Time and Space Efficient Pose Clustering

Clark F. Olson
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720
clarko@robotics.berkeley.edu

Technical Report UCB//CSD-93-755[1]

Abstract

Pose clustering is a method of object recognition that determines the transformations that align hypothesized matches of groups of image and model features: an object that appears in an image corresponds to a large cluster of transformations in pose space close to the correct pose of the object. If there are m model features and n image features, then there are $O(m^3n^3)$ transformations to consider for the case of recognition of three-dimensional objects from feature points in two-dimensional images. I show that pose clustering can be equally accurate when examining only $O(mn)$ transformations, due to correlation between the transformations, 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 $O(mn)$ transformations, for a total complexity of $O(mn^3)$. Besides reducing the time necessary to perform pose clustering, this method requires much less memory and makes the use of accurate clustering algorithms less costly. Further time reductions can be gained by using grouping to determine the initial matches.

[1]This research has been supported in part by an NSF graduate fellowship to the author, NSF PYI grant IRI-8957274 to Jitendra Malik and NSF Materials Handling grant IRI-9114446.

1 Introduction

In any effort to build a general-purpose vision system capable of recognizing an unrestricted range of objects, model-based recognition techniques are important, since techniques that are not model-based face limitations in their capability to discriminate between objects [Moses and Ullman, 1992]. Typically, model-based object recognition techniques compare sets of features extracted from an image to model features in a database of object models. This paper describes a model-based recognition technique that improves upon previous pose clustering methods. This work is widely applicable in object recognition tasks, including the task of recognizing three-dimensional objects from single two-dimensional images.

Pose clustering is a method to recognize objects from the hypothesized matching of feature groups [Ballard, 1981, Stockman, 1987, Thompson and Mundy, 1987, Linnainmaa *et al.*, 1988, Grimson *et al.*, 1992]. In this method, the transformation parameters that align groups of model features with groups of image features are determined. Under a rigid-body assumption, the correct transformation corresponds to a large cluster of these transformations near the true pose of the object. Thus, pose clustering can be used to determine objects that may be present in the image by finding large clusters of transformations in pose space. Figures 1 and 2 show how pose clustering works in a pictorial fashion. This clustering is usually performed by histograming the transformation parameters in a quantized transformation space and looking for peaks in the histogram. Unfortunately, Grimson *et al.* [1990, 1992] have shown that this method will find a significant number of false positives for complex images with substantial noise and/or occlusion. Thus, pose clustering should be used to determine hypotheses for further verification, not as the sole means of detection.

It is well known that three matches between model points and image points is the smallest number of matches that yield a finite number of transformations that bring three-dimensional model points into alignment with two-dimensional image points [Fischler and Bolles, 1981, Huttenlocher and Ullman, 1990, DeMenthon and Davis, 1992, Alter, 1992]. Thus, if m is the number of model features and n is the number of image features then there are $O(m^3n^3)$ transformations to consider. I demonstrate that if we are given two correct matches, performing pose clustering on only the $O(mn)$ transformations that can be determined using these correct matches yields equivalent accuracy to clustering all $O(m^3n^3)$ transformations, due to correlations between the transformations. Since we do not know two correct matches in advance, we must examine $O(n^2)$ such initial matches to ensure a low probability of missing a correct object, yielding $O(mn^3)$ total time. Additional speedup can be achieved by using clustering or indexing to generate the initial matches.

Previous pose clustering methods 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 can now examine subsets of only size $O(mn)$ at a time, we require much less storage to perform clustering using this method. Due to the time complex-

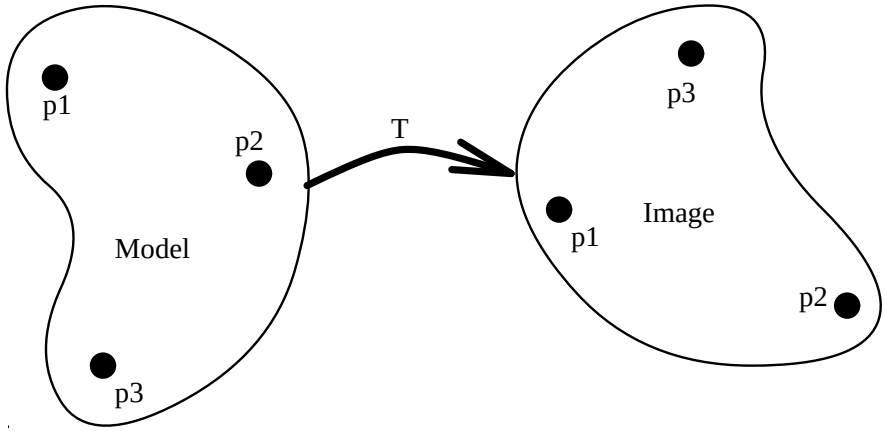

Figure 1: There exists a transformation that aligns any three non-collinear model points with any three image points.

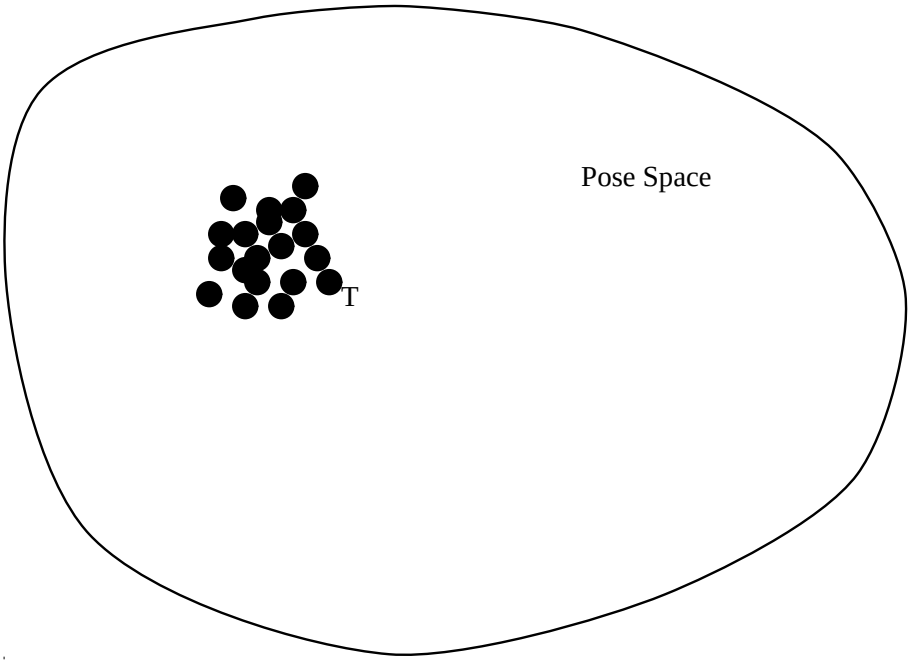

Figure 2: Correct matches between model and image point sets yield transformations that cluster close to the correct transformation.

ity of other clustering methods, most pose clustering methods have used binning 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 $O(m^3n^3)$.

The remainder of the paper will be structured as follows. First, I'll discuss prior work on pose clustering. In Section 3, I will discuss the time and space necessary to perform pose clustering. I also prove that examining small subsets of the possible transformations is adequate to determine if a cluster exists with equivalent accuracy and discuss the implications of this result on pose clustering algorithms. Section 4 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 5 discusses the time complexity necessary to implement these techniques and compares it to other algorithms for this problem. Section 6 describes the implementation of these ideas to recognize three-dimensional objects from single two-dimensional images. Experiments that have been performed to demonstrate the utility of the system are presented in Section 7. Then, Section 8 will discuss the use of hierarchical clustering techniques rather than binning to perform pose clustering. Section 9 discusses some interesting issues and Section 10 concludes the paper.

2 Prior Work

Ballard [1981] showed how the Hough transform [Hough, 1962, Duda and Hart, 1975] can be generalized to detect arbitrary two-dimensional shapes undergoing translation. First, a mapping between image space and pose space is constructed. Then, a table is created quantizing pose space. Cells of this table that are consistent with each edge pixel are then incremented. Peaks in the table correspond to possible instances of the object in the image. This system can be generalized to rotations and scaling in the plane, but since individual pixels are examined independently, a two-dimensional surface in the quantized pose space must then be incremented for each edge pixel.

Stockman *et al.* [1982] describe a pose clustering system for two-dimensional objects undergoing similarity transformations. This system examines pairs of image and model features to reduce the subset of the four-dimensional pose space consistent with a hypothesis 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.

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 (α_1, α_2) at one of the feature points corresponding to the direction of edges terminating at the point. They quantize the two-dimensional space of the possible image angles (α_1, α_2) and for each model vertex-pair, they precompute some of the transformation parameters for each

of the quantized angles. At run-time, the 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, which are assumed to correspond to correct transformations. 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 of determining object pose 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 positives in pose clustering algorithms for many cases of feature types and object recognition problems. Thus, pose clustering should be used a means of detecting possible poses for further verification, not as the sole means of object recognition. In addition, they show that conventional binning methods of clustering must examine a very large number of hash 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 positives. A positive sign for pose clustering from the work of Grimson *et al.* is that the alignment method [Huttenlocher and Ullman, 1990] produces false positives with a higher frequency than pose clustering 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 determined from sample points with maximum value. This method may miss a pose that brings many matches into alignment, but it ensures that the transformations found for any single sample point are mutually compatible.

Another related technique is to decompose pose space into regions that bring the same set of model and image features into agreement up to error bounds [Cass, 1991]. 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 harder since the transformations do not form a vector space. But, by embedding the six-dimensional affine pose space in an eight-dimensional space, it can be seen that there are $O(k^8m^8n^8)$ equivalence classes. Not all of these equivalence classes must be examined to determine the regions producings the largest matches. For example, Cass describes a method of finding the maximal match sets for two-dimensional objects undergoing similarity transformations with expected time $O(n^2m^3)$ using square uncertainty regions.

Jacobs [1991] describes a method for recognizing two-dimensional objects that shares some conceptual similarities to ours, in that once a set of initials matches are known (three for Jacobs' algorithm) a small amount of time is necessary to determine the maximum set of matches that can be brought into alignment. Rather than using pose clustering, Jacobs discretizes the six-dimensional space of possible errors in the locations of the three image feature points and determines. Each bin in this discretization represents small areas of locations where the three image features could lie when projected by the correct transformation. Thus, there is a set of additional matches which can be brought into alignment while constraining the image features to lie in the space represented by the bin. The bin where the most matches are brought into alignment is considered optimal.

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 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.

3 Recognizing Objects by Clustering Poses

For the remainder of this paper, I will focus on the recognition of three-dimensional objects undergoing unrestricted three-dimensional rotation and translation from single two-dimensional images. To simply matters, I will assume that the only features of use in the model and image are points, but the results here can be generalized to other types of features. It has been shown that matching three model points to three image points enables us to solve for the transformation that bring the points into alignment under the perspective projection and several approximations to it (including weak-perspective) [Fischler and Bolles, 1981, Huttenlocher and Ullman, 1990,

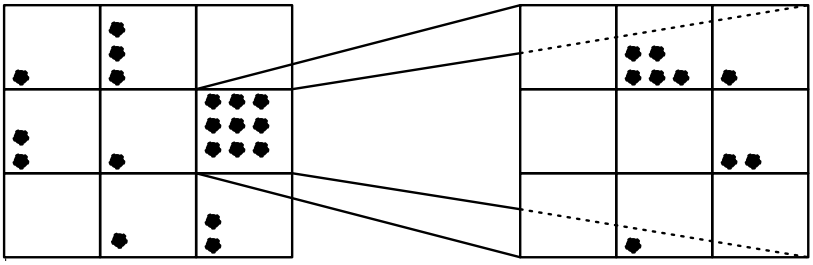

Figure 3: In coarse-to-fine clustering, coarse bins that contain many transformations are examined at a finer scale.

DeMenthon and Davis, 1992, Alter, 1992]. Note that these techniques yield at most a finite number of transformations (two or four) that bring the points into alignment. So, a pose clustering algorithm using points can use matches of size three 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 two or four transformations, depending on the model of projection used. In the ideal case, we would determine clusters by determining the exact subset of transformations space that brings each model group into alignment with each image group up to some error bound and determine points in transformation space where large numbers of these subsets intersect. Of course, due to the time limitations on current algorithms, we don't determine these exact subspaces. Most pose clustering algorithms find clusters less accurately by binning, with each group match represented by a single point in pose space. Binning is carried out by discretizing the pose space and incrementing counters for the cells that are compatible with each group match examined. Since pose space is six-dimensional for three-dimensional rotation and translation, the discretized pose space is enormous 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 binning can be performed sequentially [Thompson and Mundy, 1987, Linnainmaa *et al.*, 1988]. In coarse-to-fine clustering (Figure 3,) pose space is quantized in a coarse manner and the large clusters found in this quantization are then clustered in a more finely quantized pose space. Pose space can also be

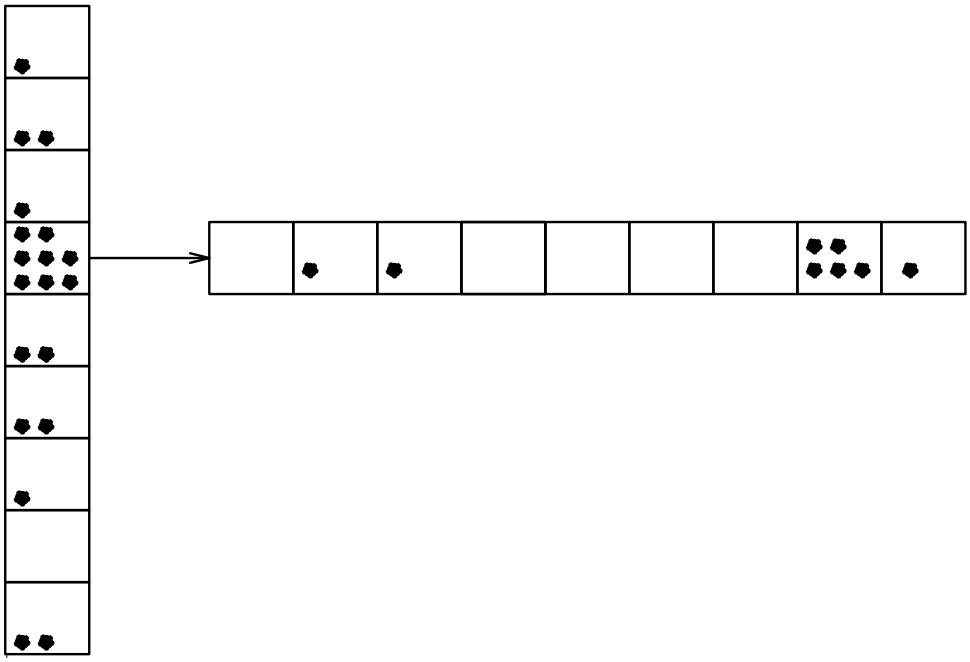

Figure 4: Pose space can be decomposed into orthogonal spaces. Clustering is then be performed in one of the decomposed spaces. Bins that contain many transformations are examined using the other decomposed spaces.

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 (Figure 4.) The clusters found in a projection of the pose space are then 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 decomposed table. In addition, we must either store with each bin the group matches that contributed to a cluster there, so that we can perform the subsequent binning steps on them, or we must reexamine all of the group matches (and redetermine the transformations aligning them) for each subsequent binning step. The first possibility requires an enormous amount of storage for previous methods and the second requires considerable extra time.

Linnainmaa *et al.* [1988] use the heuristic that only sets of three feature points that are connected by two consecutive edges are used to generate image and model groups. This reduces the number of transformations that must be clustered, but this

method has two disadvantages. First, connected edges between feature points are not easy to find due to image noise. Second, clustering performance can suffer, since a smaller number of usable image groups will be found for an object, even when they can be extracted reliably.

Clustering methods other than binning have been largely avoided due to their considerable time requirement. Clustering 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 $p = O(m^3n^3)$ transformations to cluster in previous methods this means the overall time for clustering would be $O(m^6n^6)$.

We will see that all of these problems can be solved by examining only $O(mn)$ groups at a time, but first I will show that this can be done without a loss of accuracy.

Let Θ be the space of legal poses. Each $p \in \Theta$ can be considered a function $p : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ that takes a model point to its correspondings image point. Each group match $\gamma = \{(\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3)\}$ determines some subset of 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 assume that the feature points are localized with error bounded by a circle of radius ϵ. We can define $\theta(\gamma)$ as follows:

Definition :

$$\theta(\gamma) \equiv \{p \in \Theta : ||p(\mu_i) - \nu_i||_2 \leq \epsilon, \text{ for } 1 \leq i \leq 3\}$$

Lemma 1:

There exist x distinct model points $\mu_1, ..., \mu_x$ and x distinct image points $\nu_1, ..., \nu_x$ such that for some $p \in \Theta$, $||p(\mu_i) - \nu_i||_2 \leq \epsilon$ for $1 \leq i \leq x$ **if and only if** there exist two point matches π_1, π_2 and $x - 2$ group matches that contain these point matches $\gamma_i = \{\pi_1, \pi_2, \pi_{i+2}\}$: $1 \leq i \leq x - 2$ such that $p \in \theta(\gamma_i)$ for $1 \leq i \leq x - 2$.

Proof :

If $\gamma_1, ..., \gamma_{x-2}$ exist as above, then the x point matches $\pi_1, ..., \pi_x$ that comprise these group matches must have $||p(\mu_i) - \nu_i||_2 \leq \epsilon$ by the definition of $\theta(\gamma)$.

If we have x point matches $\pi_1 = (\mu_1, \nu_1), ..., \pi_x = (\mu_x, \nu_x)$ comprised of distinct image and model points such that for some $p \in \Theta$, $||p(\mu_i) - \nu_i||_2 \leq \epsilon$ for $1 \leq i \leq x$ then choose basis matches π_1 and π_2. Each of the group matches $\gamma_i = \{\pi_1, \pi_2, \pi_{i+2}\}$: $1 \leq i \leq x - 2$ must have $p \in \theta(\gamma_i)$ by the definition of $\theta(\gamma)$. □

So, the problem of finding x distinct point matches that can be aligned to within error bounds by a single transformation has been reduced to the problem of finding $x - 2$ group matches, each containing the same basis of two point matches, such that there is a point aligning each of the group matches to within error bounds. Using this the following theorem can be proven:

Theorem 1:

There exist $g = \binom{x}{3}$ distinct group matches $\gamma_1, ..., \gamma_g$ and $p \in \Theta$ such that $p \in \theta(\gamma_i)$ for $1 \leq i \leq g$ **if and only if** there exist two point matches π_1 and π_2 and $x-2$ group matches $\gamma_{g+i} = \{\pi_1, \pi_2, \pi_{i+2}\}$: $1 \leq i \leq x-2$ such that $p \in \theta(\gamma_{g+i})$ for $1 \leq i \leq x-2$.

Proof :

If $\gamma_{g+1}, ..., \gamma_{g+x-2}$ and p exist as described above then there exist x distinct model points $\{\mu_1, ..., \mu_x\}$ and x distinct image points $\{\nu_1, ..., \nu_x\}$ such that for some $p \in \Theta$, $||p(\mu_i) - \nu_i||_2 \leq \epsilon$ for $1 \leq i \leq x$ by Lemma 1. Any group match γ composed of any three of these point matches has $p \in \theta(\gamma)$ by the definition of $\theta(\gamma)$. Since we can choose $\binom{x}{3}$ distinct group matches in this manner, there are $\binom{x}{3}$ distinct group matches γ such that $p \in \theta(\gamma)$.

If $g = \binom{x}{3}$ distinct group matches $\gamma_1, ..., \gamma_g$ and $p \in \Theta$ exist such that $p \in \theta(\gamma_i)$ for $1 \leq i \leq g$ then there must be at least x point matches $(\mu_1, \nu_1), ..., (\mu_x, \nu_x)$ such that $||p(\mu_i) - \nu_i||_2 \leq \epsilon$ for $1 \leq i \leq x$, since otherwise we could not choose g distinct group matches fulfilling $p \in \theta(\gamma_i)$. Since we have these x point matches, Lemma 1 says we must have $x-2$ group matches as described in the theorem.

This implies that clustering is equally as accurate when we examine these small subsets of the group matches, if we assume that we have an idealized clustering system that is able to determine exactly the clusters of matches that can be aligned up to the error bounds. Of course, we do not have such a system, but experiments show that an approximation can perform well (see Section 7.)

So, instead of finding a cluster of size $\binom{x}{3}$ out of 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 the same basis of two point matches. For a single basis, there are $(n-2)(m-2) = O(mn)$ group matches such that no feature is used more than once. Of course, examining just one image basis will not be sufficient to rule out the appearance of an object in an image. We could simply examine all $2\binom{n}{2}\binom{m}{2}$ possible pairs of basis matches, but we will see in Section 5 that using randomization we can examine $O(n^2)$ pairs of matches and achieve as much accuracy as desired.

4 Frequency of False Positives

The analysis above deals only with the size of clusters expected for correct objects. It is also necessary to analyze the size of clusters likely to be formed by random combinations of matches so that the likelihood of a false positive occurring can be determined. If f is the fraction of model features that appear in the image, then the size of the correct clusters has been cut by a factor of

$$\frac{\binom{fm}{3}}{fm-2} \approx \frac{(fm)^2}{6}$$

since the clusters were originally of size $\binom{fm}{3}$ and now the clusters are of size $fm-2$. We have cut the number of transformations that may potentially contribute to a false positive by a factor of

$$\frac{\binom{n}{3}}{n-2} \approx \frac{n^2}{6}$$

The potential size of the false negative has been cut more than the expected size of a correct cluster since $n > fm$. This provides hope that the accuracy of clustering is also improved by these techniques. Unfortunately, if the Bose-Einstein occupancy model accurately describes the distribution of the incorrect transformations (as experiments by Grimson and Huttenlocher [1991] indicate), the accuracy is slightly worse than indicated by previous work [Grimson *et al.*, 1992], if both analyses assume an idealized pose clustering system.

Grimson *et al.* analyze the pose clustering approach to examine 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 using the Bose-Einstein occupancy model. This analysis assumes independence in the locations of the transformations, which we have seen is not completely accurate. When we examine group matches that vary in only one point match, then the transformations have only two degrees of freedom. If we assume that the locations of spurious image points are independent, then the transformations are independent on these degrees of freedom. I shall modify the analysis of Grimson *et al.* to account for the correlation between transformations to obtain a more accurate result. First, I'll summarize their results. The Bose-Einstein occupancy model yields the following approximation of the probability that a bin will receive l or votes due to random accumulation:

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

In this equation, λ is the expected number of votes in a single bin (including redundancy due to uncertainty in the image.) For their analysis $\lambda = 6b\binom{m}{3}\binom{n}{3} \approx \frac{bn^3m^3}{6}$, where b is the average fraction of bins each group match votes for (called the redundancy factor,) n is the number of image features, and m is the number of model features.

Grimson *et al.* have determined overestimates of the size of the redundancy factor b necessary to ensure that the correct bin is among those voted for by an image group for various noise levels using a bounded error model. In the bounded error model, the location of each image feature is known to be within some distance ϵ of the measured location.

Using this redundancy factor, Grimson *et al.* determine the maximum number of image features that can be tolerated without surpassing a given error rate δ. 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 get:

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

As noted above, this analysis can be made more accurate by considering the correlations between the transformations. Theorem 1 and Lemma 1 together imply that there exist $\binom{fm}{3}$ group matches and some point p in transformation space that brings each of the model points in each of the 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. To do this we first determine the fraction of transformation space that brings a single point match into alignment (which I'll call b_p.) Note that for any rotation and scaling, there is set of translations of area $\pi\epsilon^2$ that brings the point into alignment up to the error bounds. So, if we assume that all transformations computed take each model point to somewhere within the image boundaries it can be seen that the redundancy is $\frac{\pi\epsilon^2}{d_1 d_2}$, where d_1 and d_2 are the height and width of the image. Note that this is the same as the translational redundancy in [Grimson *et al.*, 1992]. This assumption is not completely correct, but since each transformation aligns three model points with three points within the image boundaries, most of the other model points should be transformed to lie within the image boundaries. In any case, this assumption provides us with an overestimate of the necessary redundancy.

If we otherwise follow the analysis of Grimson *et al.* we get:

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

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

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

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

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

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

Note that this is not very different from the result derived by Grimson *et al.* since $b_p = \sqrt[3]{b}$ if the redundancy necessary for each match can be determined exactly. 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.* for the same system.

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 must recognize this as a possible instance of the object. Two possible solutions to this problem present themselves. First, we could use more descriptive features. The results presented here, are easily generalized to encompass features conveying more information, such as line segments or oriented points. Another solution is to examine subregions of complex images. Since most complex images will not have objects that span their entire range, regions of these images can be examined separately to determine objects present in each region. This solutions runs the risk of an large object being split among several regions, but segmentation algorithms can be used to help prevent this, and most algorithms can recognize objects despite missing some features.

The primary implication of the above analysis on the techniques presented here is that unless we are limited to simple images or use more descriptive features than points to determine the transformations, we must still use pose clustering as a method of find likely hypotheses for further verification, not as the sole means of recognition.

5 Complexity

This section discusses the time complexity necessary to perform pose clustering using techniques that exploit the analysis described above. We can use a randomization technique proposed by Fischler and Bolles [1981] and also used by Lamdan *et al.* [1988] and Cass [1991] to limit the number of pairs of matches that must be examined. A random pair of image points is chosen to examine as the image basis points. All basis matches using these image points are examined and if one of them leads to recognition of the object then we may stop. Otherwise, we continue choosing image basis points at random until we have reached a sufficient probability of recognizing the object if it is present in the image. If we randomly pick image basis points in this manner and we require fm model points to be present in the image to ensure recognition, the probability of not choosing a correct image basis in k tries is:

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

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

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

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

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

(To a first-order approximation: $k_{\min} = \frac{n^2}{(fm)^2} \ln \frac{1}{\delta}$)

For each image basis, we must examine each of the $2\binom{m}{2} = O(m^2)$ permutations of model points which may match them. So, in total we must examine $O(\frac{n^2}{m^2}) \cdot O(m^2) = O(n^2)$ basis matches to achieve accuracy $1 - \delta$, assuming that picking a correct image basis results in identification of the object. Since we examine $O(mn)$ group matches for each basis, our method requires $O(mn^3)$ time per object in the database, where previously $O(m^3n^3)$ was required. A comparison against other object recognition algorithms for this problem may be useful.

Lowe does not analyze the complexity of the SCERPO system [Lowe, 1987], but there are $O(m^3n^3)$ possible initial matches that may be used in his system. The system iteratively adds additional matches by examining possible matching image features for each unmatched image feature. Since this step is $O(m \log n)$ and this step may be repeated $O(m)$ times if the model is present in the image, the entire system appears to be $O(m^5n^3 \log n)$, although Lowe uses grouping to considerably reduce the number of initial matches examined. The use of a randomization technique similar to the one described here (and elsewhere) would reduce this to $O(m^2n^3 \log n)$.

The alignment method [Huttenlocher and Ullman, 1990] examines each of the $O(m^3n^3)$ matches between three model points and three image points and determines the transformation that aligns each of them. An $O(m \log n)$ verification step is performed for each match for a total of $O(m^4n^3 \log n)$. If the randomization technique is used to limit the number of group matches examined, the running time of the alignment method can be reduced to $O(mn^3 \log n)$. Huttenlocher and Ullman use virtual points from directional information at the feature points to reduce the complexity when this information is available to $O(m^3n^2 \log n)$ without randomization, and it can be further reduced to $O(mn^2 \log n)$ with randomization. When this directional information is available we can use it to generate virtual points for our algorithm as well, reducing the complexity to $O(mn^2)$.

Since Thompson and Mundy [1987] examine pairs of image features, they have only $O(m^2n^2)$ initial matches that must be examined. Their system assumes that directional information from edges can be reliably determined at each of the feature points. Our system can by easily modified to use the same features as used by Thompson and Mundy, when they are available. This would reduce the complexity

of our algorithm to $O(mn^2)$ in the case where directional information at the feature points is reliably determined.

The system of Linnainmaa *et al.* [1988] would be $O(m^3n^3)$ if all possible feature groups were examined. The use of the heuristic that only the feature points connected by consecutive edges are considered as groups reduces the complexity, but this method will suffer from problems if edges can not be reliably extracted. Even when edges are reliably extracted, objects may have a small number of such groups present in an image, making them extremely difficult to recognize.

6 Implementation

It would be too time consuming to determine exactly which bins in pose space intersect the set of poses that bring each match into alignment up to the error bounds at the resolution necessary to limit false positives. Determining overestimates of the subset of transformation space that bring each group into alignment as done by Grimson *et al.* [1992] requires considerable extra time and they find that their overestimates are approximately 1000 times larger than the true redundancy. A transformation sampling [Cass, 1988] approach could be used, but this method requires explicitly examining each possible match for each sampled point to determine if it is brought into alignment.

My system follows previous pose clustering algorithms in that transformations are represented by a single point in pose space. Overlapping bins that are large enough to contain most, if not all, of the transformations consistent with the bounded error are used. My implementation uses overlapping bins such that $3^6 = 729$ bins may contain a single transformation. This method should be able to find almost all of the correct transformations without having as much redundancy as the method of Grimson *et al.*, but it does not have optimal accuracy. False positives may be a problem for this system for complex or noisy images. For such images, we may want to find clusters using a slower method, such as sampling transformation space and determining which matches are brought into alignment by each sampled point (similar to Cass [1988].) This alternative will find no cases where the transformations in a cluster are not mutually consistent, at the risk of missing a cluster due to the sampling rate.

I will now describe a simple, space-efficient, and fast method of performing binning in the complete pose space. The method of Huttenlocher and Ullman [1990] is used to determine the transformation parameters that bring the three model points into alignment with three image points in the weak-perspective imaging model. Varying image noise levels are accounted for in this implementation by varying the size of the bins used in the binning procedure.

Since binning is used to find clusters, either coarse-to-fine clustering or decomposition of the pose space is required, since the six-dimensional pose space is quite large. Pose space can be decomposed into the six orthogonal spaces corresponding to each of the transformation parameters. To solve the clustering problem, binning can

be performed recursively using a single transformational parameter at a time. In the first step, all of the transformations are binned in a one-dimensional array, using just the first parameter. Each bin that contains more than $fm-2$ transformations[1] is retained for further examination, where f is now some predetermined fraction of model features that must be present in the image for us to recognize the object. Each of these bins is recursively clustered by binning the transformations held in each of them separately, 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 transformational parameters and thus form a cluster in the complete pose space.

This method can be viewed as a variant of depth-first tree search. The root of the tree corresponds to the entire pose space, each node corresponds to some volume of the pose space, and the leaves correspond to the individual bins in the discretized, six-dimensional pose space. Each level of the tree corresponds to examining the transformations in the bins corresponding to the nodes at the previous level of the tree using a previously unexamined transformation parameter. Thus, the tree has height six. 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 5 gives an outline of this algorithm. If each of the parameters has been examined on some branch of the tree (i.e. we are at a leaf,) then we output a cluster if the leaf corresponds to volume of pose space containing at least $fm-2$ transformations. Otherwise, we bin the transformations using a previously unexamined transformation parameter. Each of the bins that contains at least $fm-2$ transformations is then clustered recursively using the remaining transformation parameters. Nodes corresponding to bins containing less than $fm-2$ transformations are pruned.

Although, this decomposition of the binning problem has not previously been formulated as a tree search, Grimson and Huttenlocher's analysis [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 needed to be clustered.

The following analysis may give some intuition into why this is the case. If r is the maximum number of bins that can overlap at a single point (and thus contain a single transformation,) then there could be $O(\frac{rn^3}{f^3})$ bins that hold as many as $\binom{fm}{3}$ transformations at each level of the tree for previous pose clustering methods, since there are $O(m^3n^3)$ transformations. Using the techniques presented here, we can have no more than $\frac{rn}{f}$ bins that contain as many as fm transformations, since there are less than nm transformations clustered at a time.

This means that there can be at most $\frac{rn}{f}$ bins that are not pruned at each level of the tree, for this system, containing no more than rmn total transformations. Thus, $O(n)$ bins are examined and $O(mn)$ total time is required. In addition, $O(mn)$ space

[1]If fm model points are present in the image, a single basis will yield $fm-2$ correct transformations

```
function find-clusters(input: transformation-set,parameter-set)
    if (cardinality(parameter-set)==0) then
        if (cardinality(transformation-set)> fm − 2) then
            output-cluster(transformation-set);
    else
        choose some p ∈parameter-set;
        bin-array=bin-transformations(transformation-set,p);
        for b = 1 to array-length(bin-array)
            if (cardinality(bin-array[b])> fm − 2) then
                cluster(bin-array[b],parameter-set−p);
end
```

Figure 5: Recursive binning algorithm.

is required to execute this algorithm.

Once clusters are found we use a method similar to that of Huttenlocher and Cass [1992] to determine an estimate of the number of consistent matches. They argue that the 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 number of distinct model points or image points matched in the cluster, since it can be determined quickly (as a opposed to the maximal bipartite matching) and is reasonably accurate.

7 Results

I have performed several experiments to verify the usefulness of techniques described above. Experiments have been performed on two systems, the first uses the conventional pose clustering technique of clustering all of the possible transformation simultaneously, but otherwise was implemented as described above. The second implements all of the ideas discussed in this paper.

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

1. Model points were determined randomly from a cube with side length 200 units, such that no two model points were with 10 units of each other.

2. The model was transformed by a random rotation and translation and was projected using the perspective projection onto the image plane.

3. Bounded noise ($\epsilon = 1.0$) was added to each image point.

	System 1			System 2		
m	optimal	average	%	optimal	average	%
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	7400.78	.749	38	30.79	.810
50	19600	14569.93	.743	48	40.47	.843

Table 1: Performance finding correct clusters. m: number of object points; optimal: size of optimal cluster: $\binom{p}{3}$ for System 1 and $p-2$ for System 2; average: the average size of cluster found; %: the average fraction of optimal cluster found.

4. If any two image points were within 4 pixels of each other, steps 2 and 3 were repeated.

5. In some experiments, additional random image points were added such that no two image points were within 4 pixels of each other.

Table 1 shows the performance of these two systems at finding correct clusters. The first system finds much larger clusters, of course, since it clusters many more correct transformations, but the size of the false clusters is expected to be rise at approximately the same rate (see the analysis in Sections 3 and 4.) The new techniques actually find a larger percentage of the optimal size clusters. This is because these clusters have smaller extent; when using a basis set of two matches, the noise associated with those two image points stays constant over the entire cluster. This noise moves the cluster from the true location, but does not increase the size of the cluster, as it does when we do not use a basis set. For System 1, the noise inherent in varying all three of the matches comes into play, increasing the size of the clusters. This demonstrates an advantage the new pose clustering techniques have over previous methods that is not reflected in the analysis of previous sections.

Table 2 shows the size of incorrect clusters found by the second system for models of 20 random model points for various image complexities. Shown are the average size of the largest cluster found for each image basis, the standard deviation among these, and the size of the largest cluster over all of the image bases. Since the system found clusters of average size 15.02 for models of twenty points that appear in the image, these levels of complexity will not cause a large number of false positives to be found.

Table 3 shows the results of experiments determining the number of trials necessary to recognize objects in the presence of random extraneous image points. A cluster is required to be 80% of the optimal size (14 for models of size 20.) Even though clusters of suboptimal size are allowed, these experiments assume $f = 1.0$

n	average	std. dev.	maximum
20	3.79	0.81	6
40	5.32	1.20	10
60	6.35	1.49	12
80	7.23	1.66	12
100	7.91	1.86	13
120	8.22	2.02	14
140	8.51	2.14	14
160	8.68	2.19	15

Table 2: Size of false negative clusters found by System 2 for objects with 20 feature points. n: number of image points; average: average size of largest cluster for each image basis; std. dev.: the standard deviation among the size of the largest cluster for each image basis; maximum: the largest cluster found for any image basis.

since each model point appears in each image. We cannot assume that each correct basis will result the algorithm finding a clustering cluster of even 80% of the optimal size. If we estimate that in pathological models and/or images, only 50% of the correct bases will result in a sufficiently large cluster, then we have:

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

For each value of n, Table 3 shows k_{limit} for $\delta = 0.01$, the average number of trials necessary to recognize the object, the maximum number necessary, and the number of objects (out of 100) that required more than k_{limit} trials. For each case, at least 98 of the 100 objects were recognized within k_{limit} trials. Overall, 99.3 percent of the objects were recognized within k_{limit} trials, with the expectation of recognizing $1 - \delta = 99.0$ percent of the objects.

To summarize the experimental results, the new pose clustering method has been determined to find a larger fraction of the optimal cluster than previous methods and result in very few false negatives for non-complex images. In addition, the number of basis matches we must examined to recognize objects has been confirmed experimentally to be $O(n^2)$, making the total time required $O(mn^3)$.

8 Hierarchical Clustering

Due to the number of transformations that were clustered in previous pose clustering methods, binning had an enormous time advantage over other clustering techniques. For example, many hierarchical agglomerative clustering techniques have

n	k_{limit}	average	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

Table 3: Number of trials to find objects with 20 points. n: image points; k_{limit}: number of trials the analysis says is necessary for $\delta = 1.0$; average: the average number of trials necessary to recognize the object; max: the maximum number of trials necessary to recognize the object; over: number of objects (out of 100) that required more than k_{limit} trials to recognize.

been proposed that operate in $O(p^2)$ time (where p is the number of points to cluster.) These algorithms use the pairwise distances between points and find nearest neighbors to determine which points to agglomerate (e.g. [Sibson, 1973, Defays, 1977, Day and Edelsbrunner, 1984].) Some results are summarized by Murtagh [1983].

The basic structure of these methods is as follows: At the beginning each point is considered a cluster by itself. A matrix of intercluster distances using some distance metric (upon which some restrictions may be placed depending upon the algorithm used) is determined. Some pair of mutual nearest neighbors are then combined into a single cluster and the distances to the new cluster are updated. This process is repeated until all of the points belong to a single hierarchical cluster. Such algorithms require $O(p^2)$ time when the updating can be performed efficiently.

The result of hierarchical clustering algorithms is a structure called a dendrogram, which is a representation that records which clusters were linked at each step (see Figure 6.) This single hierarchical cluster can be easily decomposed into clusters with less than some maximum radius by recursively subdividing clusters that are too large into the two clusters that were linked to form each of them.

Since previous pose clustering algorithms required the clustering of $O(m^3n^3)$ points, the use of these clustering methods would have required far too much time. Now that the pose clustering problem has been reduced to subproblems of size $O(mn)$, these clustering methods are less costly. In particular, since clustering is performed $O(n^2)$ times and clustering using hierarchical clustering techniques takes $O(m^2n^2)$ time, the overall time bound becomes $O(m^2n^4)$ when hierarchical clustering is used

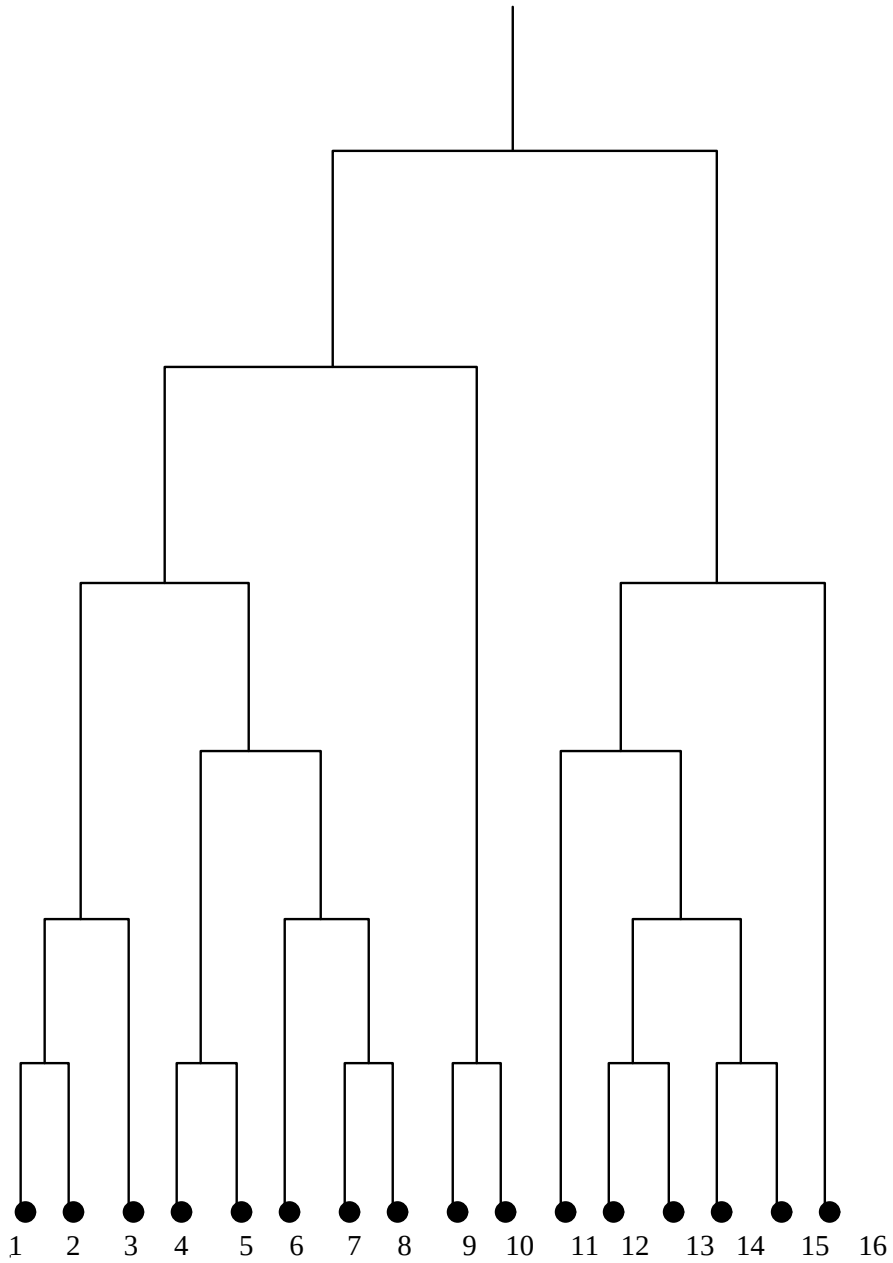

Figure 6: A dendrogram shows how the clusters are merged hierarchically.

instead of binning. This may be acceptable for small problem sizes or on parallel networks, where p points can be clustered $O(p \log p)$ time [Li and Fang, 1989, Olson and Ranade, 1993]. Clustering p points using hierarchical clustering can be performed using $O(p)$ space for many algorithms. Since $O(mn)$ transformations are clustered at a time, this means $O(mn)$ space is required to cluster the transformations.

Grimson and Huttenlocher [1990] note that partitional clustering could be used (i.e. the k-means clustering method,) but do not consider hierarchical clustering. The k-means method is not acceptable for pose clustering for two reasons. First, the number of clusters into which the points are partitioned must be known in advance. Second, each point is placed in one of the partitions. Since most of the transformations clustered in pose clustering are spurious, forcing them into a cluster will not yield good results, in general.

9 Discussion

The techniques described in this paper can be used with recognition strategies other the pose clustering, if these strategies examine pose space to determine where the transformations aligning several groups of points lie. For example, Breuel [1992] recursively subdivides pose space to find volumes that intersect the most consistent 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 found to be linear in the number of constraint regions. To recognize three-dimensional objects from two-dimensional 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 (in Section 3) implies that Breuel's algorithm will still find the best match if it examines only the $O(mn)$ constraint regions associated with a given basis of two correct matches of feature points. Since we don't know two correct matches in advance, we must examine $O(n^2)$ of them (using the randomization technique from Section 5,) yielding a total time of $O(mn^3)$, as with our pose clustering algorithm.

These results can be generalized to the case of features other than points. For example, Breuel examines features that are line segments (with unoccluded end points.) For the case of two-dimensional objects undergoing similarity transformations, only one match of a model line segment to an image line segment is necessary to determine a bounded constraint region. Since any possible basis match will contain all of the information necessary to determine the transformation, the techniques presented in this paper will not be useful. For the case of three-dimensional objects, more than one match will be necessary. So, for this case we can realize a significant speedup using these techniques.

10 Conclusion

I have shown that pose clustering for the case of three-dimensional object recognition from two-dimensional objects does not require the clustering of $O(m^3n^3)$ transformations. Pose clustering with the same accuracy can be achieved by clustering $O(mn)$ transformations, if two correct point matches are given to us. In the case where we do not know two correct point matches, $O(n^2)$ initial point matches must be examined to achieve a negligible probability of a false negative, for a total time requirement of $O(mn^3)$. Since far fewer transformations are clustered at a time, this method requires much less memory and accurate clustering methods become more feasible with its use.

Acknowledgements

The author thanks Jitendra Malik for his guidance on aspects of this research. This research has been supported by a National Science Foundation Graduate Fellowship to the author, NSF Presidential Young Investigator Grant IRI-8957274 to Jitendra Malik, and NSF Materials Handling Grant IRI-9114446.

References

[Alter, 1992] T. D. Alter. 3D pose from 3 corresponding points under weak-perspective projection. Massachusetts Institute of Technology, A.I. Memo No. 1378, July 1992.

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

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

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

[Cass, 1991] T. A. Cass. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. Manuscript, October 1991.

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

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

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

[Duda and Hart, 1975] R. O. Duda and P. E. Hart. Use of the Hough transform to detect linea and curves in pictures. *Communications of the ACM*, 15:11–15, 1975.

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

[Fischler and Bolles, 1981] 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.

[Grimson and Huttenlocher, 1990] 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.

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

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

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

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

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

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

[Lamdan *et al.*, 1988] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–344, 1988.

[Li and Fang, 1989] X. Li and Z. Fang. Parallel clustering algorithms. *Parallel Computing*, 11:275–290, 1989.

[Linnainmaa *et al.*, 1988] S. Linnainmaa, D. Harwood, and L. S. Davis. Pose determination of a three-dimensional object using triangle pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5), September 1988.

[Lowe, 1987] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[Moses and Ullman, 1992] Y. Moses and S. Ullman. Limitations of non model-based recognition schemes. In *Proceedings of the European Conference on Computer Vision*, pages 820–828, 1992.

[Murtagh, 1983] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, 26:354–359, 1983.

[Olson and Ranade, 1993] C. F. Olson and A. Ranade. Parallel algorithms for hierarchical clustering. Technical report, University of California at Berkeley, 1993.

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

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

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

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