Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 251-258, 1994.

Time and Space Efficient Pose Clustering

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

Abstract

This paper shows that the pose clustering method of object recognition can be decomposed into small subproblems without loss of accuracy. Randomization can then be used to limit the number of subproblems that need to be examined to achieve accurate recognition. These techniques are used to decrease the computational complexity of pose clustering. The clustering step is formulated as an efficient tree search of the pose space. This method requires little memory since not many poses are clustered at a time. Analysis shows that pose clustering is not inherently more sensitive to noise than other methods of generating hypotheses. Finally, experiments on real and synthetic data are presented.

1 Introduction

Model-based object recognition systems determine which objects appear in images using a catalog of object models and estimate their positions and orientations (poses) relative to the camera. This paper examines methods of improving the efficiency of the pose clustering method of object recognition. This is done by decomposing the pose clustering problem into many small subproblems, which can be examined sequentially or in parallel. Only a small fraction of these subproblems need to be examined in order to achieve accurate object recognition. In addition, I present a method of clustering the poses that guarantees time and space efficiency. The analysis presented here shows that pose clustering is not inherently more sensitive to noise than other methods of object recognition.

This paper focuses on the recognition of three-dimensional objects undergoing unrestricted rigid transformations from monocular intensity images. To simplify matters, the only features used for recognition are feature points in the model and the image, but the results here can be generalized to any features from which the pose can be determined. For this problem, the computational complexity of pose clustering is reduced from $O(m^3n^3)$ to $O(mn^3)$ and the space required is shown to be $O(mn)$, where m is the number of model points and n is the number of image points.

2 Pose Clustering

Pose clustering is a technique that is used to recognize objects in images from hypothesized matches between feature groups [10, 12, 13]. The transformation parameters that align these hypothesized matches are determined. Under a rigid-body assumption, all of the correct hypotheses will yield a transformation 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 have typically examined the poses from all possible matches.

To prevent a combinatorial explosion in the number of poses considered, we want to use as few as possible matches between image and model points to determine the pose of the object. It has been shown that matching three model points to three image points is sufficient to constrain the pose to a finite set of points under the perspective projection and the weak-perspective approximation [5, 9]. A pose clustering algorithm can thus use matches between three model points and 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 these group matches yields two or four possible transformations. A cluster among these poses in the pose space indicates that there is a pose that very nearly brings several group matches into alignment. (The pose takes each of the model points in the group matches close to its corresponding image point.) We should therefore recognize this as a possible instance of the object.

2.1 Clustering Techniques

Ideally, we would find exactly those points in pose space that would bring a large number of model points into alignment with image points up to some error boundary. Work in this direction has been undertaken by Cass [2, 3], but these methods can be time consuming and are difficult for the case of three-dimensional objects.

Most pose clustering algorithms perform clustering less accurately by histograming the poses. In this method, each pose is represented by a single point in pose space, rather than the subset of poses that bring the point matches in the group match into alignment up to the error bounds. The pose space is discretized into overlapping bins and the poses are histogramed in these bins to find large clusters. 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 [12] and decomposing the pose space into orthogonal subspaces in which histograming can be performed sequentially [10, 13]. In coarse-to-fine clustering, 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. Alternately, 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. The clusters found in the 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 [6] 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 histogram.

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 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 an enormous amount of storage 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 while still achieving high accuracy. Similar techniques in the context of transformation equivalence analysis can be found in [4].

3 Decomposition of the Problem

In this section, I show how the pose clustering problem can be decomposed into much smaller subproblems. Each of these subproblems examines only those those group matches which contain a basis of two point matches. If the point matches are correct then equivalent accuracy to the original problem is achieved.

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 corresponding 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. I will consider a generalization of this function $\theta(\gamma)$ that applies to sets of point matches of any size.

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 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 examining the subproblems has equivalent accuracy to examining the original pose clustering problem.

Theorem 1:
The following statements are equivalent:

1. There exist $g = \binom{x}{3}$ distinct group matches that pose $p \in \Theta$ brings into alignment up to the error bounds. Formally,
$$\exists \gamma_1, ..., \gamma_g \text{ s.t. } p \in \theta(\gamma_i) \text{ for } 1 \leq i \leq g$$
2. There exist x distinct point matches $\pi_1, ..., \pi_x$ that pose $p \in \Theta$ brings into alignment up to the error bounds:
$$\exists \pi_1, ..., \pi_x \text{ s.t. } p \in \theta(\{\pi_i\}) \text{ for } 1 \leq i \leq x$$
3. There exist $x - 2$ distinct group matches sharing some pair of point matches that pose $p \in \Theta$ brings into alignment up to the error bounds:
$$\exists \pi_1, ..., \pi_x \text{ s.t. } p \in \theta(\{\pi_1, \pi_2, \pi_i\}) \text{ for } 3 \leq i \leq x$$

Proof :
The proof of this theorem has three steps. I 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 clearly 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 accuracy equivalent to the examining all of the group matches when we examine subproblems in which only those group matches that share some basis of two 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 the same basis of two point matches. Furthermore, it is clear that any two correct point matches can be used as this basis. For a single basis, there are $(m-2)(n-2) = O(mn)$ 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 the next section that we can examine $O(n^2)$ pairs of matches and achieve as much accuracy as desired.

```
Function recognize(input: model-points, image-points)
  Repeat:
    Choose two random image points $\nu_1$ and $\nu_2$.
    For all pairs of model points $\mu_1$ and $\mu_2$:
      For all point matches $(\mu_3, \nu_3)$:
        Determine the poses aligning the group
        match $\gamma = \{(\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3)\}$.
      Find and output clusters among these poses.
End
```

Figure 1: New pose clustering algorithm.

Figure 1 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 proposed by Fischler and Bolles [5] 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 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 image basis in k tries, where each trial consists of examining a random pair of point matches. Since the probability of a single image point being a correct model point is at least $\frac{fm}{n}$, the probability of a basis being incorrect is at most $1-(\frac{fm}{n})^2$. Thus, the probability that k random trials will all be unsuccessful is:

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

If we solve for the minimum number of trials to achieve accuracy δ we get:

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

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

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 $1-\delta$. 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. The time bound varies with the logarithm of the desired accuracy, so very high accuracies can be achieved without greatly increasing the running time of the algorithm.

5 Frequency of False Positives

While the above analysis has been interpreted in terms of the correct clusters so far, it also applies to incorrect 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 systems.

Grimson *et al.* [7] analyze the frequency of this occurring for a pose clustering system that examines all of the hypotheses simultaneously. This analysis assumes that the locations of the poses in pose space from each of the group matches are independent, but this is not quite 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 $\binom{6}{3} = 20$ group matches that can be formed using these six point matches is also brought into alignment by this pose. The poses determined from these group matches are therefore highly correlated. Theorem 1 shows 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. A similar analysis accounting for this source of correlation [11] shows that the expected frequency of false positives is actually slightly worse than previously thought.

It should be noted that this result is a fundamental limitation of all object recognition systems that use only point features to recognize objects. Similar results (although less restrictive) exist for other features. Any time there exists a pose that brings a large number of model features into alignment with corresponding image features, a system dealing with only such features should recognize this as a possible instance of the object.

The primary implication of this analysis on the techniques presented here is that, unless we are limited to simple images or use more descriptive features than points, we must use pose clustering as a method of finding likely hypotheses for further verification, not as the sole means of recognition. As an additional verification step, we could, for example, verify the presence of edge information in the image as done by Huttenlocher and Ullman [9].

6 Efficient Clustering

This section discusses how clustering of the poses can be performed efficiently with respect to both time and space. This analysis will assume that we are considering a single object model. Multiple objects are handled sequentially by this system.

6.1 Recursive Histograming

This system uses histograming to achieve fast clustering. Each transformation is 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. This prevents clusters from being missed due to their falling on a boundary between bins. This method is able to find most of the correct transformations, but it does not have optimal accuracy. More accurate techniques (e.g. [3]) may be used at the cost of lower speed.

My implementation uses the method of Huttenlocher and Ullman [9] to determine the transformation parameters that bring the three model points into alignment with three image points in the weak-perspective imaging model. Varying levels of image noise are accounted for by varying the size of the bins used in the clustering procedure.

Since histograming is used to find clusters, either coarse-to-fine techniques or decomposition of the pose space is required, since the six-dimensional pose space is immense. I use the decomposition approach. 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 transformational 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 some predetermined fraction of model features that must be present in the image for us to recognize the object. The transformations in each of these bins are then clustered 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.

6.2 Formulation as Tree Search

This method can be formulated as a depth-first tree search. The root of the tree corresponds to the entire pose space, each node corresponds to a volume of the pose space, and the leaves correspond to individual bins in the 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 2 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 the 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 bin (after all of the parameters have been examined) that contains enough poses, we output the location of the cluster.

6.3 Efficiency of Clustering

Although this decomposition of the binning problem has not previously been formulated as a tree search, Grimson and Huttenlocher's analysis [6] 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.

```
Function find-clusters(  input:
                         $\mathcal{P}$ - set of poses,
                         $\Pi$ - set of pose parameters)
  If $|\Pi| > 0$ then
    Choose some $\pi \in \Pi$.
    Histogram poses in $\mathcal{P}$ by parameter $\pi$.
    For each bin $b$ in the histogram:
      If $|b| > fm - 2$ then
        Find-clusters($\{p \in \mathcal{P} : p \in b\}$,$\Pi \setminus \pi$);
  Else
    Output the cluster location.
End
```

Figure 2: Recursive clustering algorithm. (See text.)

Since previous systems would histogram $O(m^3n^3)$ 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 binning in a high-dimensional space, we may have a large number of bins at even low levels of the tree, since we are clustering so many transformations. Using the techniques presented here, we 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 can be only $O(n)$ unpruned bins at each level and these bins contain $O(mn)$ total transformations. Thus, we do not have saturation near the root of the tree for this system. $O(mn)$ time and space is required per clustering step, since we must cluster $(m - 2)(n - 2)$ poses.

Once a cluster is found, I use a method described by Huttenlocher and Cass [8] to determine an estimate of the number of consistent matches in each cluster. 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 lesser of the number of distinct model points and the number of distinct image points matched in the cluster, since they can be determined quickly (as opposed to the maximal bipartite matching) and is reasonably accu-

	Old method			New method		
m	opt	avg	pct	opt	avg	pct
10	120	96	.80	8	6.6	.83
20	1140	882	.77	18	15.0	.83
30	4060	3047	.75	28	23.2	.83
40	9880	7401	.75	38	30.8	.81
50	19600	14570	.74	48	40.5	.84

Table 1: Performance finding correct clusters. m =the number of object points; opt=the size of optimal cluster; avg=the average size of clusters found; pct=the average fraction of optimal cluster found.

rate. This estimation requires time linear in the number of matches in the cluster.

7 Results

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

7.1 Synthetic Data

For the experiments on synthetic data, models were generated from random points inside a cube, rotated and translated randomly, and projected onto the image using the perspective projection. Bounded noise ($\epsilon = 1.0$) was added to each image point.

The first experiment determined whether the correct clusters were found. Table 1 shows the performance of two systems 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 techniques actually find a larger percentage of the optimal size clusters. This is because these clusters are smaller. When using a basis set of two matches, the noise associated with those two image points stays constant over the entire cluster. This noise may move 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.

Experiments were run to determine the size of false hypotheses generated by the new system for models of 20 random model points and various image complexities. Table 2 shows the average size of the largest incorrect cluster found for each image basis and the size of the largest cluster over all of the image bases. Since the system found correct clusters of average size 15.0 for models of twenty points and the average size of the largest false positive cluster for each incorrect basis is 8.68 for 160 random image points, these levels of complexity do not appear to cause a large number of false positives to be found.

n	avg	max
20	3.79	6
40	5.32	10
60	6.35	12
80	7.23	12
100	7.91	13
120	8.22	14
140	8.51	14
160	8.68	15

Table 2: Size of false positive clusters found for objects with 20 feature points. n=the number of image points; avg=the average size of the largest cluster for each image basis; max=the largest cluster found for any image basis.

To summarize these results, the new pose clustering method finds a larger fraction of the optimal cluster than previous methods and results in few false negatives for images of moderate complexity.

7.2 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 makes no use of the fact that the figures are planar. No benefit is gained from using this data set except that corners are easy to detect. Furthermore, the only features used in either data set to generate hypotheses are the locations of corner points in the image.

In these tests, I do not assume that all of the correct hypotheses will yield poses that fall in the same bin. Small modifications are thus made to the equations determining the number of trials that must examined and the number of poses that a bin must contain not to be pruned. These modifications do not change the computational complexity of the system.

Figure 3 shows an example of recognizing objects from the first data set in an image. The top image

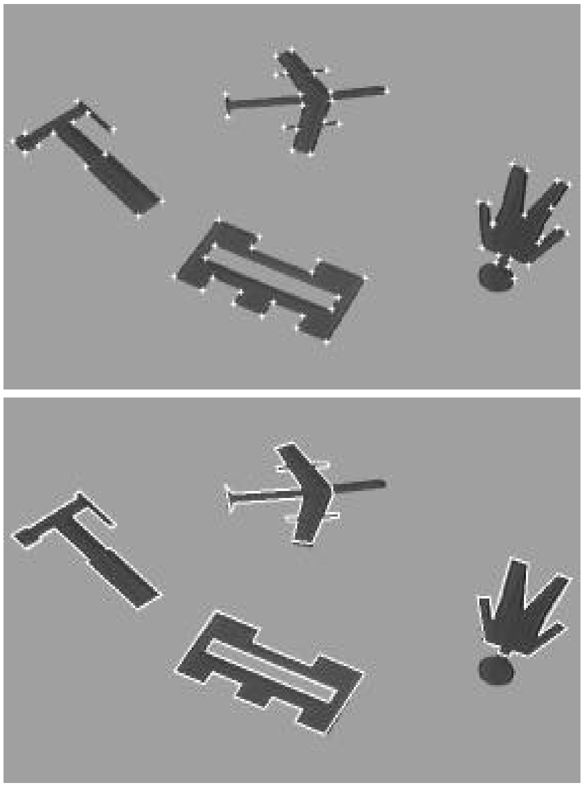

Figure 3: Recognition example for 2D objects. Top: The corners found in an image. Bottom: The four best hypotheses found, with edges drawn in. (The nose of the plane and head of the person do not appear because they were not in the models.)

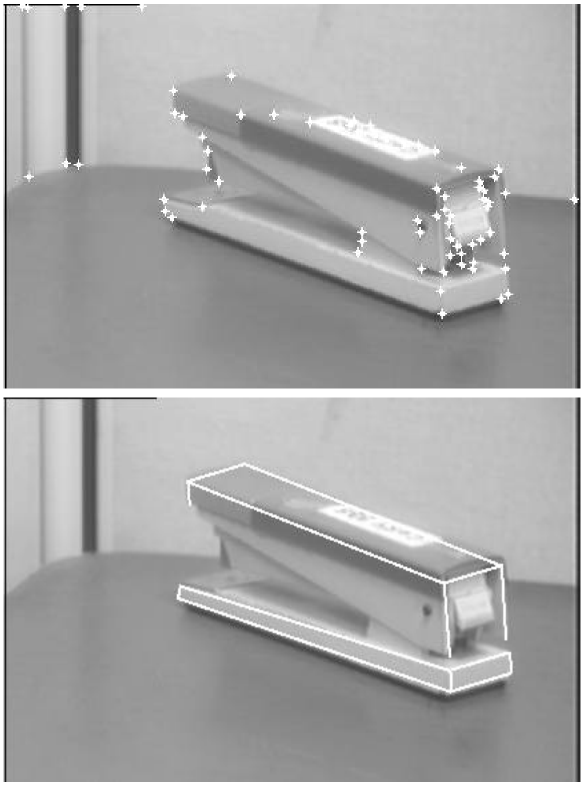

Figure 4: Recognition example for a 3D object. Top: The features found in the image. Bottom: The best hypothesis found.

shows the 84 feature points found by a corner detector. The bottom image shows the best hypotheses found for this image. Figure 4 shows an example of recognizing a stapler from the second data set. The top image shows the 70 feature points used to recognize the stapler. The bottom image shows the best hypothesis found.

The largest source of error in many of the experiments on real images is the use of weak-perspective as the imaging model. It appears that the assumption that this model is adequate for most problems may prove incorrect as the accuracy of algorithms improves.

8 Discussion

The decomposition techniques described in this paper can be used with recognition strategies other the pose clustering. For example, Breuel [1] 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 $O(m^3 n^3)$ 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 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,) yielding a total time of $O(mn^3)$. Of course, this introduces a probability δ that a correct basis will not be chosen, and thus recognition may fail where it would not in the original system.

A point worth discussing is that some previous researchers in pose clustering have claimed that finding a large enough peak in the pose space is sufficient to consider the object present in the image. Others have claimed that pose clustering is more sensitive to noise and clutter than other algorithms. Grimson *et al.* [6, 7] have shown that we should not simply assume large clusters are instances of the object; additional verification is needed to ensure against false negatives. While it is clear that further verification is required for hypotheses generated by pose clustering, other methods, such as alignment, also require this additional verification step. The analysis of Section 5 shows that pose clustering is not inherently more sensitive to noise and clutter than other algorithms.

9 Conclusion

I have shown that pose clustering for the case of three-dimensional object recognition from two-dimensional objects from point features 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 known. 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 few transformations are clustered at a time, this method requires little memory. These techniques can be easily generalized to recognize objects from any features from which the pose can be determined.

Analysis has shown that a fundamental bound exists on the accuracy that can be achieved by algorithms that recognize objects by finding sets of features that can be brought into alignment. Within the limitations of the bound, pose clustering performs well.

Acknowledgments

The author thanks Jitendra Malik for his guidance on this research. Thanks are also due to Paul Debevec, Ruth Rosenholtz, Aaron Wallack, and Joe Weber for proofreading this work. 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.

References

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

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

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

[4] T. A. Cass. *Polynomial-Time Geometric Matching for Object Recognition.* PhD thesis, Massachusetts Institute of Technology, February 1993.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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):634–647, September 1988.

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

[12] 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.

[13] 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*, volume 1, pages 208–220, 1987.