformations.

Figure 6 gives an outline of this algorithm. If unexamined parameters remain at the current branch of the tree, we histogram the remaining poses using one of these parameters. Each of the bins that contains at least $fm-2$ poses is then clustered recursively using the remaining parameters. The other bins are pruned. When we reach a leaf (after all of the parameters have been examined) that contains enough poses, we output the location of the cluster.

Although this decomposition of the clustering algorithm has not previously been formulated as a tree search, the analysis of Grimson and Huttenlocher (1990) implies that previous pose clustering methods saturate such decomposed transformation spaces at the levels of the tree near the root, due to the large number of transformations that need to be clustered. For those methods, virtually none of the branches near the root of the tree can be pruned.

Since previous systems would cluster $O(m^3n^3)$ transformations, there are $O(n^3)$ bins that could hold as many as $\binom{fm}{3}$ transformations at each level of the tree. Thus, despite histograming in a high-dimensional space, these systems may have a large number of unpruned bins at even low levels of the tree, since they are clustering so many transformations. Using the techniques presented here, we can have only $O(n)$ bins that contain as many as $fm-2$ transformations at any level of the tree, since there are $O(mn)$ transformations clustered at a time. This means that there are only $O(n)$ unpruned bins at each level. Thus, we do not have saturation at any level of the tree for this system. $O(mn)$ time and space is required per clustering step.

7. Implementation

This section describes our implementation of the techniques described in the previous sections of this paper. Of course, in general, we follow the algorithm given in Fig. 5.

Recall that the analysis of Section 4 showed that we need to examine

$$k \geq \frac{\ln \delta}{\ln\left(1-\left(\frac{fm}{n}\right)^2\right)}$$

pairs of random image points to achieve probability $1-\delta$ that we examine a pair from the model, if fm model points appear in the image. Now, since we do not use a perfect clustering system, we cannot assume that each correct pair of point matches will result in the implementation finding a cluster of the optimal size. The next section describes experiments determining how many we actually find. Knowing this, we can set a threshold on the number of matches necessary to output a hypothesis and a threshold on the number of trials necessary to achieve a low rate of failure. If we estimate that in pathological models and/or images, only 50% of the correct pairs of point matches will result in a cluster that surpasses this threshold, then we have:

$$k_{\min} = \left\lceil \frac{\ln \delta}{\ln\left(1-\frac{1}{2}\left(\frac{fm}{n}\right)^2\right)} \right\rceil$$

For each pair of random image points that we examine, we consider each pair of model points that may match them. We then form the $(m-2)(n-2)$ distinct group matches that contain them. For each such group match, we use the method of Huttenlocher and Ullman (1990) to determine the transformation parameters that bring three model points into alignment with three image points in the weak-perspective imaging model. Each group match yields two transformations, and the parameters of these transformations are stored in a preallocated array, since we know in advance how

many we will have. The use of this method makes the implicit assumption that weak-perspective is an accurate approximation to the actual imaging process for the problems we consider. This has been demonstrated to be true for the case when the depth within the object is small compared to the distance to the object (Thompson and Mundy, 1987). However, this does introduce error into our pose estimates. If we know the center of projection and focal length of our camera, we can use the full perspective projection to eliminate this source of error.

We find clusters among the poses using the recursive histograming techniques of the previous section. The order in which the parameters are examined is: scale, then translation in x and y, and then the three rotational parameters. Changing the order of the parameters has no effect on the clusters found and little effect on the running time.

We use overlapping bins to avoid missing clusters that fall on cluster boundaries. Each parameter is divided into small bins and a sliding box that covers three consecutive bins is used to find clusters. The size of the bins is changed with varying image noise levels, but the number of bins used in each dimension typically varies from 30 to 200. For each bin, we maintain a linked list of pointers to the transformations that fall into the bin and an associated count of the number of such transformations. This allows us to easily perform the recursive binning steps on subsequent parameters once the initial binning steps have been performed. At each position of the sliding box, the poses in the box are recursively clustered only if the number of transformations in the bins surpasses the threshold. When a cluster is found after considering all of the transformation parameters, the hypothetical pose of the object is estimated by averaging all of the poses in the cluster.

Once a cluster has been found, we use the method of Huttenlocher and Cass (1992) to determine an estimate of the number of consistent matches. They argue that the total number of matches in a cluster is not necessarily a good measure of the quality of the cluster, since different matches in the cluster may match the same image point to multiple model points, or vice versa, which we do not wish to allow. Huttenlocher and Cass recommend counting the lesser of the number of distinct model points and distinct image points matched in the cluster, since it can be determined quickly (as opposed to the maximal bipartite matching) and is reasonably accurate.

8. Results

This section describes experiments performed on real and synthetic data to test the system.

8.1. Synthetic Data

Models and images have been generated for these experiments using the following methodology:

1. Model points were generated at random inside a 200 $\times$ 200 $\times$ 200 pixel cube.
2. The model was transformed by a random rotation and translation and was projected using the perspective projection onto the image plane. The focal length that was used was the same as the distance to the center of the cube, which was approximately 10 times the depth within the object.
3. Bounded noise ($\epsilon = 1$ pixel) was added to each image point.
4. In some experiments, additional random image points were added.

The first experiment determined whether the correct clusters were found. Table 1 shows the performance of two methods at finding correct clusters. The first system uses the old method of clustering all of the poses simultaneously. The second system uses the new method of clustering only those poses from group matches sharing a pair of point matches. The old method finds much larger clusters, of course, since it clusters many more correct transformations, but the size of the incorrect clusters is expected to rise at the same rate. The new

Table 1. The performance in finding correct clusters.

	Old method			New method		
m	opt.	avg.	%	opt.	avg.	%
10	120	95.5	.796	8	6.64	.831
20	1140	882.2	.774	18	15.02	.834
30	4060	3046.9	.750	28	23.23	.830
40	9880	7400.8	.749	38	30.79	.810
50	19600	14569.9	.743	48	40.47	.843

We use the following terms in the above table:
m: the number of object points.
opt.: the size of the optimal cluster.
avg.: the size of the average cluster found.
%: the average fraction found of the optimal cluster.

Table 2. The size of false positive clusters found for objects with 20 feature points.

n	average	std. dev.	maximum
20	3.84	0.88	6
40	5.32	1.14	8
60	6.35	1.35	10
80	7.06	1.52	12
100	7.64	1.68	13
120	7.94	1.80	13
140	8.21	1.87	13
160	8.42	1.95	14
180	8.61	1.98	14
200	8.79	2.02	15

We use the following terms in the above table:
n: the number of image points.
average: the average size of the largest cluster found.
std. dev.: the standard deviation of the cluster size.
maximum: the largest cluster found overall.

techniques actually find a larger percentage of the correct poses in the best cluster. This is because these clusters are smaller. Since we examine only those group matches that share some pair of point matches, the noise associated with those two image points stays the same over the entire cluster. This noise may move the cluster from the true location, but it does not increase the expected size of the cluster, as it does when we examine all possible group matches, since each pose is computed using this same pair of points.

Experiments were run to determine the size of false hypotheses generated by the new method for models of 20 random model points and various image complexities. Table 2 shows the average size of the largest cluster found for each pair of image points, the standard deviation among these clusters, and the size of the largest cluster over all of the pairs of image points. Since the new method found correct clusters of average size 15.02 for models of twenty points and false positive clusters of average size 8.79 for 200 random image points, these levels of complexity do not cause a large number of false positives to be found.

An experiment determining the number of trials necessary to recognize objects in the presence of random extraneous image points was run. Table 3 shows the results of this experiment. To generate a hypothesis of the model being present in the image, this experiment required a cluster to be at least 80% of the optimal size (14 for models of size 20). For each value of n, Table 3 shows $k_{\min}$ for $\delta = 0.01$, the average number of trials necessary to generate a correct hypothesis that the object was present in the image, the maximum number of trials necessary to generate such a hypothesis, and the number of objects (out of 100) that required more than $k_{\min}$ trials. For each case, at least 98 of the 100 objects were recognized within $k_{\min}$ trials. Overall, 99.3 percent of the objects were recognized within $k_{\min}$ trials, with the expectation of recognizing $1 - \delta = 99.0$ percent of the objects.

Table 3. The number of trials required to find objects with 20 points.

n	$k_{\min}$	avg.	max.	over
20	6.65	1.51	11	2
40	34.52	5.28	20	0
60	80.65	14.50	165	2
80	145.20	25.24	270	1
100	228.19	33.39	223	0
120	329.61	51.70	412	1
140	449.47	55.86	280	0
160	587.77	109.97	2321	1
180	744.51	113.31	556	0
200	919.69	145.95	697	0

We use the following terms in the above table:
n: number of image points.
$k_{\min}$: expected number of trials necessary for $\delta = 1.0$.
avg.: average number of trials required for 100 objects.
max.: maximum number of trials required.
over: number of objects that required $> k_{\min}$ trials.

To summarize the results on synthetic data, the new pose clustering method has been determined to find a larger fraction of the optimal cluster than previous methods and to result in very few false positives for images of moderate complexity. In addition, the number of pairs of point matches that we must examine to recognize objects has been confirmed experimentally to be $O(n^2)$, validating the analysis that indicated the total time required by this algorithm is $O(mn^3)$.

8.2. Real Images

This pose clustering system has also been tested on several real images from two data sets. The first data set consists entirely of planar figures. The second consists of three-dimensional objects. Note that when applied to the first data set, this algorithm made no use of the fact that the figures were planar. No benefit is gained from using this data set, except that corners are easy to detect on them. Furthermore, the only features used in either data set to generate hypotheses are the locations of corner points in the image.

Hypothesis generation followed the following steps:

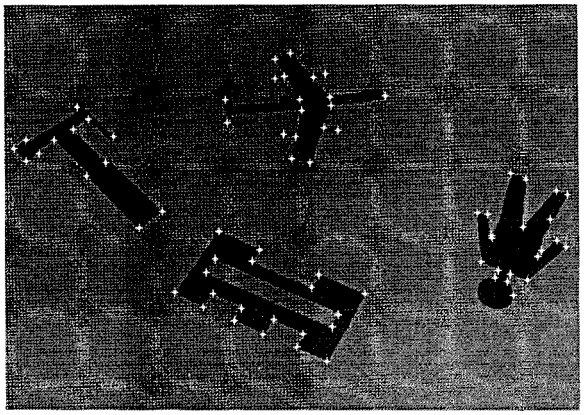
1. Object models were created. For the first data set this was done by capturing images of the object and determining the location of corners. For the second data set this was done by hand.
2. Images including the objects were captured.
3. Corners were detected in the images using a fast and precise interest operator (Förstner, 1993; Förstner and Gülch, 1987).
4. The model and image feature points were used by the pose clustering system to generate hypotheses as specified in the previous section.

Figure 7 shows an example of recognizing objects from the first data set in an image. Figure 7(a) shows the 84 feature points found by the interest operator. While there is no occlusion in this image, the interest operator did not find all of the correct corners. In several cases where two corners were close together (e.g., the engines on the plane) only one corner is found. Figure 7(b) shows the best hypotheses found for this image with the edges drawn in. The projected model edges line up very well with the object edges in the images. Figure 7(c) shows the largest incorrect match that was found for this image. This is a rotated and scaled version of the person model. For this pose of this model, several of the points in the model are brought very close to the corners detected in the image. When large false positives are found, they can be easily disambiguated from the correct hypotheses by examining whether the transformed model edges agree with edges in the image.

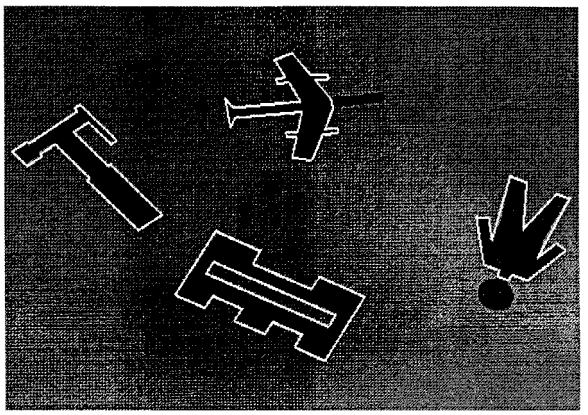
Several images from this data set included occluded objects. See, for example, Fig. 8. Despite the occlusion, we are able to find good hypotheses, since we only require some fraction, f, of the model points to appear in the image. The algorithm was still able to find the correct hypotheses for objects with up to 40% occlusion.

Figure 9 shows an example recognizing a stapler from the second data set. Figure 9(a) shows the 70 feature points detected in this image. Self-occlusion prevented many of the features points on the stapler from being found. In addition, a large number of spurious points were found due to shadows and unmodeled stapler points. Figure 9(b) shows the best hypothesis found.

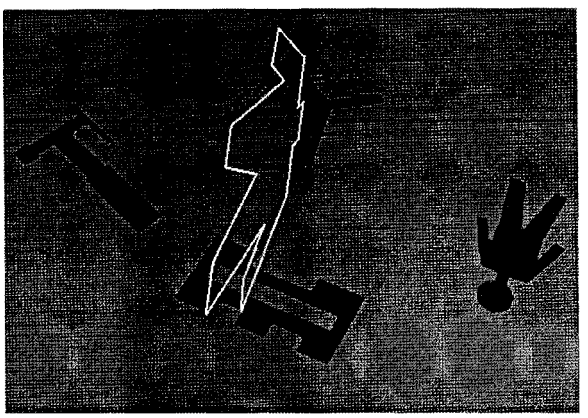
The largest source of error in the experiments on both real and synthetic images was the use of weak-perspective as the imaging model. The poor pose



(a)

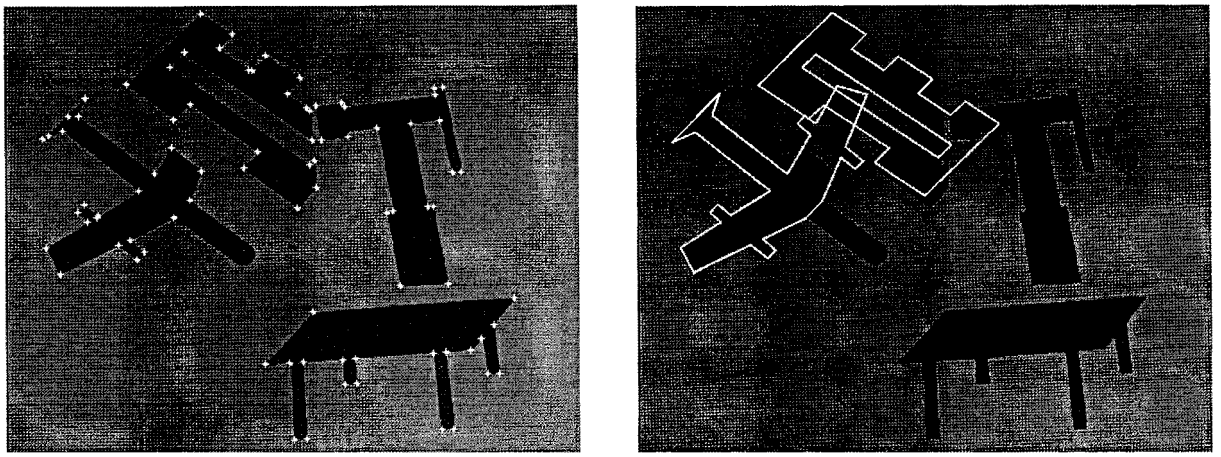


(b)



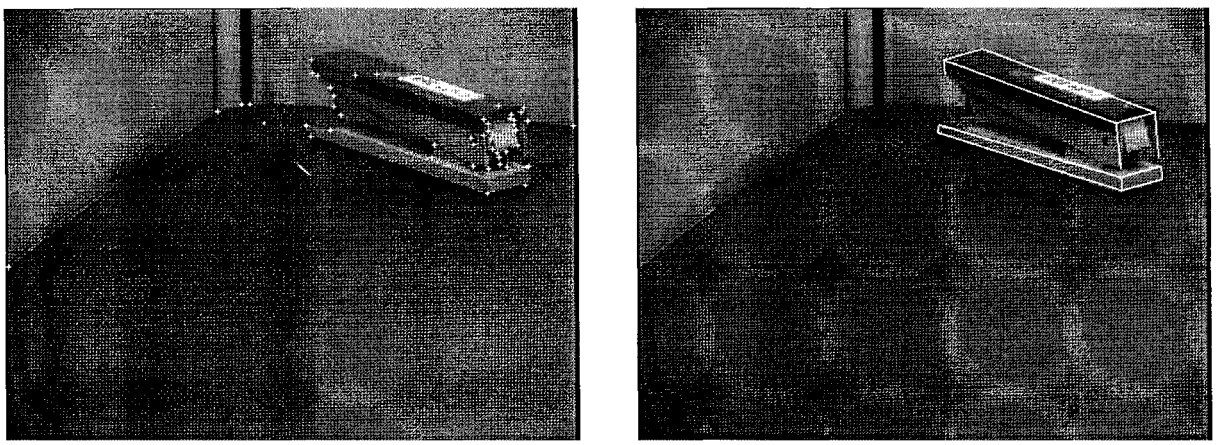
(c)

Figure 7. Recognition example for two-dimensional objects. (a) The corners found in an image. (b) The four best hypotheses found with the edges drawn in. (The nose of the plane and the head of the person do not appear because they were not in the models.) (c) The largest incorrect match found.



(a) (b)

Figure 8. Recognition example for occluded two-dimensional objects. (a) The corners found in an image. (b) The best hypotheses found for the occluded objects with the edges drawn in.



(a) (b)

Figure 9. Recognition example for a 3D object. (a) The features found in the image. (b) The best hypothesis found.

recovered in Fig. 10 demonstrates the problems that perspective distortion can cause. The use of weak-perspective is the limiting factor on the current accuracy of this system.

9. Discussion

The algorithm that has been described can be parallelized in a straightforward manner. We simply partition the subproblems such that each processor performs an approximately equal number of the subproblems. In this manner, the use of p processors yields a speedup of approximately p until p reaches the total number of subproblems. We thus require $O(mn)$ time on n^2 processors. We still require $O(mn)$ space on each processor. Further speedup might be achieved with $p > n^2$ by considering parallel histograming techniques.

Some of the techniques described in this paper can be used with recognition strategies other than pose clustering, when these strategies examine pose space to determine the transformations aligning several matches between features. For example, Breuel (1992) recursively subdivides the pose space to find volumes that are consistent with the most matches. These volumes are found by intersecting the subdivisions of pose space with bounded constraint regions arising from hypothesized matches between sets of model and image features. The expected time was empirically found to be linear in the number of constraint regions. To recognize three-dimensional objects from two-dimensional

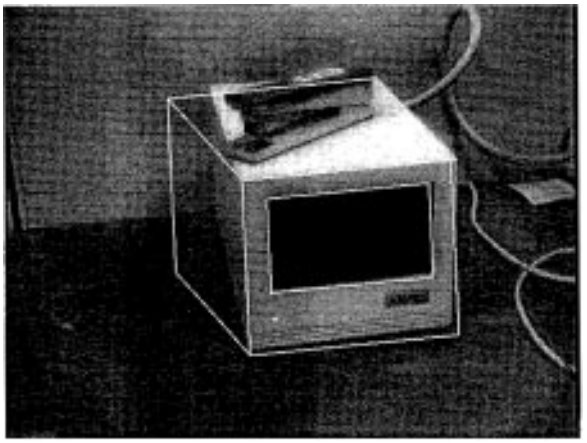


Figure 10. Perspective distortion can cause error in the recovered pose or even recognition failure when a weak-perspective model is used.

images using point features, matches of three points are necessary to generate bounded constraint regions. Thus, there are $O(m^3n^3)$ such constraint regions for this case. Theorem 1 implies that Breuel's algorithm will still find the best match if it examines only the $O(mn)$ constraint regions associated with a given pair of correct matches of feature points. Since we don't know two correct matches in advance, we must examine $O(n^2)$ of them (using randomization). Of course, this introduces a probability, δ, that a correct pair of point matches will not be chosen, and thus recognition may fail where it would not in the original algorithm.

Clustering methods other than histograming have been largely avoided due to their considerable time requirements. For example, algorithms based on nearest-neighbors (Sibson, 1973; Defays, 1977; Day and Edelsbrunner, 1984) require $O(p^2)$ time, where p is the number of points to cluster. Since there are $p = O(m^3n^3)$ transformations to cluster in previous methods, this means the overall time for clustering would be $O(m^6n^6)$. While most pose clustering methods have used histograming to find large clusters in pose space, less efficient, but more accurate, clustering methods become more feasible with this method, since only $O(mn)$ transformations are clustered at a time, rather than $O(m^3n^3)$.

Another point worthy of discussion is that some previous researchers in pose clustering have assumed that finding a large enough peak in the pose space is sufficient to consider the object present in the image, while others have claimed that pose clustering is more sensitive to noise and clutter than other algorithms. Grimson et al. (Grimson and Huttenlocher, 1990; Grimson et al., 1992) have shown that we should not simply assume that large clusters are instances of the object; additional verification is needed to ensure against false positives. However, while it is clear that further verification is required for hypotheses generated by pose clustering, other methods also require this additional verification step. The analysis in Sections 3 and 5 shows that pose clustering is not inherently more sensitive to noise and clutter than other algorithms.

Clutter affects the efficiency of pose clustering similarly to other algorithms. On the other hand, noise and other sources of error are handled in considerably different ways among various algorithms. While considerable research has gone into analyzing how to best handle error in the alignment method (Jacobs, 1991; Alter, 1993; Alter and Jacobs, 1994; Grimson et al., 1994), very little has been done in this regard for pose clustering. Work by Cass (1990, 1992) demonstrates how to handle noise exactly in the context of transformation equivalence analysis, for the case where the localization error is bounded by a polygon, but this is not directly applicable to pose clustering. At present, the system described here handles noise heuristically and further study in this area should be beneficial.

We can compare the noise sensitivity of pose clustering to generate-and-test methods such as alignment. While careful alignment (Grimson et al., 1992; Alter, 1993; Alter and Jacobs, 1994; Grimson et al., 1994) ensures that each of the additional point matches can separately be brought into alignment with the initial set of matches, up to some error bounds, by a single transformation, this transformation may be different for each such additional point match. (A different error vector may be assigned to the initial matches for each of the additional matches.) It does not guarantee that *all* of the additional point matches and the initial set of matches can be brought into alignment up to the error bounds by a single transformation. Ideally, a pose clustering system could guarantee this, but due to the limitations imposed by discretizing the pose space and the heuristic handling of noise, it is not achieved by this system. Interestingly, the analysis of Grimson et al. (1992) indicates that pose clustering techniques will find fewer false positives than the alignment method for similar levels of noise and clutter.

10. Related Work

This section describes previous work that has been performed on techniques related to those presented here.

Ballard (1981) showed that the Hough transform (Hough, 1962; Illingworth and Kittler, 1988) could be generalized to detect arbitrary two-dimensional shapes undergoing translation by constructing a mapping between image features and a parameter space describing the possible transformations of the object. This system was generalized to encompass rotations and scaling in the plane.

Stockman et al. (1982) describe a pose clustering system for two-dimensional objects undergoing similarity transformations. This system examines matches between image segments and model segments to reduce the subset of the four-dimensional pose space consistent with a hypothetical match to a single point. Clustering is performed by conceptually moving a box around pose space to determine if there is a position with a large number of points inside the box and is implemented by binning. The binning is performed in a coarse-to-fine manner to reduce the overall number of bins that must be examined.

Silberberg et al. (1984, 1986) describe a pair of systems using generalized Hough transform techniques to perform object recognition. In the first, they assume orthographic projection with known scale. Objects are modeled by straight edge segments. They solve for the best translation and rotation in the plane for each match between an image edge and a model edge for each viewpoint on a discretized viewing sphere and cluster these transformations. In the second, they consider the recognition of three-dimensional objects that lie on a known ground plane using a camera of known elevation. Matches between oriented feature points are used to determine the three remaining transformation parameters.

Turney et al. (1985) describe methods to recognize partially-occluded two-dimensional parts undergoing translation and rotation in the plane. A generalized Hough transform voting mechanism with votes weighted by a saliency measure is used to recognize the parts.

Dhome and Kasvand (1987) recognize polyhedra in range images using pairs of adjacent surfaces as features. Initially compatible hypotheses between such features in the model and in the image are determined and then clustering is performed hierarchically in three subsets of the viewing parameters: the view axis, the rotation about the view axis, and the model translation. Complete-link clustering techniques are used to determine clusters with some maximum radius in each stage. The clusters from earlier stages are considered separately in the later stages to ensure that the final clusters agree in all of the parameters.

Thompson and Mundy (1987) use *vertex-pairs* in the image and model to determine the transformation aligning a three-dimensional model with the image. Each vertex-pair consists of two feature points and two angles at one of the feature points corresponding to the direction of edges terminating at the point. At run-time, precomputed transformation parameters are used to quickly determine the transformation aligning each model vertex-pair with an image vertex-pair and binning is used to determine where large clusters of transformations lie in transformation space. In addition, Thompson and Mundy show that for objects far enough from the camera, the scaled orthographic projection (weak-perspective) is a good approximation to the perspective projection.

Linnainmaa et al. (1988) describe another pose clustering method for recognizing three-dimensional objects. They first give a method for determining object pose under the perspective projection from matches of three image and model feature points (which they call *triangle pairs*). They cluster poses determined from such triangle pairs in a three-dimensional space quantizing the translational portion of the pose. The rotational parameters and geometric constraints are then used to eliminate incorrect triangle pairs from each cluster. Optimization techniques are described that determine the pose corresponding to each cluster accurately.

Grimson and Huttenlocher (1990) show that noise, occlusion, and clutter cause a significant rate of false positive hypotheses in pose clustering algorithms when using line segments or surface patches as features in two- and three-dimensional data. In addition, they show that binning methods of clustering must examine a very large number of histogram buckets even when using coarse-to-fine clustering or sequential binning in orthogonal spaces.

Grimson et al. (1992) examine the effect of noise, occlusion, and clutter for the specific case of recognizing three-dimensional objects from two-dimensional images using point features. They determine overestimates of the range of transformations that take a group of model points to within error bounds of hypothetically corresponding image points. Using this analysis, they show that pose clustering for this case also suffers from a significant rate of false positive hypotheses. A positive sign for pose clustering from the work of Grimson et al. is that pose clustering

produced false positive hypotheses with a lower frequency than the alignment method (Huttenlocher and Ullman, 1990) when both techniques use only feature points to recognize objects.

Cass (1988) describes a method similar to pose clustering that uses transformation sampling. Instead of binning each transformation, Cass samples the pose space at many points within the subspaces that align each hypothetical feature match to within some error bounds. The number of features brought into alignment by each sampled point is determined and the object's position is estimated from sample points with maximum value. This method may miss a pose that brings many matches into alignment, but it ensures that the matches found for any single sample point are mutually compatible.

Another related technique is to divide pose space into regions that bring the same set of model and image features into agreement up to error bounds (Cass, 1992). For the two-dimensional case, if each image point is localized up to an uncertainty region described by a k-sided polygon, then each of the mn possible point matches corresponds to the intersection of k half-spaces in four-dimensions. The equivalence classes with respect to which model and image features are brought into agreement can be enumerated using computational geometry techniques (Edelsbrunner, 1987) in $O(k^4m^4n^4)$ time. The case of three-dimensional objects and two-dimensional images is more difficult since the transformations do not form a vector space. But, by embedding the six-dimensional pose space in an eight-dimensional space, it can be seen that there are $O(k^8m^8n^8)$ equivalence classes. Not all of the equivalence classes must be examined, particularly if approximate algorithms are used to find transformations that align many features. Several techniques to reduce the computational burden of these techniques are given in (Cass, 1993).

Breuel (1992) has proposed an algorithm that recursively subdivides pose space to find volumes where the most matches are brought into alignment. While this method has an exponential worst case complexity, Breuel's experiments provide empirical evidence that, for the case of two-dimensional objects undergoing similarity transformations, the expected time complexity is $O(mn)$ for line segment features (or $O(m^2n^2)$ for point features). The case of three-dimensional objects and two-dimensional data is not discussed at length, but if the expected running time remained proportional to number of constraint regions then it would be $O(m^3n^3)$ for point features.

11. Summary

This paper has described techniques to efficiently perform object recognition through the use of pose clustering. Of particular interest has been a theorem that shows that three different formalizations of the object recognition problem are equivalent, and thus they can be used interchangeably, assuming that other parameters are unchanged. This theorem has been used to show that object recognition using pose clustering can be decomposed into small subproblems that examine only the sets of feature matches that include some initial set of matches. Randomization has been used to limit the number of such subproblems that need to be examined. The overall time required for recognizing three-dimensional objects using feature points has been shown to be $O(mn^3)$ for m model features and n image features, the lowest known complexity for this problem. Since far fewer poses are clustered at a time, this method can be implemented using much less memory than previous pose clustering systems. The total space requirement is $O(mn)$.

An improved analysis on the rate of false positives that are expected for a given image complexity has been given. While the results indicate the rates are slightly worse than previously thought, analysis has shown that a fundamental bound exists on the rate of false positives that can be achieved by algorithms that recognize objects by finding sets of features that can be brought into alignment. Within the limitations of this bound, pose clustering performs well.

A new formalization of clustering using efficient histograming has been given. This formalization casts the recursive histograming of poses as a pruned tree search. Since there are $O(n)$ unpruned branches at each level of the tree, this method achieves time and space that is linear in the number of poses that are clustered.

Experiments have been described that have validated the performance of the system. The new techniques find a greater percentage of the poses that correspond to the correct cluster than previous techniques, when a correct pair of initial matches is used, and the size of false positives found in moderately complex images is small. It has been verified experimentally that the number of initial matches that must be examined to locate, with high probability, an object that is present in the image is $O(n^2)$, even when noisy features are considered. The largest source of error in the experiments arose from the use of weak-perspective as the imaging model, suggesting that its use is limiting the performance of object recognition algorithms in some cases.

The algorithm has considerable inherent parallelism and can be implemented on a parallel system simply by dividing the subproblems among available processors. It has been observed that the implications of the theorem showing the equivalence of several formalisms of the object recognition problem apply to alternate methods of recognition and can yield improvements even when pose clustering is not used. We conclude by noting again that, while we have considered primarily the problem of 3D from 2D recognition using feature points, these techniques are general in nature and can be applied to other recognition problem where we have a method for determining the hypothetical pose of an object from a set of feature matches.

Acknowledgments

This research was performed while the author was a graduate student at the University of California at Berkeley. The author thanks Jitendra Malik for his guidance on this research.

Note

1. This assumes that $n^2 \gg (fm)^2$. On the other end of the scale, $k_{\min}$ approaches 0 as $(fm/n)^2$ approaches 1, although, of course, $k_{\min}$ can never be less than one, since we must take an integral number of trials. $K_{\min}$ is still $O(n^2/m^2)$ in this case, since we must have $m = O(n)$ for recognition to succeed.

References

Alter, T.D. 1994. 3-D pose from 3 points using weak-perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):802–808.

Alter, T.D. and Grimson, W.E.L. 1993. Fast and robust 3d recognition by alignment. In *Proceedings of the International Conference on Computer Vision*, pp. 113–120.

Alter, T.D. and Jacobs, D.W. 1994. Error propagation in full 3d-from-2d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 892–898.

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

Besl, P.J. and Jain, R.C. 1985. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145.

Breuel, T.M. 1992. Fast recognition using adaptive subdivisions of transformation space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 445–451.

Cass, T.A. 1988. A robust implementation of 2d model-based recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 879–884.

Cass, T.A. 1990. Feature matching for object localization in the presence of uncertainty. In *Proceedings of the International Conference on Computer Vision*, pp. 360–364.

Cass, T.A. 1992. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. In *Proceedings of the European Conference on Computer Vision*, pp. 834–842.

Cass, T.A. 1993. Polynomial-Time Geometric Matching for Object Recognition. Ph.D. thesis, Massachusetts Institute of Technology.

Chin, R.T. and Dyer, C.R. 1986. Model-based recognition in robot vision. *ACM Computer Surveys*, 18(1):67–108.

Day, W.H.E. and Edelsbrunner, H. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1(1):7–24.

Defays, D. 1977. An efficient algorithm for a complete link method. *Computer Journal*, 20:364–366.

DeMenthon, D. and Davis, L.S. 1992. Exact and approximate solutions of the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1100–1105.

Dhome, M. and Kasvand, T. 1987. Polyhedra recognition by hypothesis accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):429–438.

Edelsbrunner, H. 1987. *Algorithms in Combinatorial Geometry*. Springer-Verlag.

Feller, W. 1968. *An Introduction to Probability Theory and Its Applications*. Wiley.

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

Förstner, W. 1993. Image matching. *Computer and Robot Vision*, R. Haralick and L. Shapiro (Eds.), Addison-Wesley, Vol. II, Chapter 16.

Förstner, W. and Gülch, E. 1987. A fast operator for detection and precise locations of distinct points, corners, and centres of circular features. In *Proceedings of the Intercommission Conference on Fast Processing of Photogrammetric Data*, pp. 281–305.

Grimson, W.E.L. 1990. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press.

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

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

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

Hough, P.V.C. 1962. Method and means for recognizing complex patterns. U.S. Patent 3069654.

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

Huttenlocher, D.P. and Cass, T.A. 1992. Measuring the quality of hypotheses in model-based recognition. In *Proceedings of the European Conference on Computer Vision*, pp. 773–775.

Illingworth, J. and Kittler, J. 1988. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116.

Jacobs, D.W. 1991. Optimal matching of planar models in 3d scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 269–274.

Linnainmaa, S., Harwood, D., and Davis, L.S. 1988. Pose determination of a three-dimensional object using triangle pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5): 634–647.

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

Olson, C.F. 1995. On the speed and accuracy of object recognition when using imperfect grouping. In *Proceedings of the International Symposium on Computer Vision*, pp. 449–454.

Sibson, R. 1973. SLINK: An optimally efficient algorithm for the single link cluster method. *Computer Journal*, 16:30–34.

Silberberg, T.M., Davis, L., and Harwood, D. 1984. An iterative Hough procedure for three-dimensional object recognition. *Pattern Recognition*, 17(6):621–629.

Silberberg, T.M., Harwood, D.A., and Davis, L.S. 1986. Object recognition using oriented model points. *Computer Vision, Graphics, and Image Processing*, 35:47–71.

Stockman, G. 1987. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361–387.

Stockman, G., Kopstein, S., and Benett, S. 1982. Matching images to models for registration and object detection via clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(3):229–241.

Thompson, D.W. and Mundy, J.L. 1987. Three-dimensional model matching from an unconstrained viewpoint. In *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 208–220.

Turney, J.L., Mudge, T.N., and Volz, R.A. 1985. Recognizing partially occluded parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):410–421.