Fast Object Recognition by Selectively Examining Hypotheses

by

Clark Francis Olson

B. S. (University of Washington) 1989
M. S. (University of Washington) 1990

A dissertation submitted in partial satisfaction of the
requirements of the degree of
Doctor of Philosophy
in
Computer Science
in the
GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Jitendra Malik, Chair
Professor John F. Canny
Professor Stephen E. Palmer

1994

The dissertation of Clark Francis Olson is approved:

Chair Date

Date

Date

University of California at Berkeley

1994

Fast Object Recognition by Selectively Examining Hypotheses

©Copyright 1994

by

Clark Francis Olson

Abstract

Fast Object Recognition by Selectively Examining Hypotheses

by

Clark Francis Olson

Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Jitendra Malik, Chair

Several systems have been proposed to recognize three-dimensional objects in (two-dimensional) intensity images by computer. A problem that has plagued most object recognition systems for this problem is the low rate at which images are processed unless the problem is constrained, due to the vast number of hypothetical matches between sets of image features and sets of model features. Hypothetical poses can be determined from a small number of model features appearing in the image. The number of correct matches between these small sets of model features and image features (and thus correct hypotheses) is combinatorial in the number of model features appearing in the image. Since, ideally, only one of these correct hypotheses needs to be found to recognize the object, an exhaustive examination of all hypothetical matches is not necessary. I describe techniques to obtain fast object recognition through the selective examination of the possible hypotheses.

First, I describe how the pose clustering method of object recognition can be decomposed into subproblems of much smaller size. In addition, I show that only a small fraction of these subproblems need to be examined to recognize objects with a negligible probability of introducing a false negative. This allows us to reduce the computational complexity of the algorithm, as well as reducing the amount of space necessary. I show how the clusters of poses that indicate a good hypothesis can be found quickly in a space efficient manner. A noise analysis and experiments on real images indicate that this system has good performance.

Next, I describe a probabilistic indexing system to determine which of the initial hypothesized matches between three model points and three image points are most likely to be correct. This system takes advantage of the probabilistic peaking effect,

which implies that if all viewing directions are equally likely, the distribution of angles and ratios of distances in the image will have a sharp peak at the model value. This effect can be used to select hypotheses to examine that are more likely to be correct than others. The probabilistic indexing system is used with noise criteria to obtain a speedup of two orders of magnitude in the alignment method. It is expected that these techniques will also result in a significant speedup when applied to pose clustering.

The implementation of these ideas in a connectionist framework is discussed. While alignment and pose clustering methods can be implemented in this framework, the best approach for this case is to use election methods. Such methods allow much of the computation to be performed off-line, thus simplifying the processing elements required. Election methods use indexing to generate hypothesized matches between groups of points. Voting is then performed to determine which objects have the most support in the image. My analysis shows that model-based object recognition can be performed extremely quickly given a large number of simple processing elements.

These techniques vastly improve the speed at which model-based object recognition algorithms can be performed.

Contents

List of Figures

List of Tables

Acknowledgments

I'd like to thank my adviser Jitendra Malik, whose advice and suggestions were very helpful over the four years I've been at Berkeley. I am also grateful to the other members of my thesis committee, John Canny and Stephen Palmer. I'd like to thank Joe Weber and Ruth Rosenholtz for their friendship and answering my various questions. Finally, I'd like to thank my wife Rebecca for her loving support.

Portions of this thesis have appeared or will appear in technical reports, conferences, and journal articles (e.g. [Olson, 1992,1993a,1993b,1993c,1994a,1994b].)

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.

Chapter 1

Introduction

1.1 Object recognition is a difficult problem

The recognition of three-dimensional objects in images by computer has been an active area of research since the seminal work of Roberts [1963]. Roberts provided an early method of determining an object's position and orientation in an image (hereafter called *pose*) from a set of matches between object points and image points and determined criteria under which such a pose would be considered correct. This formed a primitive version of a now-popular object recognition method called the alignment method. He considered that not each set of points in the image is equally likely to result in a correct pose and determined good sets for use through the use of connectivity in edge maps in the image, setting the stage for grouping techniques in later work. In fact, Roberts considered many problems that are still active topics of research, such as pose estimation, feature matching, and grouping, as mentioned above, as well as, occlusion of objects, edge detection and the problem of incomplete edge maps, and hierarchical object recognition.

Despite considerable research since that time, we have only scratched the surface of potential in object recognition, as demonstrated by comparing performance to what can be considered the most powerful of computers, the human brain. Electronic computers possess a small fraction of human capability in this problem. Since people have been studying this problem for over 30 years, you might ask why we have not advanced further. The answer is that object recognition is a extremely complicated task. It should be noted that the human brain is an extremely powerful computational device with between 10^{10} and 10^{11} neurons. Approximately 30% of the cerebral cortex is devoted to vision. Thus, humans expend a huge amount of computational resources in visual processes.

Consider the image in Figure 1.1. While humans are immediately able to recognize the tree in the foreground, the grassy ground area, and the forest in the background, consider the difficulty of determining that the object in the foreground is a tree. Generic trees can take a wide variety of shapes, since we may see the trunk, branches, leaves or any combination of these. Additionally, many different objects have a cylindrical shape similar to what we determine is the shape of the trunks of the tree in the image. Furthermore, we determine this shape from only the two-dimensional image data, with no prior information as to the boundaries between objects, the illumination conditions in the image, or the nature of the scene. From this two-dimensional array of brightness values we are able to determine the approximate shape and depth of most of the points in the scene, and are able to reason about the general class of various objects in this scene, and if we were previously familiar with them, we could often determine the location of the scene in the image, the type of tree, or even the specific instance of a tree. Clearly, this process requires a large amount of computation and reasoning.

In particular, the following problems make object recognition (and other machine vision problems) very difficult:

- Projection: The recognition of three-dimensional objects from two-dimensional intensity images is complicated by the loss of information in the imaging process, in which the three-dimensional structure of the object is projected onto the two-dimensional image. This loss of information results in ambiguity in the position of the objects in the image. In other words, we don't know the distance from the camera to the imaged points. Each image could have been the result of an infinite number of scenes varying in the distances of the various points from the camera.

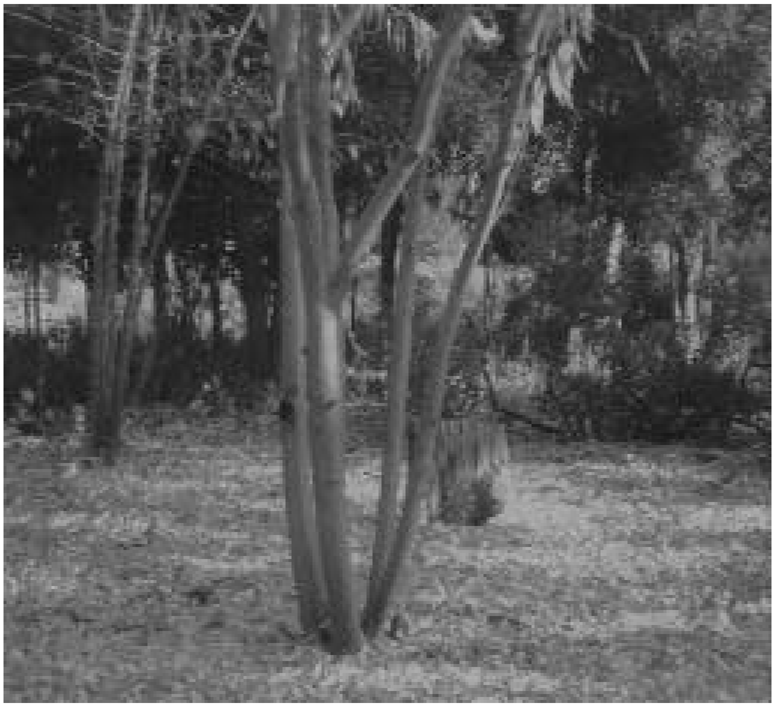

Figure 1.1: An image demonstrating the complexity of machine vision problems. (This image is from the SRI Sequence from the IEEE Motion Workshop Database at Sarnoff Research Centre, courtesy of NASA-Ames Research Center and SRI International.)

- Unrestricted viewpoint: When we capture an object in an image, the viewpoint from which the object is seen is arbitrary. In general, there are six degrees of freedom in the position of the camera with respect to an object: translation in the x-, y-, and z-directions, and rotation about each axis. Thus, an object may appear in an infinite number of different positions and orientations, giving rise to widely varying appearances in images.

- Interaction of processes: The imaging process consists of the interaction of several complex processes. The brightness of any point in a scene depends on several factors including the position, orientation, and intensity of the illumination, the color and texture of the object, how the reflectance of the object changes with the viewing direction (e.g. specular vs. lambertian surfaces,) as well as atmospheric considerations such as fog, and camera considerations that cause blurring.

- Noise: The imaging process is not perfect. The image is discretized into cells represented by pixel values. The average brightness intensity of the cell is the desired pixel value, but even if we obtain this average, we are condensing information and losing the precise location of various image phenomena. Furthermore, it is difficult to sense brightness intensities accurately and such intensities are also discretized resulting in further inaccuracy. Cameras often introduce image distortion through imperfections in the lens, and some electronic imaging devices are known to produce high-frequency noise in images.

- Information content: Images provide a wealth of specific information. A typical size for an image is 512×512 pixels and each pixel value usually has a range of 0 to 255. Thus, there are $256^{512^2} > 10^{630,000}$ possible images that could be produced in this case. This wealth of information has to be transformed into an understandable format at an abstract level. Condensing this volume of very specific data into a few meaningful pieces of information is difficult.

- Occlusion: Our ability to recognize an object in an image is limited if we are not able see all of the object. Not only will self-occlusion be present in opaque objects, since we will not be able to see the back of the object, but other objects may occlude some portion of the object we wish to recognize that would otherwise be visible.

- Spurious features: Of course, not all of the features (such as edges or corners) we will find in a typical image will result from the object we wish to recognize. Other objects in the image (that we may also want to recognize) will also produce features that we must examine and texture or undesirable image attributes such as glare or noise can and will result in additional features. While such spurious features don't necessarily hide the correct object features, they make the recognition problem considerably more difficult by introducing a large number of hypothetical object positions that must be considered.

- Complex objects: Objects with subparts that are very similar to other objects or subparts of objects can be difficult to distinguish. Furthermore, object symmetries can cause a substantial amount of extra work to be performed, since two or more distinct poses of the object will result in a very similar image. Algorithms may spend considerable time examining each possibility.

To summarize, a wide variety of problems make the creation of a general object recognition system an extremely difficult task. While this area has been an active area of research for some time, the current systems are still quite limited.

1.2 Is there hope?

Given these substantial difficulties you may ask why we attempt to solve the problem at all, and indeed, if it is even solvable. To answer the second question, we have the human brain as our existence proof that solving this problem can be done, although researchers into human object perception still have little idea how this task is performed in the brain. (In contrast, some low-level human vision tasks are fairly well understood.) The answer to the first question is (at least) twofold. First, possible applications of object recognition systems are tremendous. A general robot vision system is the ultimate goal, but even the partial solutions currently available are useful in industrial tasks such as the automatic registration and inspection of machined parts. Commercial, medical, and military applications have also been explored. Second, an understanding of computational techniques for object recognition by machine may help us to understand human object recognition processes. While it not reasonable to assume that human object recognition processes function in a manner similar to current computer object recognition techniques, an understanding of the computational requirements of various techniques can give us insight into what techniques would be feasible in the brain and thus help guide human vision researchers.

Some insights have been key to the partial solutions that have been obtained to this time. Lowe [1987] argued that $2\frac{1}{2}$- or 3-dimensional information from depth reconstruction (using stereo, motion, shading, or texture) is not needed for object recognition. In fact, while such depth reconstruction is important for other tasks, human object recognition appears not to require it, since this information is often not available. While Lowe gives additional arguments, let's just consider a typical edge image (e.g. Figure 1.2.) This figure yields very little information about the three-dimensional shape of the scene, since there is no shading and little texture, and motion and stereo don't apply to single images. It is true that some shape information might be obtained through line drawing labeling [Clowes, 1971, Huffman, 1971, Mackworth, 1973, Malik, 1987] or other interpretation using line drawings [Barrow and Tenenbaum, 1981, Stevens, 1981, Koenderink, 1984, Malik and Maydan, 1989], but depth information derived strictly from line drawings is sparse and imprecise and deriving it requires noise-free line drawings that are very difficult to generate from real images. Thus this information is not very useful in determining the relative depths of the various points in the image.

Despite this lack of depth information, people have little problem recognizing the object in the Figure 1.2 as a stapler. Since this information doesn't seem to be necessary (or even useful [Biederman, 1985]) in human object recognition capability, researchers in this area are now attempting to recognize objects directly from the two-dimensional, geometrical information in images.

To alleviate the problems due to the interactions of complex processes and the

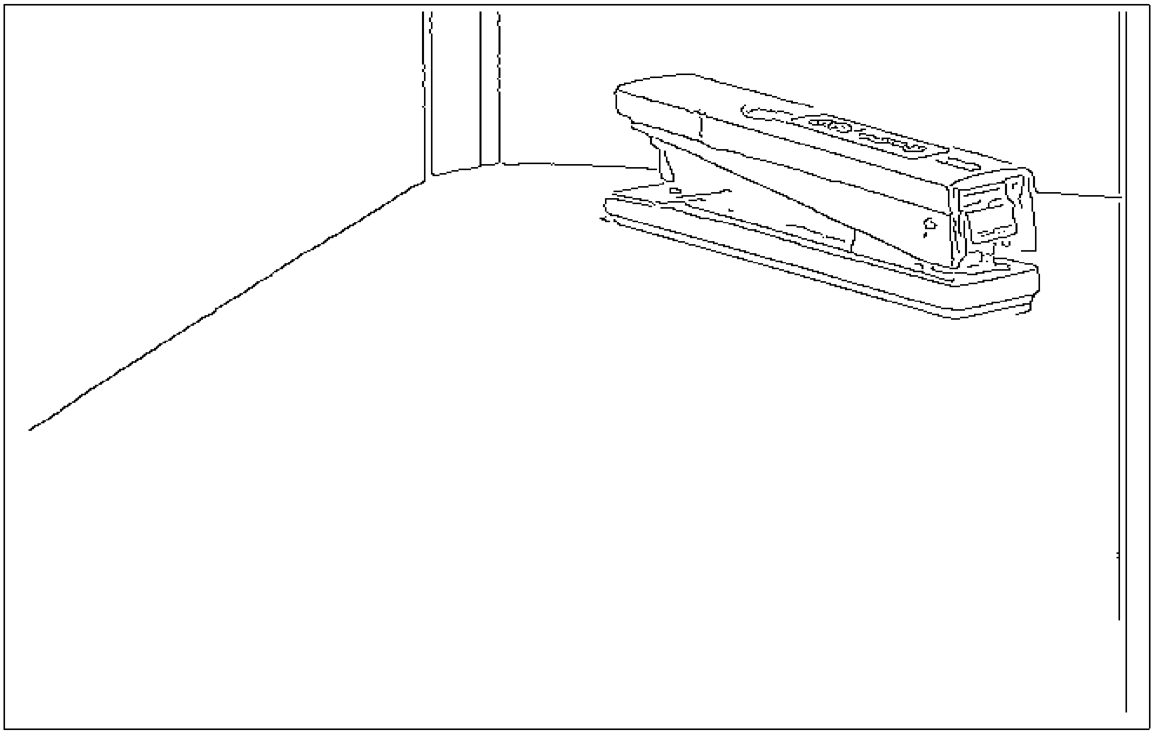

Figure 1.2: A typical edge image illustrating the point that three-dimensional data is not necessary to recognize objects.

huge amount of specific data in the imaging process, the intensity images are typically reduced to a set of features such as edges or corners (typically through filtering and thresholding) or segmented regions (from region growing or split-and-merge algorithms,) which can be used to recognize objects. To deal with occlusion, we want these features to be local, so that if a part of the object is not seen, we can still find features accurately in the portion of the object that does appear in the image. Image noise is an omnipresent problem in the localization of image features, but an understanding of the sources of noise allows us to model its effects and determine which objects may be present under this noise model.

For some types of features, attributes that are invariant to the viewing direction can be found to help recognize objects. In other cases, we might find a hypothetical pose from some small set of possibly corresponding model and image features. Model-based techniques can be used to guide the search for objects and exclude spurious image features from undue consideration. Finally, hierarchical recognition techniques can help prevent the complexity arising from similar subparts or symmetry.

Substantial gains have been made in the area of object recognition by computer, but the distance that remains to be covered before we achieve our goal is considerably further than the distance we have already covered.

1.3 Model-based object recognition

This thesis will be primarily concerned with a field of object recognition called model-based recognition. This field simplifies the problem by requiring the detection of only those objects in the image that are present in some database describing their attributes. Many model-based object recognition systems require precise description (through feature locations, etc.) of the models in the database, possibly from computer-aided design (CAD) models. Such algorithms are sometimes called CAD-based recognition algorithms.

In model-based recognition, a set of object descriptions is predetermined in some manner. This catalog of object models can then be actively used in the recognition process. For example, we could match features in our model to hypothetically matching points in the image and use such matches to determine the hypothetical pose of the object in the image. In addition, we could then transform addition features in our model according to this pose and determine if the pose transforms them such that they are aligned with corresponding image features. (This forms the basis of the alignment method.) This and similar techniques use the object models to guide the search for objects appearing in the image.

In a survey of model-based object recognition techniques, Chin and Dyer [1986] describe three central issues that must be resolved in creating a model-based recognition system. These are:

> 1. What features should be extracted from an image in order to describe physical properties and their spatial relations in a scene adequately?
>
> 2. What constitutes an adequate representation of these features and their relationships for characterizing a semantically meaningful class of objects?
>
> 3. How should the correspondence or matching be done between image features and object models in order to recognize the parts in a complex scene?

So, to paraphrase, the keys to designing a model-based recognition system are determining what features to use, how they should be combined to form the object models, and how to match the model and image features in recognizing the object in the image. In this thesis, an object model is taken to be a set of object features and geometrical relationships among them. Statistics generated from a model such as moments or invariants that do not convey geometrical relationships between features will not be considered object models, although such statistics can be useful for generating hypotheses regarding which objects are present in the image.

The use of model-based techniques simplifies the problem in several ways. First, it limits the possibilities that must be examined, since we are considering only those objects that are present in the database. If we didn't have these models to use, we would be forced to determine the shape using depth reconstruction techniques and reason about these shapes, both of which are difficult problems. Thus, the active use of this database prevents us from having to determine object shape in a bottom-up manner. The use of precise object models allows use to dispense with contextual information and concentrate on the geometrical properties of the object and image.[1]

While using specific geometric models gives us much information to use in the recognition process, the use of CAD-based models can also be somewhat limiting. Many objects are hard to model precisely (e.g. a crumpled wad of newspaper or a chocolate chip cookie.) In addition, the recognition of a previously unobserved object as being in some class of objects is not possible in this framework, since we will not have a precise description of the new object from which to recognize it.

It is important to note that some kind of model is almost necessary in object recognition, since it can be proven that techniques that do not use models face limitations in their ability to discriminate between objects [Moses and Ullman, 1992].

1.4 New techniques

While all three of the issues described by Chin and Dyer must be addressed, my work has concentrated on improving algorithms for determining correspondences between model and image features. A study of what are the best features to use and how they should be combined to form the object models has not been undertaken. My justification for this is that while these issues need to be kept in mind, they can be studied separately. The techniques that I describe, while focusing on specific features and model representations, can be generalized to virtually any set of local features that are stable with respect to viewpoint changes and any geometrically precise object representation.

The limitations inherent in the approach I take are two-fold. The techniques do not always generalize well to very complex image features and they do not all apply well if a geometrically precise object representation is not available. The first limitation is not, in fact, very limiting, since complex image features are difficult to find with precision and few complex features are typically present in models. The use of simple, local features promotes robustness and flexibility in the recognition

[1] While this simplifies the problem at one level, it also limits us. If we are to build a general system we must use this contextual information.

process. While the the techniques I describe cannot be applied directly to smoothly curved objects, it is my hope that generalizations of these techniques will be useful in recognizing such objects. The second limitation is more damaging, although much current research is on systems that demand such precise object models. As noted above such systems cannot recognize many objects since they are difficult to model precisely and classification of previous unobserved objects is not possible. While this reliance is acceptable for many industrial tasks such as the automatic registration or inspection of machine parts, it is a limitation that will need to be overcome if general robot vision systems are to be achieved.

The primary problem that my techniques address is the low speed at which object recognition techniques perform in finding three-dimensional objects in unrestricted two-dimensional (intensity) images. Many currently popular recognition systems (including the alignment and pose clustering methods, and indirectly election methods, such as geometric hashing) use matches of small sets of simple features in the image to corresponding features in the model to generate hypothetical object poses. Of course, correct matches aren't known in advance, so many matches must be examined, and it must then be determined whether each hypothetical match is correct. The number of correct small sets of image features (and thus correct hypotheses) is combinatorial in the number of model features appearing in the image. Ideally, only one of these correct hypotheses needs to be found to recognize the object, so an exhaustive examination of all hypothetical matches is not necessary. I describe techniques to obtain fast object recognition through the selective examination of the possible hypotheses.

First, I describe how the pose clustering method of object recognition can be decomposed into subproblems of much smaller size based on hypotheses using matches between two model points and two image points. I then show that only a small fraction of these subproblems must be examined to recognize objects with a negligible probability of introducing a false negative (when an object appears in the image but is not found.) In addition, I show how the clusters of poses that indicate a good hypothesis can be found quickly in a space efficient manner. This allows a reduction in the computational complexity of the algorithm, as well as reducing the amount of space necessary to perform the recognition. A noise analysis of this system and experiments on real and synthetic data show that this system has good performance.

Next, I describe a probabilistic indexing system to determine which of the initial hypothesized matches of three model points to three image points are likely to be correct. This system takes advantage of the probabilistic peaking effect, which implies that if all viewing directions are equally likely, the distribution of angle and ratios of distances in the image will have a sharp peak at the value taken by the features in the three-dimensional object. This effect can be used to select hypotheses to examine that are more likely to be correct that others. The probabilistic indexing system is used with noise criteria to obtain a speedup of two orders of magnitude in the alignment method.

The implementation of these ideas in a connectionist framework is then discussed. While alignment and pose clustering can be implemented in this framework, the best approach for this case appears to be election methods, since they allow much of the computation to be shifted off-line, thus simplifying the processing elements required. Elections methods use indexing to generate hypothesized matches between sets of features. Voting is then performed to determined which hypotheses are best. My analysis shows that model-based object recognition can be performed extremely quickly in this framework, given a large number of simple processing elements.

Finally, I conclude with a look at some of the future directions for this research and object recognition in general.

Chapter 2

Review of Previous Work

Many different methods have been used to provide (partial) solutions to the model-based object recognition problem. I certainly won't discuss all of them, due to space limitations, but I will discuss several important algorithms and analyses. I've divided these into several categories, these being: search, alignment, pose clustering and invariance. More extensive surveys of recognition techniques can be found elsewhere [Binford, 1982, Besl and Jain, 1985, Chin and Dyer, 1986].

2.1 Object transformations

Before discussing the techniques themselves, it is important to understand that these techniques do not all solve the same problem. They vary in whether they apply to two-dimensional (planar) or three-dimensional objects, as well as, the transformations that are allowed in the imaging process. For example, for planar objects, some algorithms may be limited to similarity transformations where others model full three-dimensional rotation and translation. Researchers may assume rigid transformations or affine transformations that allowing skewing. Furthermore, while cameras are generally governed by the perspective projection, many researchers use a linear approximation to this called weak-perspective or scaled orthography. Finally, some researchers recognize objects from range or tactile data that yields depth information, rather than intensity images. These variations will be discussed in this section.

2.1.1 Similarity transformations

Similarity transformations apply only to planar objects. The object is allowed to be translated in the x- and y-directions, rotated about the z-axis, and scaled, so this transformation has four degrees of freedom. This class of transformations accurately models the case where the object is always perpendicular to the viewing direction. Rotation about the x- and y-axes are not allowed. If object points are represented by two parameters m_x and m_y, and the similarity transform has translations t_x and t_y, rotation θ and scale s, then we can determine the coordinates of the transformed point i_x and i_y by:

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} m_x \\ m_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

2.1.2 Perspective transformations

Perspective transformations allow the full range of the six-dimensional transformation space (three translations and three rotations) and accurately models the imaging process. Let $R_{\theta,\phi,\psi} \in SO(3)$ be a 3×3 matrix denoting the rotations and $\begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$ be the translation. The transformed points (before projection) can be found by:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = R_{\theta,\phi,\psi} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

If f is the focal length of the camera, then applying the perspective projection yields:

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \frac{f}{p_z} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

Solving for the perspective projection given a set of matching points is difficult due to its nonlinearity. Furthermore, it requires knowing the focal length and center point of the camera in advance. Alternately the focal length can be treated as an additional variable, but this case requires more feature matches to determine the transformation and is somewhat unstable with respect to the calculated distance to the object, since this can only be recovered from perspective effects [Alter, 1992].

This transformation can also be used with planar objects. In this case, we simply have $m_z = 0$ for each model point.

2.1.3 Weak-perspective transformations

Thompson and Mundy [1987] have shown that for objects that have little depth with respect to their distance from the camera, the weak-perspective class of transformations (also called scaled orthographic) is an accurate, linear approximation to perspective transformations. It is assumed that each of the transformed model points is approximately the same distance z_0 from the camera. Like perspective transformations, weak-perspective transformations have six degrees of freedom. In this case, we get the following for the full transformation and projection:

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \frac{f}{z_0} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

where $\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \end{bmatrix}$ is the first two rows of $R_{\theta,\phi,\psi}$. The focal length f and average depth z_0 are often condensed into a single scale factor.

It is also possible to use unscaled orthographic projections, where the f/z_0 is one, but this assumes that the object always has the same size (but not orientation) in the image. In this case, the transformation only has five degrees of freedom.

2.1.4 Affine transformations

The affine class of transformations consist of all fully linear transformations. We no longer constrict the transformation to be rigid, in that we allow linear skewing in the x- and y- directions, yielding a transformation with eight degrees of freedom. An advantage to this class of transformations is that they model the transformations (neglecting perspective effects) that would take place when capturing an image of a picture of an object [Jacobs, 1992]. The transformation is similar to weak-perspective except that $\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \end{bmatrix}$ is no longer constrained to be the first two rows of a rotation matrix, thus allowing skewing and implicitly scaling the object.

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

For planar objects, the affine class of transformations is equivalent to weak-perspective, since R_{13} and R_{23} are irrelevant. The transformation has six degrees of freedom and is given by the following equation:

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} m_x \\ m_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

2.1.5 Summary

The class of transformations that is to be modeled is another parameter that must be considered in the design of a model-based object recognition system. For planar objects, similarity transformations may be sufficient if the orientation of the objects is constrained, but perspective or weak-perspective is necessary if it is not. Non-planar objects can sometimes be treated as planar objects if we view them from a finite number of viewpoints, but more general models are usually necessary.

Different models of the imaging process yield varying degrees of difficulty in solving for the transformation from matches between model features and image features. Perspective transformations are the most difficult to compute, requiring prior knowledge of the focal length and center point of the camera and then solution of a quartic equation (see [Haralick *et al.*, 1991] for a review of the solutions to this problem.) The advantage to perspective transformations is that they model the imaging process completely accurately.

Weak-perspective transformations are considerably easier to solve for [Huttenlocher and Ullman, 1990] due to the linearization of the transformation and prior knowledge of camera parameters is no longer necessary, but they neglect perspective effects in the imaging process. Quadratic constraints on the transformations (since we are constrained to rigid rotations) complicate the solution slightly.

Affine transformations are the simplest to determine. Since the transformation has no nonlinear constraints simple linear algebra techniques are applicable. Affine transformations neglect perspective effects and require more feature matches to determine since they have two additional degrees of freedom for three-dimensional objects.

2.2 Search

While several researchers have done interesting work involving a tree search of the space of possible matches (e.g. [Haralick and Shapiro, 1979, Ben-Arie and Meiri, 1987, Flynn and Jain, 1991], I will discuss only the work of Grimson and Lozano-Pérez [1984, 1987], since I concentrate on non-search methods. Their work is interesting due to the in-depth analysis which has been performed on the complexity of this system and search in general.

Grimson and Lozano-Pérez describe a constrained search method for recognizing two-dimensional objects from intensity images and three-dimensional objects from range or tactile data, generalizing previous work by Gaston and Lozano-Pérez [1984]. An interpretation tree is constructed that at each level of the tree expands each unpruned node by matching an unexamined image feature to each possibly consistent model feature. The basic idea is that local constraints on the distances and angles between image features can be used to eliminate inconsistent interpretations in a pairwise fashion. At each node of the interpretation tree, the new match is checked for consistency with each of the previous matches and if any pair of matches does not satisfy the local constraints, the node is pruned from the tree. The set of model features is augmented by the null match to account for spurious image features.

Analysis by Grimson [1990, 1991] has shown that if not all of the image data is from the model that is being recognized, the expected recognition time is generally exponential, but the use of a heuristic search termination condition can reduce the expected time required to a polynomial function of the number of features under certain noise and clutter limitations.

2.3 Alignment

The basic idea of the alignment method is to determine hypothesized poses from small sets of matches between image features and model features. These poses can then be tested to determine if they are correct. For this reason, such techniques are sometimes called generate-and-test methods. As mentioned previously, Roberts [1963] work uses a very basic form of this algorithm. Some other important results for these techniques are given here.

2.3.1 RANSAC

Fischler and Bolles [1981] describe the RANSAC (for Random Sample Consensus) system. They provide a closed-form solution to the perspective-3-point problem for pose estimation (determining the pose under the perspective transformation from matches between three model points and three image points,) and prove that no more than four positive solutions exist for this problem. They randomly choose sets of image points for use in determining hypothesized poses. Such random sets are examined until the object is recognized or there is a small probability that the object can be found. Fischler and Bolles accept a model as correct if some hypothesis transforms some percentage of the model points close to corresponding image points.

2.3.2 HYPER

Ayache and Faugeras [1986] describe the HYPER system (for Hypotheses Predicted and Evaluated Recursively) for the recognition of partially occluded planar objects undergoing similarity transformations. Hypotheses are generated by matching a privileged model segment (the longest model segments are privileged) to an image segment. This provides an initial estimate of the model transformation. Additional segments are added to the matching in order of their proximity to the initial segment, while updating the transformation at each step. Analysis is stopped when a high quality matching is achieved or enough hypotheses have been examined. The number of such hypotheses that need to be examined before stopping appears to be determined heuristically.

Faugeras and Hebert [1986] describe a similar system for the recognition of three-dimensional objects from range data. In this case, hypotheses are generated from small sets of feature matches. Faugeras and Hebert provide various solutions for object pose from sets of features, including a closed-form solution for the case of three point matches. A special data structure is used to determine which model features are brought into alignment with image features under the hypothesized transformation and a tree-search method is used to determined the best overall matching. This method is thus a hybrid alignment/search method; hypotheses are generated using alignment, but search is used to verify and optimize the hypotheses.

2.3.3 SCERPO

The SCERPO system (for Spatial Correspondence, Evidential Reasoning, and Perceptual Organization) has been developed by Lowe [1987]. This system is able to recognize three-dimensional objects from two-dimensional images through alignment techniques. Lowe uses Newton's method to iteratively solve for the pose of the object using the perspective projection given a hypothesized set of matching features. This method requires an initial guess to start iterating from and can encounter local minima. Furthermore, since multiple solutions are possible for small sets of features, several starting points must be tried. After initial hypotheses are generated, additional matches that are brought into alignment are added based on a probabilistic evaluation of how likely it is that the match arose at random. The hypothesized pose is updated at each step. Lowe's work [1985] on grouping (also known as perceptual organization) has been exploited in this system to determine which sets of image features are the best to use as the initial hypotheses.

2.3.4 Huttenlocher and Ullman

Huttenlocher and Ullman [1990] describe an alignment method for the case of weak-perspective transformations. They give a fast method for determining the two transformations that exist under weak-perspective for matches between three model points and three image points. They also prove that the number of solutions for this case is always exactly two. (The solutions are the same if the plane of the model points is perpendicular to the viewing direction.) Huttenlocher and Ullman also describe how an additional virtual point can be computed from two model or image points and associated orientations (although this requires that rays emanating from the point in the direction of the orientations intersect and are thus coplanar.) The computation of virtual points allows the algorithm to examine fewer hypotheses. The overall method of Huttenlocher and Ullman is to examine each possible hypothesis and then use two verification techniques to determine if each hypothesis is correct. The first verification technique examines where the remaining model points are transformed by the hypothesized pose. If enough of these are transformed close to image points, the second verification is also used. This more extensive verification step examines where the model edges are transformed by the hypothesized pose and uses evidence accumulation techniques to try to ensure that the object is actually present in the image.

2.3.5 Basri and Ullman

Basri and Ullman [1988] describe a curvature method of aligning objects with smooth surfaces. Objects are represented by a small number of two-dimensional contours with an associated depth and curvature at each point. Experiments indicate that smooth three-dimensional objects can be modeled effectively in this manner and the appearance of the object from all viewpoints can be accurately predicted. In recognizing objects, the method of Huttenlocher and Ullman [1990] was used to determine hypothesized poses. Basri [1992] analyzes the error in the curvature method of aligning objects with smooth surfaces. He concludes that a small number of models is required to predict how an ellipsoid will appear from all possible viewpoints in an image.

A related system [Ullman and Basri, 1991] described models using linear combinations of a small set of model images. Under an affine transformation, objects with sharp edges can be modeled exactly in this manner. The curvature method is also used with this system for smooth objects.

2.3.6 Error analyses

Error analyses of alignment systems have been performed by several researchers [Grimson *et al.*, 1992a, Grimson *et al.*, 1992b, Alter and Grimson, 1993].

Grimson *et al.* [1992b] analyze the effects of noise on systems recognizing planar objects undergoing affine transformations from point features. They give exact bounds on the location a fourth model point may be in the image given correct matches between three model and image points, where the image points are localized up to a disk of radius ϵ. Similarly, they determine the 'selectivity' of sets of four model points. This is the probability that a random set of four image points could be the projection of the four model points to within the error radii. Their conclusion is that techniques that examine such small sets of points will often produce false positive matches for moderate levels of noise and image complexity.

An extension of this work to three-dimensional models [Grimson *et al.*, 1992a] determines overestimates of the range of transformations that take a set of four three-dimensional model points to within error boundaries of corresponding image points. The occurance of false positives in the three-dimensional case is shown to be significant.

Alter and Grimson [1993] analyze the use of line segment features in the alignment method. They show that the selectivity is expected to be lower for this case and thus false positives will be less common. They also show how to tighten the bounds on the propagated uncertainty regions from [Grimson *et al.*, 1992a] using a numerical technique.

2.4 Pose clustering

The basic idea of pose clustering systems is that if we calculate the pose from every possible combination of the minimal amount data necessary, many of the matches will be correct if the object is present in the image. Each of these will yield a pose close to the correct pose of the object. So, finding clusters of these poses in the pose space will yield hypotheses for the location of the object in the image.

2.4.1 Generalized Hough transform

Ballard [1981] showed how the Hough transform [Hough, 1962, Duda and Hart, 1972] could 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 was 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.

2.4.2 Stockman *et al.*

Stockman *et al.* [1982] describe a pose clustering system for two-dimensional objects undergoing similarity transformations. This system examines matches between image segments and model segments to reduce the subset of the four-dimensional pose space consistent with a 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 multi-dimensional histograming. This histograming is performed in a coarse-to-fine manner to reduce the overall number of bins that must be examined.

2.4.3 Thompson and Mundy

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 histograming is used to determine where large clusters of transformations lie in transformation space, which are assumed to correspond to correct transformations.

2.4.4 Linnainmaa *et al.*

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

2.4.5 Transformation sampling

Cass [1988] describes a method similar to pose clustering that uses transformation sampling for the case of two-dimensional objects undergoing similarity transformations. Cass uses line segments as the features to recognize objects. Instead of histograming 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 the sample points with maximum value. This method may miss a pose that brings many matches into alignment if the sampling is not fine enough, but it ensures that the transformations found for any single sample point are mutually compatible.

2.4.6 Analyses

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

Grimson *et al.* [1992a] examine the effect of noise, occlusion, and clutter for the specific case of recognizing three-dimensional objects from two-dimensional images using point features. They determine overestimates of the range of transformations that take a group of model points to within error bounds of hypothetically corresponding image points. Using this analysis, they show that pose clustering for this case also suffers from a significant rate of false positive hypotheses. A positive sign for pose clustering from the work of Grimson *et al.* is that the alignment method [Huttenlocher and Ullman, 1990] produces false positive hypotheses with a higher frequency than pose clustering when both techniques use only feature points to recognize objects.

2.4.7 Pose constraint methods

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, 1993]. For two-dimensional models undergoing similarity transformations, if each image point is localized up to an uncertainty region described by a k-sided polygon then each of the mn possible 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 producing 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 undergoing affine transformations. Given three correct matches this system can quickly determine the maximal set of matches that includes those matches. Jacobs discretizes the six-dimensional space of possible errors in the locations of the three image feature points. Each bin in this discretization represents a small volume of feature space where the true locations of the three image features could be. There exists 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.

2.5 Indexing

Indexing systems for machine vision attempt to generate a single number (or a finite set of numbers) from an image or images that can be used to index a table and determine the set of models that could have generated the data. Ideally, one is able to represent the set of features comprising all or part of the object as a single number which remains the same regardless of transformation or projection. Such a number is called an *invariant*.

Once invariants have been found, an index table can be created by discretizing the space of invariant parameters. Model features are then stored in the table at locations corresponding to their parameters. At recognition time, the parameters associated with the image features can be used to look up the model features in the index table that may correspond to them.

2.5.1 Lamdan *et al.*

Lamdan *et al.* [1988] describe invariants for two-dimensional point sets (of size four or more) undergoing general three-dimensional affine transformations and orthographic projection. They represent a group of four points in a look-up table by the coordinates of the fourth point p_4 in a coordinate system with origin p_1 and unit axes $\overline{p_1p_2}$ and $\overline{p_1p_3}$. These relative coordinates are invariant to affine transformations. Model groups are then indexed by selecting three image points as a basis and using the relative coordinates of the remaining image points as keys into the index table. Voting is done to determine which objects might be present in the image.

They have extended their system such that it can also deal with the case of three-dimensional models and two-dimensional data. Their system accomplishes this by indexing on groups of five image points. While each model group is represented only once, each image group must index a line in the look-up table. Clemens and Jacobs [1991] show that this method does not take advantage of all of the constraints available and thus unnecessarily produces false positive group matches. The requirement of groups of five points also means that there are $O(n^5)$ image groups and $O(m^5)$ model groups to consider.

2.5.2 Clemens and Jacobs

Despite the lack of general invariants for the three-dimensional case, Clemens and Jacobs [1991] have shown that an indexing system for this problem can be built that (in the noiseless case) indexes exactly those groups that could have projected to a specific image group in the weak-perspective imaging model. This system requires groups to be of (at least) four points. It uses a four-dimensional index table and each group of four points must be represented over a two-dimensional space in this table. The requirement of four points per group means that there are $O(n^4)$ image groups and $O(m^4)$ model groups to consider. While this system achieves greater relative speedup by increasing the size of the point groups examined and the dimensionality of the index table, Clemens and Jacobs show that grouping is necessary for larger point groups to be of significant use, due to the larger number of groups found when the size of the point group is increased.

The four dimensions of the index table used by Clemens and Jacobs are as follows: the relative coordinates (x_4', y_4') of the orthographic projection of the fourth point using the projections of the first three as a basis (note that after projection these two parameters are the same representation used by Lamdan *et al.*,) the angle formed by the projections of the first three points, and the ratio of vector lengths of the projections of the first to second and first to third points. These four parameters are invariant for any group of four points over translation, scaling, and rotation about the viewing direction. These parameters are not invariant over the remaining viewing parameters (i.e. viewing direction,) thus they must represent each group from each viewing direction in their index table. Since the viewing direction has two degrees of freedom, this means that each group must be represented on a two-dimensional surface in the table.

Jacobs [1992] has shown that groups of five three-dimensional points undergoing affine transformations can be indexed from two-dimensional data by representing each group as lines in two orthogonal two-dimensional tables. To determine which model groups may have projected to an image group, model groups are indexed in both of the tables. The intersection of the two sets of indexed groups then corresponds to the possible model groups.

2.5.3 Rothwell *et al.*

Rothwell *et al.* [1993] demonstrate that invariants exist for some constrained classes of three-dimensional point sets. In particular, they describe invariants for point sets that contain several subsets of four coplanar points and point sets that possess bilateral symmetry. Seven feature point matches are required to determine invariants for the first case and eight feature point matches are required for the second.

2.5.4 Stein and Medioni

Stein and Medioni have described systems to index two-dimensional objects from two-dimensional data [Stein and Medioni, 1990] and three-dimensional objects from three-dimensional data [Stein and Medioni, 1992]. Their two-dimensional system approximates objects as polygons. A sequence of consecutive line segments in the approximation is called a super-segment. Super-segments are encoded and stored in a table for lookup at recognition time. The three-dimensional system uses a representation of local surface properties called a splash. Each splash is comprised of the surface normal at a reference point and the surface normals at discrete points on a circle around the reference point. A super-splash consists of several splashes with circles a different radii with the same reference point. Stein and Medioni encode the features of super-splashes for lookup at recognition time.

2.5.5 Model-based invariants

Weinshall [1993] describes invariants for three-dimensional objects that vary with the model (a set of points.) These are functions (that are different for each model) that are invariant to the rotation, translation and projection of the model points onto the image. These invariants can be formulated despite the proof that no invariants exist, because they explicitly use the relevant model parameters in the formulation, and the invariant function is different for each model. While Weinshall describes how indexing tables can be built using them, the dependence of the invariant on the model parameters complicates this somewhat. The model-based invariant is basically a formula that describes when a minimal set of overconstraining matches between model features and image features can be brought into alignment exactly in the noiseless case for both rigid and affine transformations. Noise will cause the invariant to vary from the ideal value.

Weinshall and Basri [1993] investigate a similar concept. They describe a transformation metric that determines how well a set of points can be brought into alignment using rigid transformations. This metric can be used to determine which sets of matches can be brought into alignment in a similar manner to the model-based invariants.

2.5.6 Invariant descriptors

Forsyth *et al.* [1991] describe descriptor functions for planar curves that are invariant to the object's pose under the perspective projection. After describing a number of invariants for various transformations, they show how invariants can be developed using techniques from invariant theory. They then show how recognition can be performed using an invariant for objects consisting of two rigidly coupled planar conics.

Chapter 3

Efficient Pose Clustering

The first application of the concept of selective examination of hypotheses that will be examined in this thesis is based on the idea that the pose clustering method of object recognition can be decomposed into small subproblems. In this chapter, I show how such a decomposition can be obtained without loss of accuracy. This decomposition allows pose clustering to be formulated as a generate-and-test algorithm. Each subproblem examines a hypothesis that a match between two image points and two corresponding model points is correct.

In addition, randomization techniques can then be used to limit the number of such hypotheses that need to be examined to gain accurate recognition. While no criterion is yet used to determine which hypotheses are best to examine prior to the clustering stage, only a selected number of the hypotheses are examined, and the decomposition techniques allow the incorporation of probabilistic information about the likelihoods of the hypotheses prior to the pose estimation and clustering step where the bulk of the computation is performed.

3.1 Introduction

Pose clustering is a technique that is used to recognize objects in images from hypothesized matching between feature groups [Ballard, 1981, Stockman, 1987, Thompson and Mundy, 1987, Linnainmaa *et al.*, 1988, Grimson *et al.*, 1992a]. In this method, the transformation parameters that align groups of model features with groups of image features 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. Unfortunately, Grimson *et al.* [1990a, 1992a] 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. Further discussion of previous work on pose clustering can be found in Chapter 2.

For the remainder of this chapter, I will focus on the recognition of general three-dimensional objects undergoing unrestricted rotation and translation from single two-dimensional images. To simplify matters, the only features used for recognition are feature points in the model and the image, but the results here can be generalized to any features from which we can estimate the pose of the object.

If m is the number of model feature points and n is the number of image feature points then there are $O(m^3n^3)$ possible transformations to consider for this problem. 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 an insignificant probability of missing a correct object, yielding an algorithm that requires $O(mn^3)$ total time. Additional speedup can be achieved by using grouping 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.

The remainder of this chapter will be structured as follows. Section 3.2 discusses some previous techniques used to perform pose clustering. Section 3.3 proves that examining small subsets of the possible transformations is adequate to determine if a cluster exists with optimal accuracy and discusses the implications of this result on pose clustering algorithms. Section 3.4 discusses the computational complexity of these techniques and compares it to other algorithms for this problem. Section 3.5 gives an analysis of the frequency of false positives using the results on the correlation between transformations to achieve more accuracy than previous work. Section 3.6 describes how clustering can be performed efficiently and discusses the implementation of these ideas. Experiments that have been performed to demonstrate the utility of the system are presented in Section 3.7. Section 3.8 discusses some interesting issues and a summary can be found in Section 3.9.

3.2 Recognizing objects by clustering poses

As mentioned above, pose clustering is an object recognition technique where the poses that align hypothesized matches between feature groups are determined. Clusters of these poses indicate the possible presence of an object in the image.

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 hypothetical poses of the object. It is well known that three matches between model points and three 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 using the perspective projection and several approximations [Fischler and Bolles, 1981, Huttenlocher and Ullman, 1990, DeMenthon and Davis, 1992, Alter, 1992] (Figure 3.1.) 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 $\gamma = \{(\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3)\}$

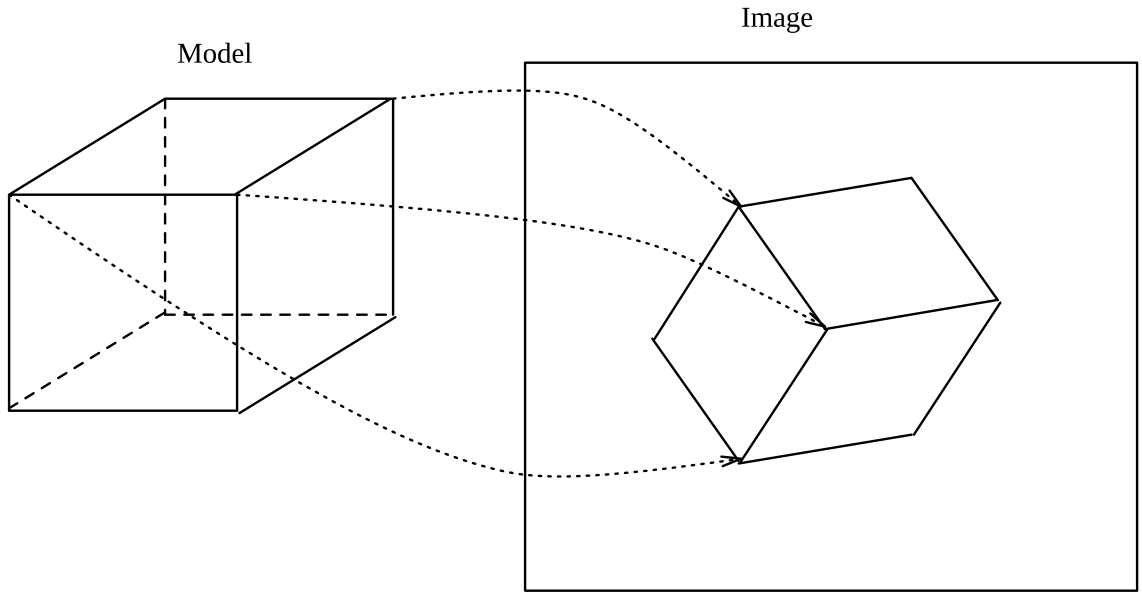

Figure 3.1: There exist a finite number of transformations that align three non-collinear model points with three image points.

will be called a *group match.*

If there are m model features and n image features then there are $6\binom{m}{3}\binom{n}{3}$ distinct group matches (since each group of three model points may match any group of three image points in six different ways,) each of which yields up to four transformations. 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 bounds. Work in this direction has been undertaken by Cass [1988, 1990], but these methods can be time consuming and are difficult for the case of three-dimensional objects.

Most pose clustering algorithms find clusters less accurately by histograming the poses in the multi-dimensional transformation space (Figure 3.2.) In this method, each pose is represented by a single point in pose space. The pose space is discretized into bins and the poses are histogramed in these bins to find large clusters. Since pose space is six-dimensional for general rigid transformations, the discretized pose space is enormous for the fineness of discretization necessary to perform accurate pose

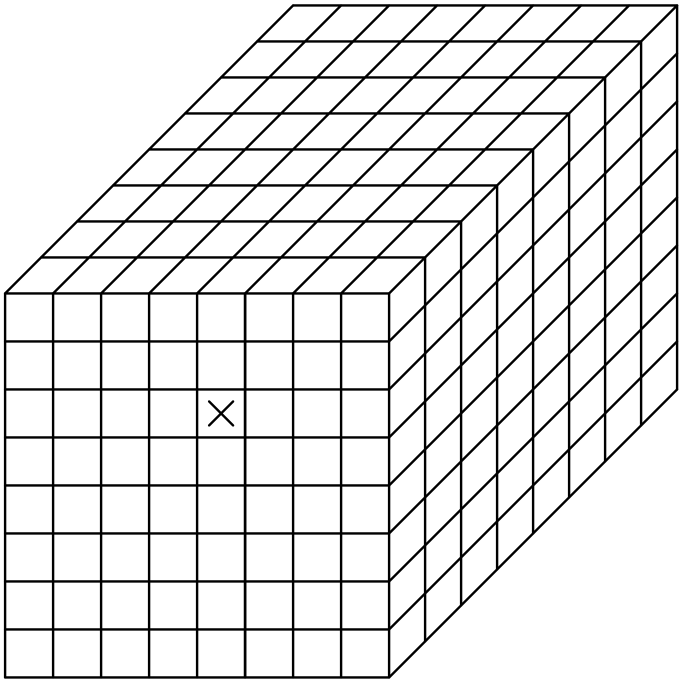

Figure 3.2: Clusters representing good hypotheses are found by performing multi-dimensional histograming on the poses. This figure represents a coarsely quantized three-dimensional pose space.

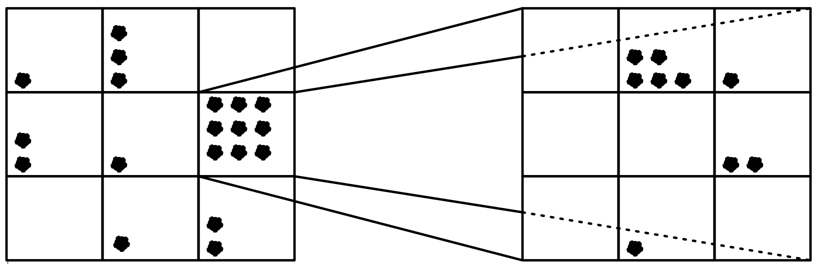

Figure 3.3: In coarse-to-fine clustering, the bins at a coarse scale that contain many transformations are examined at a finer scale.

clustering.

Two techniques that have been proposed to reduce this problem are coarse-to-fine clustering [Stockman *et al.*, 1982] and decomposing the pose space into orthogonal subspaces in which histograming can be performed sequentially [Thompson and Mundy, 1987, Linnainmaa *et al.*, 1988]. In coarse-to-fine clustering (Figure 3.3,) 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 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 3.4.) 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 [1990a] 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 recursive 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

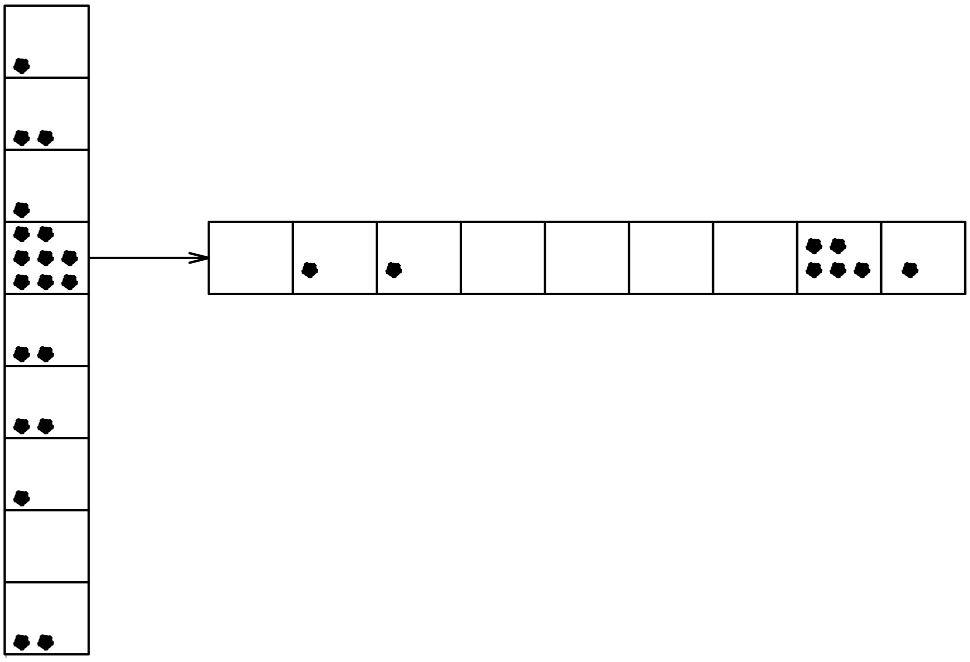

Figure 3.4: Pose space can be decomposed into orthogonal spaces. Clustering is then performed in one of the decomposed spaces. Bins that contain many transformations are examined with respect to the remaining spaces.

storage for previous methods 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 [Cass, 1993].

3.3 Decomposition of the problem

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 $\gamma = \{(\mu_1, \nu_1), (\mu_2, \nu_2), (\mu_3, \nu_3)\}$ yields 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 for each $p \in \Theta$:

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

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

2. There exist x distinct point matches $\pi_1, ..., \pi_x$ that pose p 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 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 each basis, we must examine $O(mn)$ group matches, since there are $(m-2)(n-2)$ group matches for a single basis 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 point matches, but we will see in the next section that we can examine $O(n^2)$ pairs of matches and achieve accuracy arbitrarily close to optimal.

Figure 3.5 gives the updated pose clustering algorithm.

3.4 Computational complexity

This section discusses the computational complexity necessary to perform pose clustering using the techniques described above. We can use a randomization tech-

```
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 3.5: New pose clustering algorithm.

nique proposed by Fischler and Bolles [1981] and also used by Lamdan *et al.* [1988] and Cass [1993] 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 (f is the fraction of model points appearing,) we can determine an upper bound on the probability of not choosing a correct image basis in k trials, where each trial consists of examining a random basis of two image points. Since the probability of a single image point being a correct model point is at least $\frac{fm}{n}$, the maximum probability of a basis being incorrect is approximately $1 - (\frac{fm}{n})^2$. Thus, the probability that k random trials will all be unsuccessful is:

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

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: $k_{\text{min}} = \frac{n^2}{(fm)^2}\ln\frac{1}{\delta}$)

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 $1-\delta$. 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. Since we must examine $O(mn)$ group matches for each basis, this method requires $O(mn^3)$ time per object in the database, where previously $O(m^3n^3)$ was required. A comparison against other algorithms for the problem of recognizing three-dimensional objects may be useful.

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 $O(m\log n)$ step is performed for each match to determine if enough model points are brought into alignment with image points by this transformation, yielding a total time of $O(m^4n^3\log n)$. Randomization can be used to limit the number of group matches examined. In this case, the running time of the alignment method is $O(mn^3\log n)$. 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 this case.

3.5 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.* analyze the pose clustering approach to object recognition to determine the probability of a false match having a large peak in transformation space for the case of recognition of three-dimensional objects from two-dimensional images. They use the Bose-Einstein occupancy model (see, for example, [Feller, 1968]) to estimate this probability. This analysis assumes independence in the locations of the transformations, which is not correct. Consider two group matches composed of a total of six distinct point matches. If there is some pose $p \in \Theta$ that brings both group matches into alignment up to the error conditions, then any of the $\binom{6}{3} = 20$ group matches that can be formed using the six point matches is also brought into alignment by this pose. Thus the poses determined from these group matches are highly correlated.

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

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

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

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

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:

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

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

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

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

Actually, $\frac{1}{b_p mn}$ isn't always small, but this approximation yields a conservative estimate.)

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

Note that this is not very different from the result derived by Grimson *et al.* since $b_p = \sqrt[3]{b_g}$ if the clustering is performed exactly as in the method of Cass [1990]. The primary difference is a change from a factor of $\sqrt[3]{\ln \frac{1}{\delta}}$ to $\ln \frac{1}{\delta}$, which means that the new estimate of the allowable number of image features before a given rate of false positives is produced is lower than that obtained by Grimson *et al.*

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

One solution to this problem would be to use more descriptive features. The results presented here, are easily generalized to encompass features conveying more information. This will increase the allowable clutter, but some bound will still be applicable. Grimson and Huttenlocher [1991] have performed a similar calculation for line segment features from planar models undergoing rigid planar transformations. This analysis does not suffer from the problem of correlated transformations since, in this case, each hypothesis consisted of a single match between a model feature and an image feature.

The primary implication of these results 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 finding likely hypotheses for further verification. As an additional verification step, we could, for example, verify the presence of edge information in the image as done by Huttenlocher and Ullman [1990].

3.6 Efficient clustering

This section discusses how my system performs clustering of the poses 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.

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

I still use histograming to achieve the fastest possible 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 is to prevent any clusters from being missed due to them falling on a boundary between bins. This method should be able to find almost all of the correct transformations, but it does not have optimal accuracy. More accurate techniques (e.g. [Cass, 1990]) may be used at the cost of lower speed.

My implementation uses the method of Huttenlocher and Ullman [1990] to determine the transformation parameters that bring three model points into alignment with three image points in the weak-perspective imaging model. Varying image noise levels are accounted for in my implementation by varying the size of the bins used in the histograming procedure.

Since histograming 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 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[1] 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. For each bin with enough transformations, we recursively cluster the poses in that bin using the remaining parameters. Since this procedure continues until all six parameters have been examined, the bins in the final step contain transformations that agree closely in all six of the transformational parameters and thus form a cluster in the complete pose space.

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

Figure 3.6 gives an outline of this algorithm. If unexamined parameters remain at the current branch of the tree, we histogram the remaining poses using one of these parameters. Each of the bins that contains at least $fm - 2$ poses is then clustered recursively using the remaining parameters. The other bins are pruned. When we reach a leaf bin (after all of the parameters have been examined) that contains enough poses, we output the location of the cluster.

Although this decomposition of the cluster algorithm has not previously been

[1] For the moment, let's neglect the possibility that we may not find some of the correct transformations. In this case, if fm model points are present in the image, a correct basis will yield $fm - 2$ correct transformations

```
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 3.6: Recursive clustering algorithm. (See text.)

formulated as a tree search, the analysis of Grimson and Huttenlocher [1990a] implies that previous pose clustering methods saturate such decomposed transformation spaces at the levels of the tree near the root, due to the large number of transformations that need to be clustered. For those methods, virtually none of the branches near the root of the tree can be pruned.

Since previous systems would 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 histograming in a high-dimensional space, these systems may have a large number of unpruned bins at even low levels of the tree, since they are clustering so many transformations. Using the techniques presented here, we 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.

Once clusters are found, I use a method described by 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 lesser of the number of distinct model points and distinct image points matched in the cluster, since it can be determined quickly (as opposed to the maximal bipartite matching) and is reasonably accurate.

Note that the alignment method [Huttenlocher and Ullman, 1990] cannot use this method of reducing the number of false positives without increasing the time bound. This is because the $O(m \log n)$ step to determine how many points are brought into alignment by a given transformation checks only to see if each transformed model point is close to any image point. It cannot find the maximum number of image points that are covered by the transformed model points without increasing the complexity of the algorithm.

3.7 Results

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

3.7.1 Synthetic data

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

1. Model points were determined randomly inside a 200 × 200 × 200 cube.

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.

4. In some experiments, additional random image points were added.

	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 3.1: The performance finding correct clusters. m=the number of object points; optimal=the size of the optimal cluster; average=the average size of clusters found; %=the average fraction found of the optimal cluster.

The first experiment determined whether the correct clusters were found. Table 3.1 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 correct poses in the best cluster. 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 3.2 shows the average size of the largest cluster found for each image basis, the standard deviation among these clusters, and the size of the largest cluster over all of the image bases. Since the system found correct clusters of average size 15.02 for models of twenty points and false positive clusters of average size 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	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 3.2: The size of false positive clusters found for objects with 20 feature points. n=the number of image points; average=the 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.

An experiment determining the number of trials necessary to recognize objects in the presence of random extraneous image points was run. Table 3.3 shows the results of this experiment. To generate a hypothesis of the model being present in the image, this experiment required a cluster to be at least 80% of the optimal size (14 for models of size 20.) We cannot assume that each correct basis will result in 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 3.3 shows k_{limit} for $\delta = 0.01$, the average number of trials necessary to generate a correct hypothesis that the object was present in the image, the maximum number of trials necessary to generate such a hypothesis, and the number of objects (out of 100) that required more than k_{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.

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 3.3: The number of trials required to find objects with 20 points. n=the number of image points; k_{limit}=the 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 an object; over=the number of objects (out of 100) that required more than k_{limit} trials to recognize.

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

3.7.2 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 on them. Furthermore, the only features used in either data set to generate hypotheses are the locations of corner points in the image.

Hypothesis generation followed these steps:

1. Object models were created. For the first data set this was done by capturing images of the object and measuring the location of corners. For the second data set this was done by hand.

2. Images including the object models were captured.

3. Corners were detected in the images using a fast and precise interest operator [Förstner and Gülch, 1987, Förstner, 1993].

4. The model and image feature points were used by the pose clustering system to generate hypotheses as specified in this chapter.

5. For each hypothesis, the pose was determined by averaging the poses in the cluster.

Figure 3.7 shows an example of recognizing objects from the first data set in an image. The top image shows the 84 feature points found by the interest operator. The bottom image shows the best hypotheses found for this image. Figure 3.8 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.

3.8 Discussion

Some of the techniques described in this chapter 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

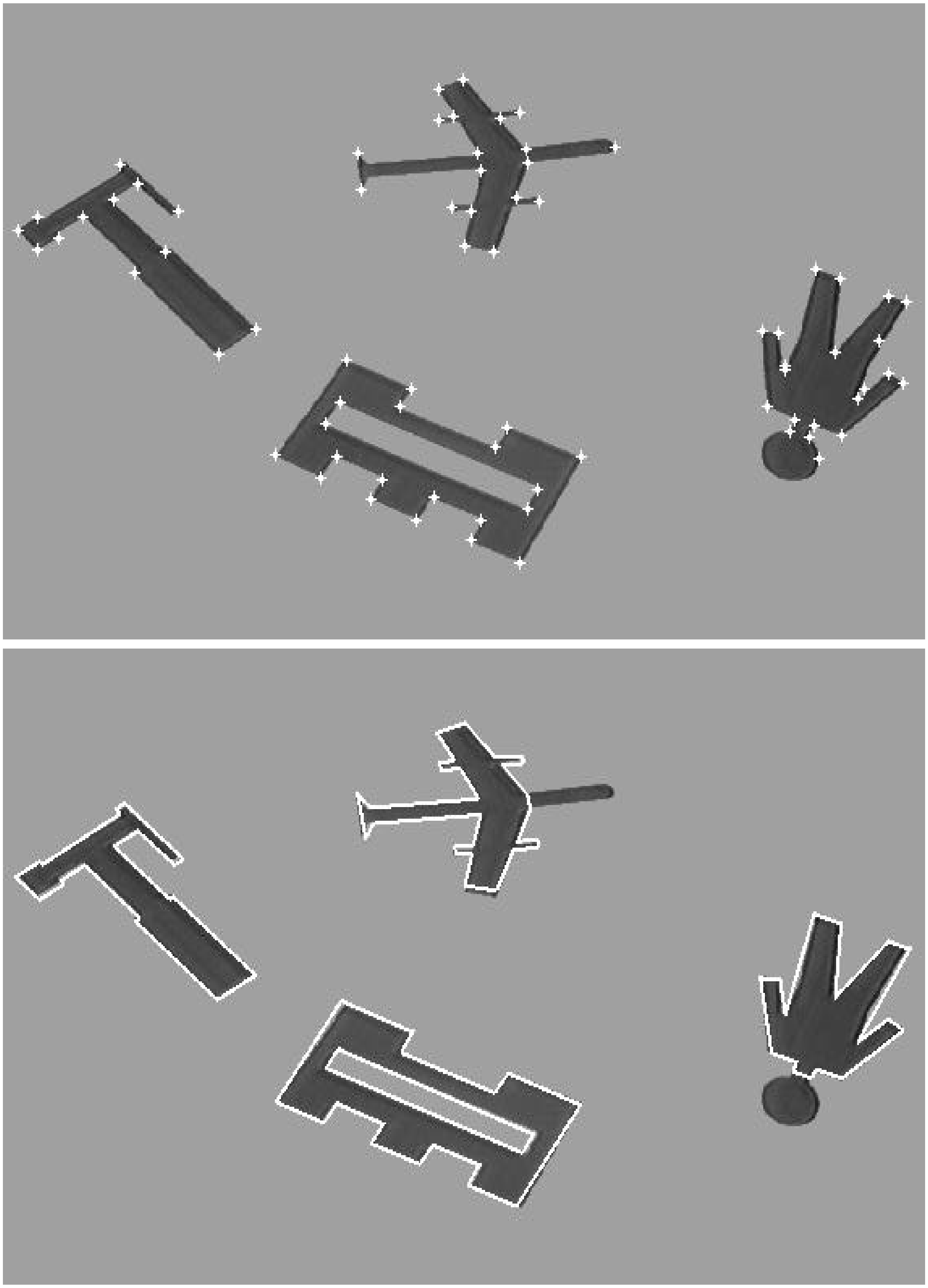

Figure 3.7: Recognition example for 2D objects. Top: The corners found in an image. Bottom: The four best hypotheses found, with edges drawn in. (The nose of the plane and the head of the person do not appear because they were not in the models.)

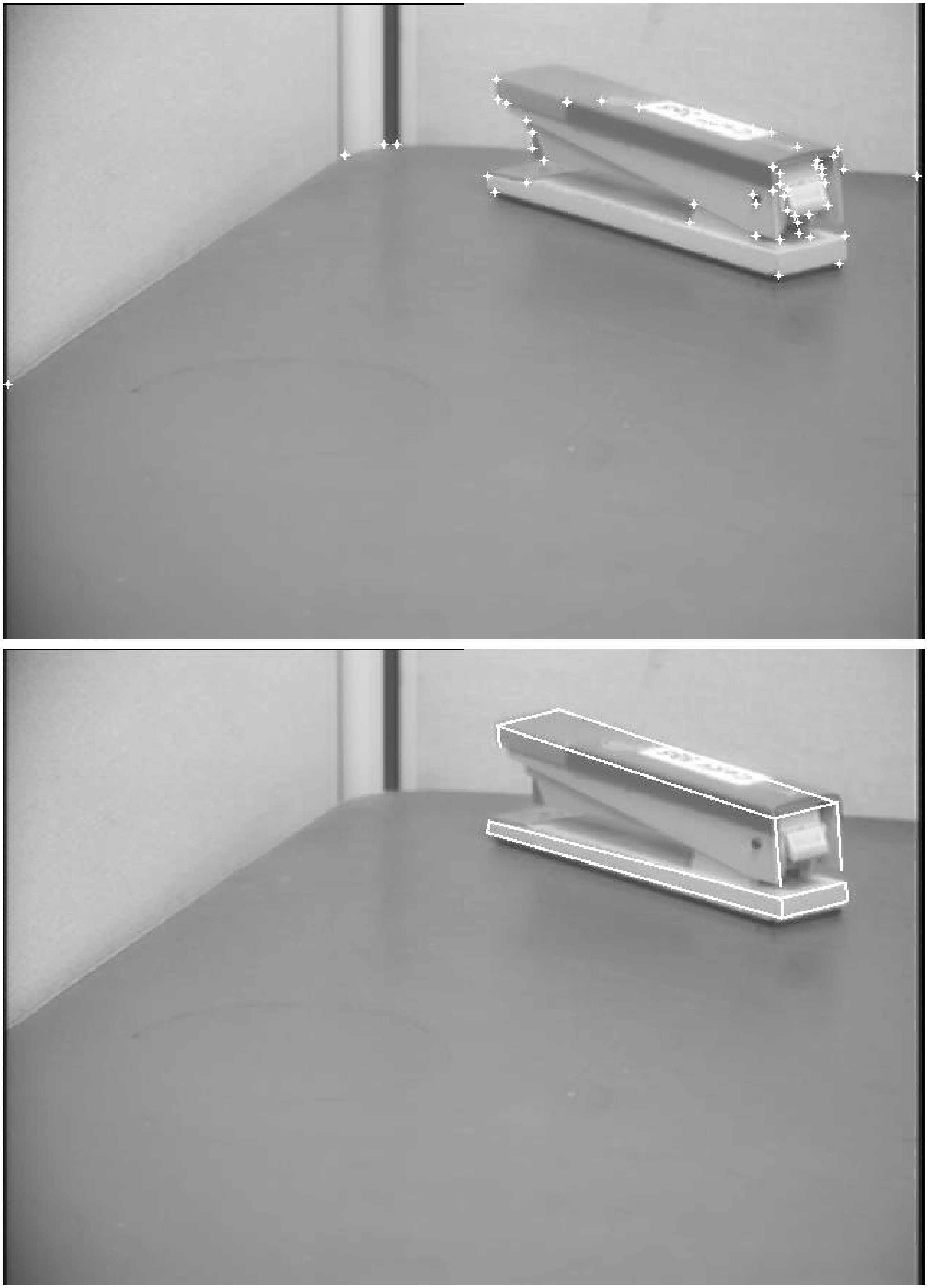

Figure 3.8: Recognition example for a 3D object. Top: The features found in the image. Bottom: The best hypothesis found.

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 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 randomization) yielding a total time of $O(mn^3)$, as with our pose clustering algorithm. 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 algorithm.

Another point worthy of discussion 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, while others have claimed that pose clustering is more sensitive to noise and clutter than other algorithms. Grimson *et al.* [1990a, 1992a] have shown that we should not simply assume large clusters are instances of the object; additional verification is needed to ensure against false negatives. But, 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 3.5 shows that pose clustering is not inherently more sensitive to noise and clutter than other algorithms.

Clutter appears to affect pose clustering similarly to other algorithms. On the other hand, noise is handled in considerably different ways among various algorithms. While considerable research has gone into analyzing how to best handle noise in the alignment method [Jacobs, 1991, Grimson *et al.*, 1992b, Alter and Grimson, 1993], very little has been done in this regard for pose clustering. Work by Cass [1990, 1992] shows how to handle noise exactly in the context of transformation equivalence analysis for the case where the noise is bounded by a polygon, but this is not directly applicable to pose clustering. At present, my system handles noise heuristically and further study in this area will be beneficial.

We can compare the noise sensitivity of pose clustering to alignment. While the alignment method ensures that each of the additional point matches for each basis match can be brought into alignment simultaneously with the basis point matches up to some error bounds, it does not guarantee that all of the additional point matches can be brought into alignment simultaneously with each other. Ideally, a pose clustering system could guarantee this, but due to the limitations imposed by discretizing the pose space and the heuristic handling of noise it is not achieved by my system. The analysis of Grimson *et al.* [1992a] indicates that pose clustering techniques will find fewer false negatives than the alignment method for similar levels of noise and clutter.

3.9 Summary

This chapter has shown that pose clustering can be performed much more efficiently than previously thought. The key to this efficiency gain has been the decomposition of the problem into small subproblems, corresponding to examining hypotheses that matches between two model points and corresponding image points are correct. Randomization has been used to limit the number of such hypotheses that need to be examined to gain accurate recognition. Since far fewer transformations are clustered at a time, this method can be implemented using much less memory than previous pose clustering systems. 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.

Chapter 4

Probabilistic Indexing for Object Recognition

This chapter discusses a method of determining which hypothetical matches between three model points and three images points are likely to be correct using only the information given by the locations of the points. Thus, if we are interested in hypotheses between matches of three model and image points, this technique gives us a method of selectively examining the hypotheses that are most likely to be correct.

4.1 Introduction

Indexing systems determine which groups of model points could have projected to specific groups of image points, eliminating the need to consider other groups of model points as possible matches for that image group. Indexing is key to the fast implementation of generate-and-test techniques, since indexing allows us to determine a priori which of the generated hypotheses are most likely to be correct.

Indexing systems require point groups to be of some minimum cardinality to perform their function correctly. A previous indexing system for indexing general three-dimensional model groups from two-dimensional image groups [Clemens and Jacobs, 1991] required groups of size at least four and each group was represented on a two-dimensional surface in a four-dimensional table. By using a probabilistic method that allows false negatives (matches between sets of model features and image features that are correct, but are not indexed), I have designed a system which can index using groups of size three, and represents each group in a single bucket in a two-dimensional look-up table. These false negatives are overcome by ensuring that we examine several correct hypotheses.

The ability to index on groups of size three is very important. If there are n image points and m image points, then there are $O(n^k)$ image groups and $O(m^k)$ model groups of size k, so reducing the required group cardinality necessary reduces the number of groups to consider immensely. Several algorithms (e.g. [Huttenlocher and Ullman, 1990, Lowe, 1987, Lamdan *et al.*, 1988]) use matches of three image points to three model points as the initial hypotheses because this is the minimum number necessary to determine a finite set of transformations that bring the points into alignment. Indexing systems that require groups of larger than three points cannot generate ideal initial matches for these algorithms.

I use the probabilistic peaking effect [Ben-Arie, 1990, Binford *et al.*, 1989, Burns *et al.*, 1990] to discriminate between likely matches and unlikely matches. The principle of the probabilistic peaking effect is that angles and ratios of distances between points in the model groups do not vary much as the viewpoint changes over much of the viewing sphere. This means that the probability density functions of these angles and ratios of distances of projected (image) points have a strong peak at the pre-projection (model) value. Binford *et al.* call such features 'quasi-invariants' because of their relative lack of variation with the change of viewpoint.

Let us now call a set of image points hypothetically grouped for use in indexing the table an *image group*, and the model points hypothetically matched to them a *model group*. If each of the points in the image group is a result of the projection of its corresponding model group point then we will say that the two groups are in *actual correspondence*. The premise of this system is that the probabilistic peaking effect is a strong enough indicator of model feature values to eliminate the vast majority of model groups which are not in actual correspondence with a specific image group while keeping a significant percentage of those that are in actual correspondence.

Ben-Arie [1990] gives an equation to approximate the joint probability density function of features for model groups of size three, but this approximation does not vary explicitly with the angle formed by the model group, only with the ratio of image angle to model angle, while the true effect varies over both quantities. I have recreated the experiments for determining the probabilistic peaking effect using the model angle as an additional parameter to achieve more accuracy modeling the effects of varying the model angle.

To incorporate the probabilistic peaking effect into a probabilistic indexing system, we can use the joint probability density functions to determine what ranges of parameters a model group may have and still be likely to have generated a specific image group. This determines which groups are indexed. This system can be extended to model and image groups larger than three points. This allows us to achieve better accuracy in discriminating between correct and incorrect matches, although it will take longer to index these matches.

The remainder of this chapter will be structured as follows: Section 4.2 describes the probabilistic peaking effect. Section 4.3 discusses how this effect can be used to build an indexing system. This is followed by a description of how the system can be extended to use image and model groups larger than three points in Section 4.4. Section 4.5 gives indexing results on real images. Finally, I discuss interesting issues and summarize the chapter in Sections 4.6 and 4.7.

4.2 The probabilistic peaking effect

While it has been proven that there is no affine or projective invariant for general three-dimensional point sets [Burns *et al.*, 1990, Clemens and Jacobs, 1991], it has been observed that there is a strong peaking effect in the probability densities of many angles and ratios of lengths in images at the values taken by the features in the model [Ben-Arie, 1990, Binford *et al.*, 1989, Burns *et al.*, 1990]. For example, the probability density of an angle formed by three ordered points in an image has a strong peak at the actual angle formed by the points in real space, assuming that every viewing direction is equally likely. This means that there is a large range of viewing directions over which the angle formed by these points in the image changes

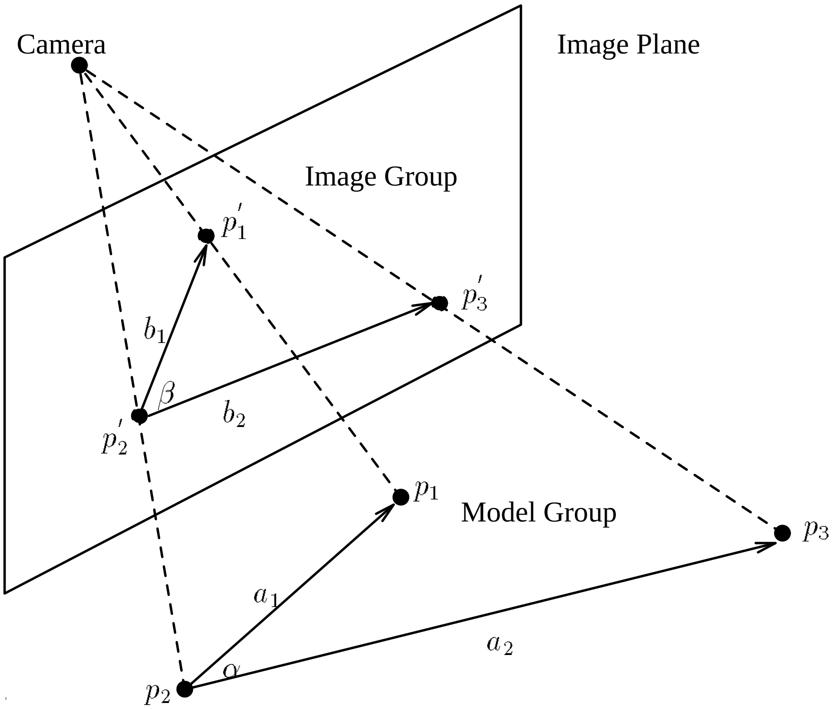

Figure 4.1: An example of a model group projected onto the image plane using the perspective projection.

little. Binford *et al.* called such values 'quasi-invariants.' This information can be used to discard matches between groups of image points and model points that have a small likelihood of being in actual correspondence.

The values used to determined which feature groups are likely to match in this system are determined as follows. Let p_1, p_2, and p_3 be the points in the model group and $p_1^{'}$, $p_2^{'}$, and $p_3^{'}$, be the corresponding image points. Also, let α be the angle $\angle p_1 p_2 p_3$ and β be the angle $\angle p_1^{'} p_2^{'} p_3^{'}$. Define the segment lengths as follows: $a_1 = |\overline{p_1 p_2}|, a_2 = |\overline{p_2 p_3}|, b_1 = |\overline{p_1^{'} p_2^{'}}|, b_2 = |\overline{p_2^{'} p_3^{'}}|$. See Figure 4.1. The features used are:

1. The angles formed by the points in the model (α) and in the image (β).
2. The ratios of the lengths of the segments in the model and in the image ($\frac{a_1}{a_2}$ and $\frac{b_1}{b_2}$).

Ben-Arie [1990] gives an equation to approximate the probabilistic peaking effect as it varies over $\frac{\beta}{\alpha}$ and $\frac{b_1 a_2}{b_2 a_1}$. It should be noted that the peaking effect varies not only with the ratios $\frac{\beta}{\alpha}$ and $\frac{b_1 a_2}{b_2 a_1}$, but also with α (or alternatively with α, β, and $\frac{b_1 a_2}{b_2 a_1}$). Ben-Arie's approximation of the joint probability density does not model this effect. To better model the probabilistic peaking effect, I have created probability histograms with the additional variable α through numerical integration. Like the experiments performed by Ben-Arie, the viewing sphere was tessellated and the area of each tessellation was added to the bucket corresponding to the image angle β and the logarithm of the ratio of lengths ($\log \frac{b_1 a_2}{b_2 a_1}$) from the viewing direction at the center of the tessellation. Ben-Arie uses buckets that vary uniformly with $\log \frac{\beta}{\alpha}$ to measure the angle variation. I use buckets varying uniformly in β because it has explicit bounds (0,180) and because the variation in α is modeled explicitly. Since it is unclear how the objects in the images will be distributed with respect to distance from the camera, the orthographic projection was used in these experiments. Note that using the orthographic project, the probability density varies only with $\frac{b_1 a_2}{b_2 a_1}$. Changing the pre-projection ratio of lengths $\frac{a_1}{a_2}$ has no effect on this probability density.

The result of these numerical integrations is an array of two-dimensional joint probability histograms, where Ben-Arie had a single joint probability density. Figure 4.2 shows the probability histograms in the noiseless case for selected values of α. As expected, the closer the model angle α is to 0 or 180 degrees, the stronger the peak.

To account for noise, I have also generated the probability histograms with bounded noise ($\epsilon = 1.0$ and 3.0) added to the image parameters. The bounded noise model specifies that the true location of each image feature is within some distance ϵ of the measured location. The bounded noise model is used here for simplicity, but any noise model can be accommodated with in this fashion. This method averages the effects of noise on groups at different scales. It is possible that more accuracy could be obtained by treating each scale separately, but the gain would be small for the considerable extra work necessary. Figure 4.3 shows the joint probability histograms for the case with noise ($\epsilon = 1.0$).

This method of accounting for noise should be adequate if we are dealing with images with approximately the same noise distribution. If we examine a number of

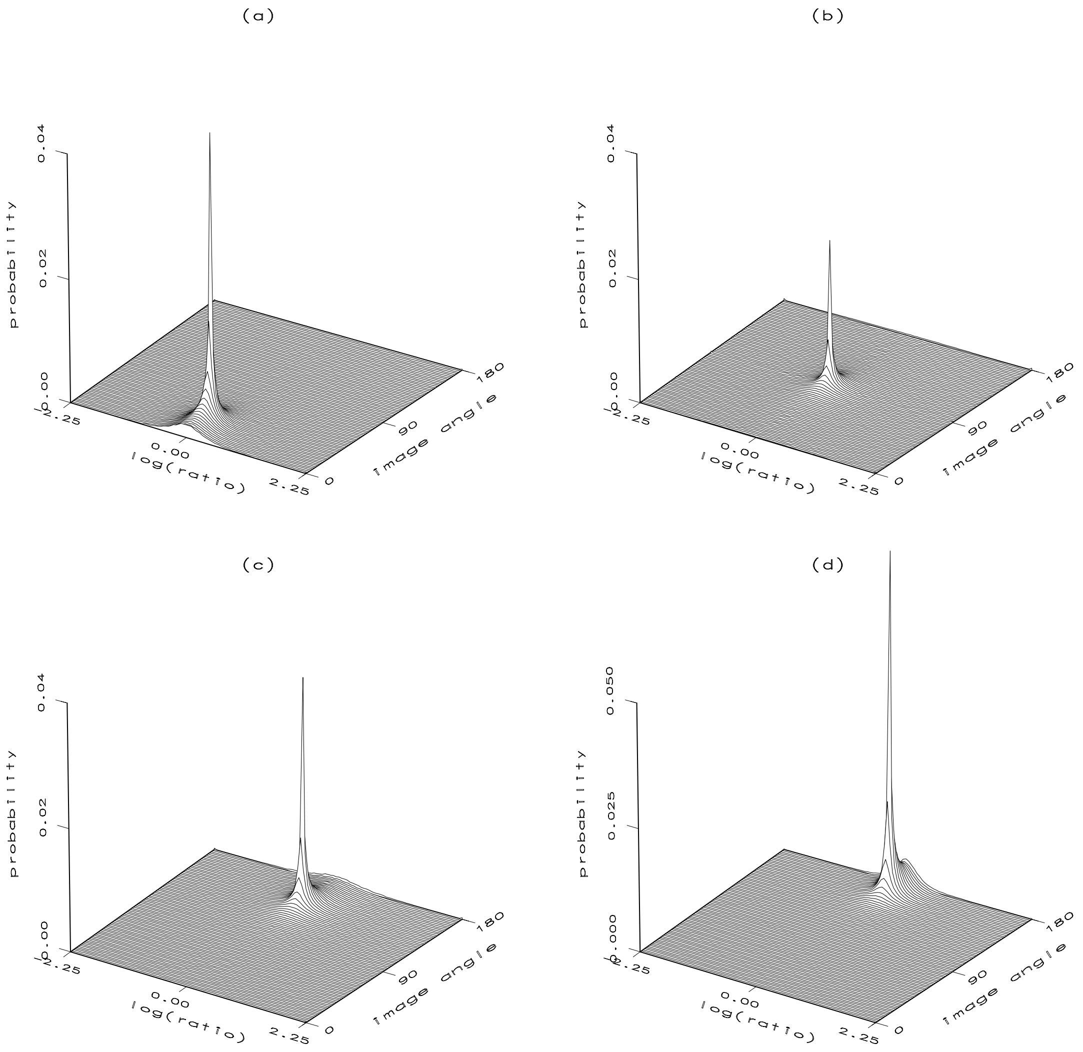

Figure 4.2: The joint probability histograms describing the probabilistic peaking effect for selected values of model angle α with no noise. The x-axis is the image angle β. The y-axis is the logarithm of the ratio of lengths $\log \frac{b_1 a_2}{b_2 a_1}$. The z-axis is the probability. (a) $\alpha = 30°$ (b) $\alpha = 90°$ (c) $\alpha = 140°$ (d) $\alpha = 160°$.

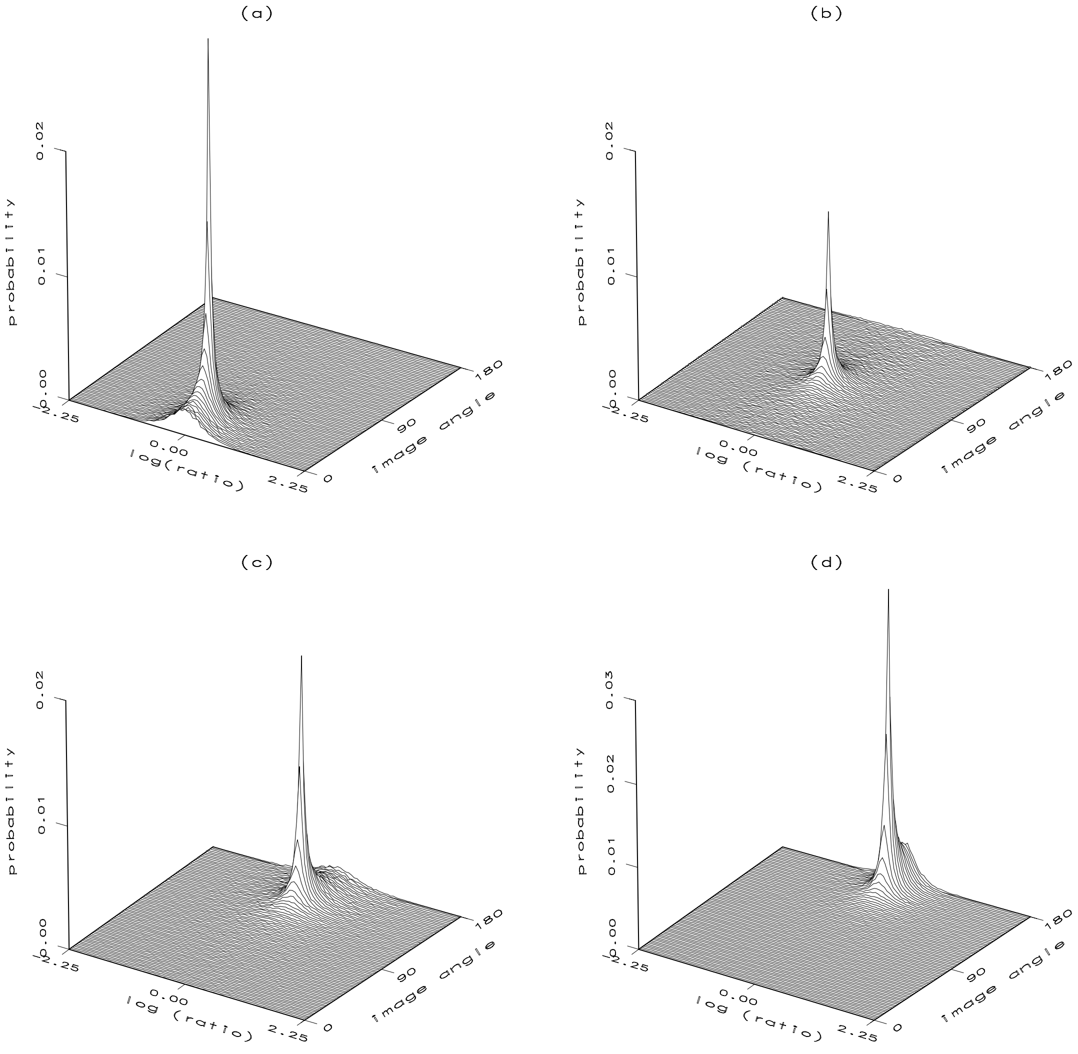

Figure 4.3: The joint probability histograms describing the probabilistic peaking effect for selected values of model angle α with noise ($\epsilon = 1.0$). The x-axis is the image angle β. The y-axis is the logarithm of the ratio of lengths $\log \frac{b_1 a_2}{b_2 a_1}$. The z-axis is the probability. (a) $\alpha = 30°$ (b) $\alpha = 90°$ (c) $\alpha = 140°$ (d) $\alpha = 160°$.

images with different noise distributions we may not want to store several of the joint probability histograms, since they are large. An alternative would be to determine the possible ranges of the true values of the image features at run time from the observed values and the noise distribution. We would then determine which look-up table buckets must be examined using these ranges of values. This alternative has the disadvantage of slower run-time operation.

4.3 Probabilistic indexing

The probabilistic peaking effect can be used to create a probabilistic indexing system to determine which model groups are most likely to have projected to specific image groups. The first step is to create a look-up table containing the model group information. The angle (α) and ratio of lengths ($\frac{a_1}{a_2}$) are determined for each model group in each model and the necessary information about these model groups is stored in the appropriate bucket in the table. This table is quantized in the same manner as the peaking effect probability histograms to facilitate indexing. Note that this table is two-dimensional and each model group is stored in a single bucket.

To determine which model groups are likely to have projected to an image group, we search the probability histograms described in the previous section. The parameters over which this search must vary are the angle α (this determines which histogram we examine) and the ratio $\frac{b_1 a_2}{b_2 a_1}$ within each histogram. We do not need to vary the angle β within each histogram because this is fixed by the image group angle. Since the probability of a particular set of image features is highest when the model values are the same as the image features, we search outward from the bucket corresponding to the image feature values to determine which buckets in the look-up table we must examine. This search determines an area of buckets in the index table that I call a cloud. Each bucket in the cloud is examined for model groups that may match this image group.

Let $r(\beta)$ be the row of the index table corresponding to the image angle and $c(\frac{b_1}{b_2})$ be the column corresponding to the image ratio of lengths. Figure 4.4 shows an example cloud in the look-up table. Note that it is centered at the bucket corresponding

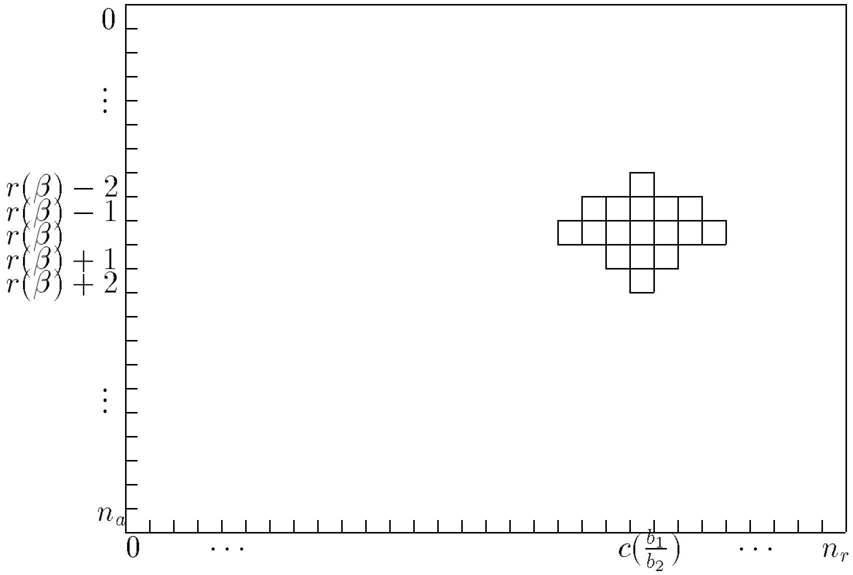

Figure 4.4: An example cloud of buckets in the index table. The hash marks along the x-axis represent the discretization of the possible values of $\frac{b_1}{b_2}$. The hash marks along the y-axis represent the discretization of the possible values of β. The marked bins in the middle are the bins that were found to have large enough probability of holding a correct match (i.e. the cloud.)

to the image feature values.

The extent of a cloud is determined as follows: for each angle α, we examine the row corresponding to the image group angle β and determine what range (if any) of ratios $\frac{b_1 a_2}{b_2 a_1}$ has a probability above a predetermined constant. (This constant is determined a priori to eliminate most groups not in actual correspondence, while keeping a large number of those that are. See below.) This provides the information to determine which buckets in the look-up table contain the model groups most likely to match this image group. For each α, we determine the range of ratios $\frac{a_1}{a_2}$ that should be examined in the look-up table from the range of $\frac{b_1 a_2}{b_2 a_1}$ determined as described above and $\frac{b_1}{b_2}$ from the image group. Each model group contained in these buckets is considered as a possible match for the current image group. We do not need to worry about noise in the index features when indexing because we have already accounted for it in the probabilistic peaking effect probability histograms.

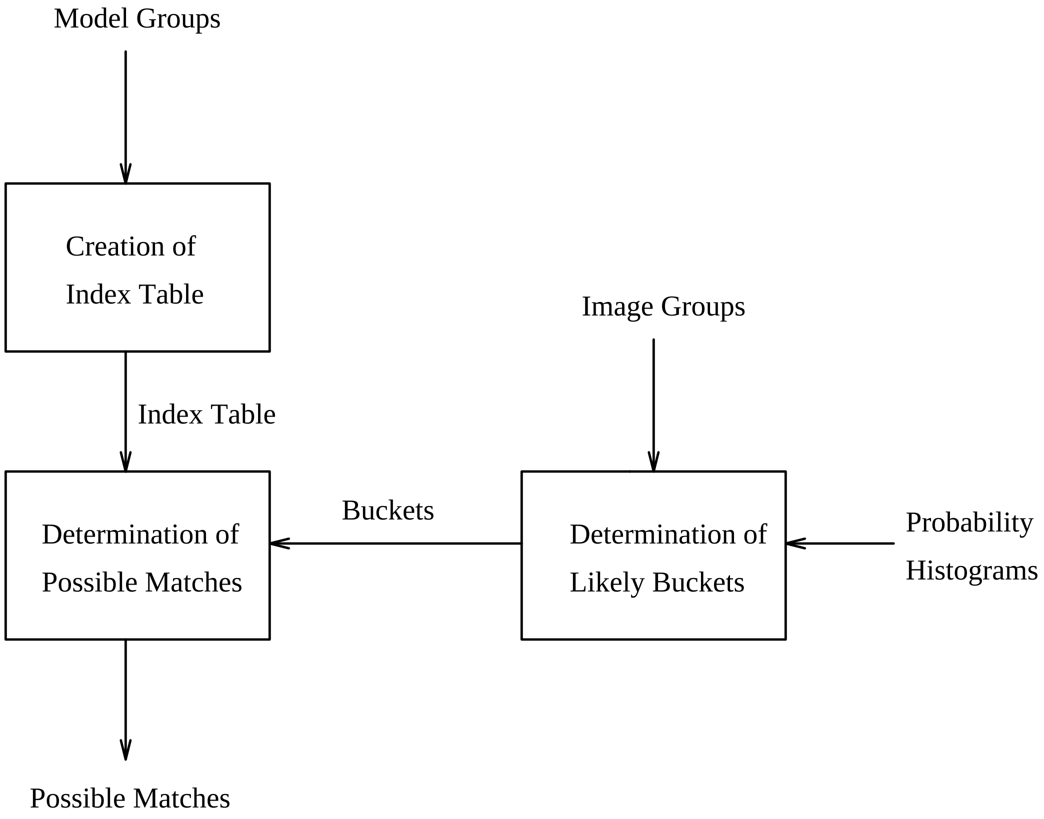

Figure 4.5: The flow of information in the indexing process.

Figure 4.5 illustrates the flow of information in the probabilistic indexing process. The index table is created by storing the model groups in the appropriate buckets. Image groups and probability histograms are used to determine which buckets in the index table are most likely to contain the matching model group. These buckets are then examined in the index table to find possible matches for the image group.

Table 4.1 shows the percentage of total matches and matches in actual correspondence indexed for various probability thresholds as determined by experiments on objects of random three-dimensional points. These experiments transformed the models by a random three-dimensional rotation and projected them using the perspective projection. Bounded noise ($\epsilon = 1.0$) was added to each of the feature coordinates.

The probability of indexing a correct match is expected to be better for observed

K_t	p	ρ	$\frac{\rho}{p}$	$\frac{1}{p}$
.001	.0663	.468	7.06	15.08
.002	.0282	.335	11.88	35.46
.003	.0164	.265	16.16	60.98
.004	.0114	.226	19.82	87.72
.005	.0085	.193	22.71	117.65
.006	.0066	.173	26.21	151.52
.007	.0054	.159	29.44	185.19
.008	.0044	.140	31.82	227.27
.009	.0036	.128	35.56	277.78
.010	.0031	.115	37.10	322.58

Table 4.1: The percent of correct and incorrect matches eliminated for various peaking parameter cutoffs with noise. K_t is the the probability threshold used to determine if the matches is eliminated, p is the percentage of incorrect matches eliminated, ρ is the percentage of of correct matches eliminated. $\frac{\rho}{p}$ is the relative frequency of indexing correct and incorrect matches and $\frac{1}{p}$ is the speedup attained if we simply used these techniques to determine matches that are likely to be in actual correspondence in conjunction with an algorithm that hypothesizes matches, such as alignment.

image points than for the random points used in these experiments. This is because model groups that appear in unlikely positions (i.e. such that they are highly foreshortened) are more likely to have one or more points occluded by the object itself, while my experiments assume no self-occlusion. It is therefore expected that groups of observed image points from real objects will produce a higher rate of indexing the correct group than random points.

If we know the prior probability distribution of image group features we can use the probabilities in Bayes' rule. Let b_i denote the bin that corresponds to the image group features and let h be the hypothesis that the model group and the image group are in actual correspondence.

$$P(h \mid b = b_i) = \frac{P(h)P(b = b_i \mid h)}{P(b = b_i)}$$

$P(b = b_i \mid h)$ is given by the peaking effect joint probability histograms and $P(b = b_i)$ is given by the prior probability histogram. I have assumed that the prior probability of each possible match (and thus each possible hypothesis h) is the same, so we can drop the $P(h)$ term without changing the ranking of the hypotheses. Of course, if we had knowledge that models were not equally likely to appear in the image we could use it here.

The joint prior probability histogram of the image group parameters β and $\log \frac{b_1}{b_2}$ for feature points that are the result of model feature points in the database (and not random image points) can be determined by averaging the probabilistic peaking histograms for the set of model groups. For each random model group, we add to the average the joint probability histogram for the correct α shifted on the ratio axis by $\log \frac{a_2}{a_1}$. (A shift is required since the peaking histograms are for $\log \frac{b_1 a_2}{b_2 a_1}$ and we want the probability of $\log \frac{b_1}{b_2}$.) Again, these can be weighted if we know the prior probability of each model group appearing in the image.

This does not account for random extraneous points in the image. We can estimate the distribution of these points by examining the distribution of feature parameters for a large set of randomly selected image points. The prior probability histogram for image parameters for both model points and random points is shown

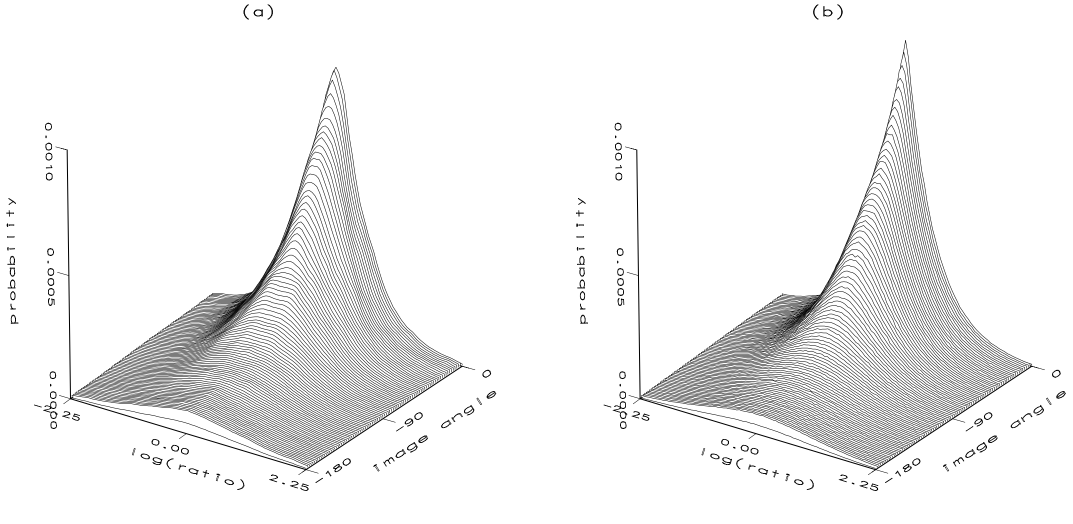

Figure 4.6: The prior probability histograms describing the likelihood of groups of image points falling into each bin. The x-axis is the image angle β. The y-axis is the logarithm of the ratio of distances $\log \frac{b_1 a_2}{b_2 a_1}$. The z-axis is the probability. (a) projected random model points (b) random image points.

K_t	γ	g	$\frac{\gamma}{g}$
.001	31.35	6.08	5.15
.002	40.89	10.41	3.93
.003	48.88	13.86	3.53
.004	54.62	16.80	3.25
.005	60.40	19.32	3.13
.006	65.16	21.84	2.98
.007	68.36	24.08	2.83
.008	72.64	25.81	2.81
.009	75.67	27.72	2.73
.010	81.37	29.86	2.72

Table 4.2: The average posterior probabilities of matching for correct and incorrect groups that are indexed for various probability thresholds. K_t is the probability thresholds, γ is the average posterior probability of correctly indexed matches, g is the average posterior probability of incorrectly indexed matches, and $\frac{\gamma}{g}$ describes their relative size.

in Figure 4.6. Since they are very close, we can use the histogram for the projected model points as the prior probability histogram of the image group parameters for all image points.

I have found that even among groups with high prior probability of matching (those that surpass the threshold, and thus are indexed), matches in actual correspondence have, on average, considerably higher posterior probability. Let's call the expected posterior probability of a correct match that is indexed γ and the expected posterior probability of an incorrect indexed match g. Table 4.2 shows the values these take (neglecting the constant $P(H)$ term both have) for several indexing thresholds along with the ratio $\frac{\gamma}{g}$. Since the matches in actual correspondence have a considerably higher expected posterior probability, we can use the posterior probability to order the matches based on likelihood, if desired.

4.4 Using larger groups

Some algorithms require hypothesized point group matches of more than three points. Probabilistic indexing can be extended to accommodate these algorithms. This has the additional benefit of increasing our ability to discriminate between correct and incorrect matches. To incorporate larger groups into probabilistic indexing, we must be able to index model groups of size k using keys determined from image groups of size k. This is accomplished by examining each subgroup of three points in the k point image groups in the manner of the previous section. If some predetermined constant number x_0 of the subgroups from a model group have high enough probability of matching, then the model group is considered a possible match. To determine if this is the case, we must index the look-up table with each combination of three points in the k point image group and determine which k point model groups are indexed at least x_0 times. The model groups are examined to ensure that each image point corresponds a model point consistently in the matches it was found in. That is, we don't want a particular model point to correspond to one image point when indexed by one subgroup and then correspond to a different image point when indexed by another subgroup and yet still be considered as a possible match.

Call p_0 and ρ_0 the values of p and ρ we had for indexing groups of size three. Table 4.3 shows the new values of p and ρ for larger groups determined experimentally using random model groups and transformations for various values of p_0, ρ_0, k, and x_0. Since, for larger point groups ρ increases or stays about the same and p substantially decreases, increasing the size of groups used increases the ability of probabilistic indexing to discriminate between correct matches and incorrect matches. The price we pay for this accuracy is the speed with which we index groups. While we will index a smaller percentage of the larger groups, grouping processes will typically find more potential groups when the size of the group is increased, and for each group we must now index $\binom{k}{3} = \frac{k!}{3!(k-3)!}$ subgroups of three points. This can be alleviated somewhat by bookkeeping techniques since there are at most $\binom{n}{3}$ total subgroups in the image, where n is the number of feature points in the image.

Clemens and Jacobs [1991] are able to increase the speedup of their system by

	ρ_0	p_0	ρ	p	$\frac{\rho}{p}$
$k = 4$	.292	.0263	.289	.00788	36.68
$x_0 = 2$	.196	.0103	.176	.00172	102.33
	.155	.0059	.127	.00075	169.33
$k = 5$	.313	.0286	.530	.00792	66.92
$x_0 = 3$	.204	.0113	.315	.00124	254.03
	.152	.0065	.206	.00039	528.21
$k = 6$	.311	.0276	.736	.00823	89.43
$x_0 = 4$	.209	.0109	.515	.00114	451.75
	.153	.0062	.348	.00040	870.00

Table 4.3: The percentages of correct and incorrect matches indexed for groups with more than three points: k is the size of the group, x_0 is the required number of indexed subgroups to index the group, ρ_0 is the fraction of correct point subgroups indexed, p_0 is the fraction of incorrect point subgroups indexed, ρ is the fraction of correct groups indexed and p is the fraction of incorrect groups indexed.

increasing the size of the groups and the dimensionality of the index table because they are able to canonically order the points in each group. This means they don't need to test each of the $k!$ orderings of each group. They can canonically order their points since each representation of a model group in the index table is from a single viewpoint. This makes the implicit assumption that localization error will not disturb the image feature points enough to change the canonical ordering. In the general case, each of the $k!$ orderings must be stored in the index table. This method cannot be used with probabilistic indexing because we can't order the image points in a canonical manner (viewing the points from a different direction would generally lead to a different ordering.) This means we would have to examine each of the $k!$ orderings. In addition, we would need to store each of the $\binom{\binom{k}{3}}{x_0}$ combinations of subgroups that would indicate that a model group should be indexed separately in the index table. This extra cost would negate any extra speedup that increasing the dimensionality of our index table could produce.

I argue that using larger groups will not be as beneficial as Clemens and Jacobs

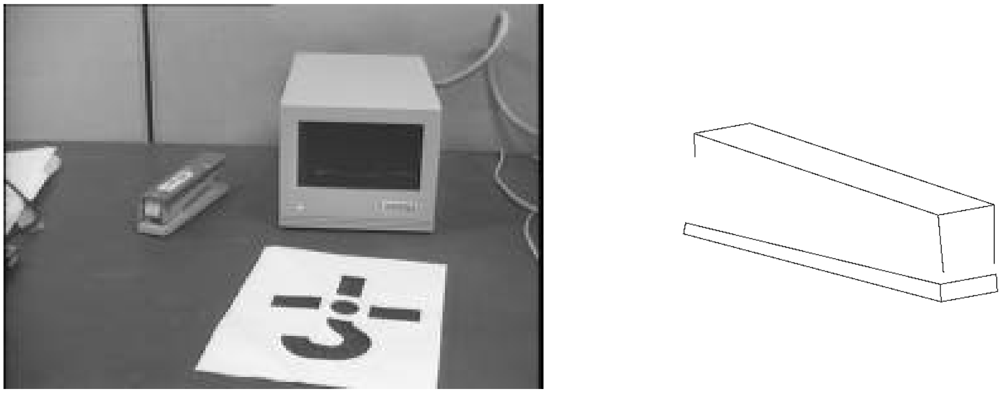

Figure 4.7: Testing probabilistic indexing on real images. (a) An sample image used (b) The stapler model with visible edges drawn in.

claim. The larger the group a grouping process must find, the less likely all of the points in a group will arise from the same object (a point Clemens and Jacobs do not consider.) Any group of points that do not all arise from the same object is useless for indexing. This means that even though the speedup may be increased considerably by examining larger groups, a smaller percentage of the groups that are examined will be useful.

4.5 Results on real images

Probabilistic indexing has been tested on several real images. For these tests, model points on each of the objects were measured by hand. Several images of these objects were captured. Corners were determined with the help of an edge detector. Figure 4.7.a shows an image used in the tests containing a disk, a stapler, and a hand rendering of a symbol from mythology (which I'll call cross.) Figure 4.7.b displays an sample object model (a stapler.) The location of many of the feature points on the model are shown with the visible edges drawn in.

Table 4.4 gives the results of using the feature points from real images to index a database of 6 real objects. The average percentage of correct groups (ρ) and percentage of incorrect groups (p) that were indexed is shown from experiments using 5 images of the stapler, 3 images of the disk, and 3 images of the cross. Also given are the results showing how often random points indexed these model groups.

In each of the cases, the real feature point groups indexed the correct model group with frequency higher than was obtained for models of random points (see Table 1.) The frequency of indexing incorrect model groups was also slightly greater in many cases, except for the stapler where it is substantially greater. The random image points indexed the real model groups with comparable, although higher, frequency than random image points indexed random model groups.

The rendering of the mythological symbol is a two-dimensional model. Figure 4.7.a shows the image of this object that I tested that had the most foreshortening. While the percentage of correct matches indexed was below the average for this instance of this object, over 10% of the correct matches were indexed, even when the probability threshold was high (K_t =.010).

4.6 Discussion

Probabilistic indexing should not be viewed as a method of using randomization in the indexing problem, since the orientations from which each group is correctly indexed are correlated. We rely on the fact that there are so many (approximately $\frac{m^3}{6}$) model groups that all viewing directions will have some model groups that are viewed in a likely orientation. If there is not a wide variety of orientation of the groups themselves, this may not be the case. In the extreme case, flat objects will have only a single orientation that all of the model groups share. Model groups from such objects will not be indexed correctly for many viewing directions, but we can easily determine which objects are flat or nearly flat prior to recognition time. For such objects we can either reduce the probability threshold or use special case techniques for flat [Lamdan *et al.*, 1988] or nearly flat [Arbter *et al.*, 1990] objects.

Contrasting indexing to grouping techniques [Ahuja and Tuceryan, 1989, Huttenlocher and Wayner, 1992, Lowe, 1985, Mohan and Nevatia, 1992] may be useful. Grouping techniques determine sets of image features that are likely to come from the same object. These techniques can also be applied to models to determine which

Object	K_t	ρ	p
Stapler	.002	.337	.0473
Disk		.416	.0277
Cross		.597	.0261
Random		-	.0291
Stapler	.004	.257	.0270
Disk		.323	.0128
Cross		.459	.0124
Random		-	.0144
Stapler	.006	.205	.0181
Disk		.271	.0074
Cross		.382	.0078
Random		-	.0090
Stapler	.008	.182	.0138
Disk		.230	.0053
Cross		.326	.0057
Random		-	.0066
Stapler	.010	.152	.0103
Disk		.204	.0037
Cross		.282	.0043
Random		-	.0053

Table 4.4: The results of experiments on indexing using real objects and images: K_t is the probability threshold, ρ is the percentage of correct matches indexed, and p is the percentage of incorrect matches indexed.

sets of points are likely to be found by grouping in the image. Grouping techniques can thus drastically lower the number of groups in the image and model that must be examined. Rather than examining groups of model points and groups of image points separately as grouping techniques do, indexing systems examine these groups together to determine which matches between them are most likely to be correct. Grouping can be used to predetermine which sets of model and image points are examined by the indexing systems to further reduce the number of matches that must be examined.

Probabilistic indexing can be easily modified to incorporate indexing using information other than geometrical. For example, if the feature points are known to have some property such as color or type, we can store this information with the point and use it to discard matches that are not feasible due to these constraints. Indeed, a model of image illumination should provide adequate information to give probabilistic information regarding the likelihood of matching based on colors assigned to features. The inclusion of this and other probabilistic information and the extension to curved surfaces provides opportunity for further study.

4.7 Summary

In this chapter, I have described an indexing system for use in solving the problem of recognizing three-dimensional objects in single two-dimensional images. The probabilistic peaking effect has been shown to be effective for use in indexing model groups undergoing general rigid transformations in three-dimensions from image group parameters in images generated using the perspective projection. Its use has allowed us to reduce the cardinality of the sets of image and model points necessary in an indexing system, while retaining the indexing speedup. The disadvantage to this system is that not all correct matches between image groups and model groups are indexed. Since a far higher percentage of correct matches than incorrect matches are indexed, probabilistic indexing is usually quite useful. Probabilistic indexing can be used as a pre-processing step for any algorithm that generates matches between groups of image and model points to perform verification on. By selecting only those matches that are likely to produce good results, probabilistic indexing can speed up and improve the performance of such algorithms considerably.

Chapter 5

Fast Alignment

This chapter further explores the concept of fast object recognition through the selective examination of hypotheses. The probabilistic indexing techniques described in the previous chapter and a set of error criteria are used to select likely hypotheses for the alignment method. These techniques result in a considerable speedup without greatly increasing the chance of missing an object.

5.1 Introduction

The alignment method [Huttenlocher and Ullman, 1990] is a model-based object recognition technique for recognizing three-dimensional objects from a single view in two-dimensional images. For each model in the database, triples of image points are matched with triples of model points. For each match, the weak-perspective transformation that brings them into alignment is determined. The remaining model points (and/or other model features) are then transformed by this transformation and compared against the remaining image points (and/or other image features) to verify the correctness of the transformation.

If every match of three model points to three image points is considered, the alignment method requires $O(m^4 n^3 \log n)$ time due to a $O(m \log n)$ verification step, where m is the number of model features and n is the number of image features. Huttenlocher and Ullman propose techniques that in some cases can reduce this to

$O(m^3 n^2 \log n)$. Unless these time bounds have a very small leading constant, the running time will be considerable, since a model with 20 features and an image with 50 features is a relatively simple problem.

If a model object is present in the image, it is likely that a substantial number triples of model points can be detected. Ideally, only one of these triples needs to be found to recognize the object. If all image triples are examined, then much extra work is being done that is not necessary. Even if we can stop once a close enough match has been found, ordering the matches based on some likelihood of a good match can reduce the number of matches examined considerably.

This chapter examines techniques to eliminate hypotheses from consideration in the alignment method. The techniques that are used to eliminate matches are based on the following two principles:

1. The probability density functions of angles and distance ratios in images peak strongly at the pre-projection (model) value [Ben-Arie, 1990, Binford *et al.*, 1989, Burns *et al.*, 1990]. The previous chapter showed how a probabilistic indexing system could be built using this effect.

2. Matches that produce a transformation with a large uncertainty are unlikely to result in a good correspondence between the model and the image and are less likely to result in the verification routine determining that the object is present in the image. Examples of the use of similar principles can be found in Mundy *et al.* [1988] and Costa *et al.* [1989].

In the next section, I'll discuss the alignment method in more detail and summarize the algorithm. Section 5.3 will discuss how probabilistic indexing is used to eliminate many matches between groups of image points and model points. Sections 5.4-5.6 describe how we can determine which matches result in transformations with large uncertainties. Section 5.7 will present experimental results including results on the efficacy of the affine approximation to the perspective projection. Section 5.8 discusses the techniques and results and gives an analysis of the speedup produced under various conditions. Finally, Section 5.9 summarizes the results.

5.2 The alignment method

The alignment method [Huttenlocher and Ullman, 1987, 1988, 1990] is a model-based technique for recognizing rigid three-dimensional objects from a monocular two-dimensional image. The premise of the alignment method is that a unique (up to a reflection) affine transformation between the model and image of the model can be found by matching three model points with three image points. This transformation is given by $p_i = \Pi(sR(p_m - p_{m0})) + p_{i0}$. Here, p_m is a model point and p_i is its corresponding image point, p_{m0} and p_{i0} are relative offsets, s is a scale factor, R is a three-dimensional rotation and $\Pi(\cdot)$ is the orthographic projection.

Once again, let's call the set of model points being matched (three points per group for alignment) the *model group* and the image points hypothetically matched to them the *image group*. If each of the points in the image group is a result of the projection of its corresponding model group point then the two groups are in *actual correspondence*. For the rest of this chapter, I will consider a single object model. In practice, each model must be examined separately in the alignment method.

If the model is present in the image and we consider all possible matches between three model points and three image points it is very likely that many matches will be in actual correspondence. The transformations determined from the matches in actual correspondence should then be close to the correct transformation. Each transformation computed must be verified against the image to determine if it is valid.

It is not advisable to examine each combination of three image points and three model points. If there are m model points and n image points, the entire algorithm would require $O(m^4 n^3 \log n)$ operations, since the verification routine requires $O(m \log n)$ operations. Huttenlocher and Ullman have proposed various techniques to lower the complexity of the algorithm. In Huttenlocher and Ullman [1990], virtual points found by using the orientations at two model and image points are used to reduce the complexity to $O(m^3 n^2 \log n)$ operations.

The algorithm can be summarized as follows:

1. For each group of model points, rotate and translate the group such that the first point lies at the origin and the other points lie in the $x - y$ plane. This

step is performed off-line for each model group and the new coordinates for the second and third points, b'_m and c'_m, are stored, as is a_m, which is the relative offset p_{m0} in the transformation described above.

2. For each possible match between and image group (a_i, b_i, c_i) and a model group (a_m, b'_m, c'_m), solve for the 2×2 linear transformation matrix L, that brings the points into alignment in two dimensions, given by the following system of equations:

$$Lb'_m = b'_i \qquad\qquad Lc'_m = c'_i$$

where

$$b'_i = b_i - a_i \qquad\qquad c'_i = c_i - a_i$$

3. Determine the 3x3 linear transformation matrix sR^+ (due to the reflective ambiguity sR^- also exists and is described below) that maps the model onto the image, given by:

$$sR^+ = \begin{bmatrix} l_{11} & l_{12} & (c_2 l_{21} - c_1 l_{22})/s \\ l_{21} & l_{22} & (c_1 l_{12} - c_2 l_{11})/s \\ c_1 & c_2 & (l_{11} l_{22} - l_{21} l_{12})/s \end{bmatrix}$$

where

$$w = l_{12}^2 + l_{22}^2 - l_{11}^2 - l_{21}^2 \qquad\qquad q = l_{11} l_{12} + l_{21} l_{22}$$

$$c_1 = \sqrt{\tfrac{1}{2}(w + \sqrt{w^2 + 4q^2})} \qquad\qquad c_2 = -q/c_1$$

$$s = \sqrt{l_{11}^2 + l_{21}^2 + c_1^2}$$

4. Perform a verification procedure to determine if the transformation is correct. Verification has two steps. The first examines whether the remaining model points are transformed close to corresponding image points. The transformed model points can be found by $p_i = \Pi(sR^+(p_m - a_m)) + a_i$ where p_m is the model point, p_i is its transformed location in the image, and sR^+, a_m, and a_i are as given above. Since there is a reflective ambiguity, two transformations must be examined. The second, sR^-, is the same as sR^+, except that the terms at positions $(1,3)$, $(2,3)$, $(3,1)$, and $(3,2)$ in the matrix are negated. If enough model points are transformed close to image points, then a second verification step is performed that examines whether edges are transformed close to image edges.

Note that the affine transformation is used in this algorithm as an approximation to the full perspective projection, and is valid only when the distance to the object is large compared to the size of the object in the z direction (after transformation) [Thompson and Mundy, 1987]. Experiments determining when the approximation is valid in practice are presented in Section 5.7.

5.3 Probabilistic indexing

We saw in the previous chapter that, while it has been proven that there is no invariant for three-dimensional point sets seen from a single view [Burns *et al.*, 1990, Clemens and Jacobs, 1991], it has been observed that there is a strong peaking effect for many angles and ratios of lengths in images at the values taken by the features in the model [Ben-Arie, 1990, Binford *et al.*, 1989, Burns *et al.*, 1990]. This information was used to create a system capable of indexing likely matches using sets of three model and image points.

Ben-Arie presents two recognition schemes using the probabilistic peaking effect. The first uses an A^* search technique [Hart *et al.*, 1968] as in Ben-Arie and Meiri [1987] but uses as a cost function the result of a stochastic labeling algorithm based on relaxation [Rosenfeld *et al.*, 1976] that uses the probabilistic peaking effect. For

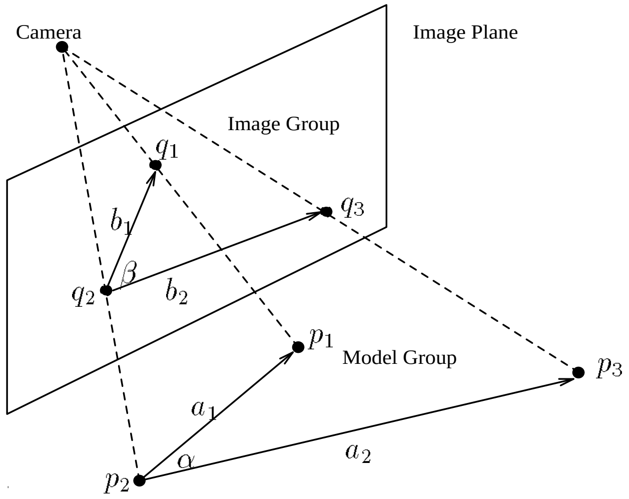

Figure 5.1: A model group projected onto the image plane.

objects with 12 features the stochastic labeling algorithm labeled 91% of the features correctly, with worse results for larger objects. No extraneous features were considered in these experiments. The second algorithm matches angular primitives based on which has the largest bayesian probability of matching. The angular primitives of the algorithm consist of an angle and the ratio of lengths of two arms. Three points are necessary to determine this information. So, unless other information is used, $O(m^3)$ model features and $O(n^3)$ image features exist. Note that any additional information used would be of equal value in the alignment method. In addition, each feature is labeled correctly only 53% of the time. The experiments on synthetic data were performed by projecting the model points using the orthographic projection and without noise or extraneous points.

I use probabilistic indexing to eliminate hypotheses in the alignment method based on their likelihood of being correct. I'll briefly review how the probabilistic index system works. Let's again refer to a generic model group projected onto the image plane (Figure 5.1.) The probabilistic indexing system creates an index table by discretizing the α-$\frac{a_1}{a_2}$ space and placing each model group into the cell of the table corresponding to its parameters. Let b_i denote the bin that corresponds to the image group features and let h be the hypothesis that the model group and the image group are in actual correspondence. We can determine the probability of h being correct by applying Bayes' rule:

$$P(h \mid b = b_i) = \frac{P(h)P(b = b_i \mid h)}{P(b = b_i)}$$

$P(b = b_i \mid h)$ is given by the peaking effect joint probability histograms and $P(b = b_i)$ is given by the prior probability histogram. $P(h)$ is assumed to be the same for each hypothesis, so it is ignored.

When presented with an image group, the system determines β and $\frac{b_1}{b_2}$ and examines the probability histograms to determine the index table cells that contain model groups that have a large enough probability of generating the image features. The model groups at those locations are then considered as possible matches for the image group. So, for a specific image group k, a model group is eliminated if:

$$\frac{P(b = b_i^{(k)} \mid h)}{P(b = b_i^{(k)})} < \gamma_p$$

where $b_i^{(k)}$ is the bin for the kth image group and γ_p is an empirically determined constant.

5.4 Eliminating matches using the condition number

Section 5.2 gave the solution for the two-dimensional affine transformation as follows:

$$L = \left[\begin{array}{cc} l_{11} & l_{12} \\ l_{21} & l_{22} \end{array} \right]$$

$$Lb'_m = b'_i \qquad\qquad Lc'_m = c'_i$$

These equations can be transformed into:

$$ML_1 = I_1 \qquad\qquad ML_2 = I_2$$

where

$$M = \begin{bmatrix} b'_{m_x} & b'_{m_y} \\ c'_{m_x} & c'_{m_y} \end{bmatrix}$$

$$L_1 = \begin{bmatrix} l_{11} \\ l_{12} \end{bmatrix} \qquad\qquad L_2 = \begin{bmatrix} l_{21} \\ l_{22} \end{bmatrix}$$

$$I_1 = \begin{bmatrix} b'_{i_x} \\ c'_{i_x} \end{bmatrix} \qquad\qquad I_2 = \begin{bmatrix} b'_{i_y} \\ c'_{i_y} \end{bmatrix}$$

Our localization of the image points will have some error, of course. Therefore, our solutions for L_1 and L_2 will also have some error. Let $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{L}_1$, and $\mathcal{L}_2$ denote the correct values of I_1, I_2, L_1, and L_2, and let δI_1, δI_2, δL_1, and δL_2 denote their errors such that:

$$\mathcal{I}_1 = I_1 + \delta I_1 \qquad\qquad \mathcal{I}_2 = I_2 + \delta I_2$$

$$\mathcal{L}_1 = L_1 + \delta L_1 \qquad\qquad \mathcal{L}_2 = L_2 + \delta L_2$$

From basic matrix computations [Watkins, 1991] we can bound the error on δL_1 and δL_2 as follows:

$$\tfrac{||\delta L_1||}{||L_1||} \leq \kappa(M)\tfrac{||\delta I_1||}{||I_1||} \qquad\qquad \tfrac{||\delta L_2||}{||L_2||} \leq \kappa(M)\tfrac{||\delta I_2||}{||I_2||}$$

where $||\cdot||$ is any vector norm (and its induced matrix norm) and $\kappa(M) = ||M||\cdot||M^{-1}||$ is the condition number of M.

So, if M has a large condition number, we may have large errors δL_1 and δL_2. The bounds given are tight. For some vectors δI_1 and δI_2 we have equality. For others, these bounds are far less accurate. Still, it is more likely that a model group will result in accurate values of L_1 and L_2 if the group has a small condition number, as the experiments I describe in Section 5.7 demonstrate.

5.5 Eliminating matches using the norm of image points

From the error bounds in the previous section, we see that if $\frac{||\delta I_1||}{||I_1||}$ or $\frac{||\delta I_2||}{||I_2||}$ is large, we may have a large relative error in L_1 or L_2. Specifically, small values of $||I_1||$ and $||I_2||$ may produce large errors. Actually, neither of these alone is enough to produce large effect, as I will show here. The equations to solve for L_1 and L_2 from Section 5.4 may be transformed into:

$$\begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

Small values of $||I_1||$ or $||I_2||$ can result in large relative error of L_1 or L_2, but this is partially a consequence of $||L_1||$ or $||L_2||$ being small. If one of these is small compared to the other (e.g. $||L_1|| << ||L_2||$) a large relative error in the small one will not be significant in the computation of $sR^+(p_m - p_{m0})$. So, we see it is only when both $||I_1||$ and $||I_2||$ are both small that we may experience problems. Using this analysis we can eliminate an image group k if:

$$\left|\left| \begin{bmatrix} I_{1_k} \\ I_{2_k} \end{bmatrix} \right|\right|_2 < K \max_i \left|\left| \begin{bmatrix} I_{1_i} \\ I_{2_i} \end{bmatrix} \right|\right|_2$$

where I_{1_i} refers to the image values for the ith image group and $K \leq 1.0$ is an empirically determined constant.

Note that eliminating matches using this technique may be counterproductive if individual objects occupy a small local portion of the image, since the points in the correct hypotheses will be close together with respect overall size of the image and thus will have small norms as calculated above.

5.6 Eliminating matches using the model group area

Even when the model group produces a condition number of one (the best case), it is possible that the group leads to a large error in the computation of the transformation matrix, as can be seen from the following analysis, similar to that done for the affine coordinates of Lamdan *et al.* [1988] in Costa *et al.* [1989]. Assuming non-singularity we have:

$$\begin{bmatrix} l_{11} \\ l_{12} \end{bmatrix} = \begin{bmatrix} b'_{m_x} & b'_{m_y} \\ c'_{m_x} & c'_{m_y} \end{bmatrix}^{-1} \begin{bmatrix} b'_{i_x} \\ c'_{i_x} \end{bmatrix}$$

Computing this inverse we get:

$$l_{11} = \frac{c'_{m_y} b'_{i_x} - b'_{m_y} c'_{i_x}}{b'_{m_x} c'_{m_y} - b'_{m_y} c'_{m_x}}$$

Substituting the original model and image values, this becomes:

$$l_{11} = \frac{(c_{m_y} - a_{m_y})(b_{i_x} - a_{i_x}) - (b_{m_y} - a_{m_y})(c_{i_x} - a_{i_x})}{(b_{m_x} - a_{m_x})(c_{m_y} - a_{m_y}) - (b_{m_y} - a_{m_y})(c_{m_x} - a_{m_x})}$$

Similarly, l_{12}, l_{21}, and l_{22} are:

$$l_{12} = \frac{(c_{m_x} - a_{m_x})(b_{i_x} - a_{i_x}) - (b_{m_x} - a_{m_x})(c_{i_x} - a_{i_x})}{(b_{m_x} - a_{m_x})(c_{m_y} - a_{m_y}) - (b_{m_y} - a_{m_y})(c_{m_x} - a_{m_x})}$$

$$l_{21} = \frac{(c_{m_y} - a_{m_y})(b_{i_y} - a_{i_y}) - (b_{m_y} - a_{m_y})(c_{i_y} - a_{i_y})}{(b_{m_x} - a_{m_x})(c_{m_y} - a_{m_y}) - (b_{m_y} - a_{m_y})(c_{m_x} - a_{m_x})}$$

$$l_{22} = \frac{(c_{m_x} - a_{m_x})(b_{i_y} - a_{i_y}) - (b_{m_x} - a_{m_x})(c_{i_y} - a_{i_y})}{(b_{m_x} - a_{m_x})(c_{m_y} - a_{m_y}) - (b_{m_y} - a_{m_y})(c_{m_x} - a_{m_x})}$$

The area of the triangle formed by a_m, b_m, and c_m is given [Thomas and Finney, 1984] by

$$\text{area}(\triangle a_m b_m c_m) = \pm \frac{1}{2} \begin{vmatrix} a_{m_x} & a_{m_y} & 1 \\ b_{m_x} & b_{m_y} & 1 \\ c_{m_x} & c_{m_y} & 1 \end{vmatrix}$$

which is half the absolute value of the denominator of each of the above equations. Interestingly, the size of the numerator is twice the area of a triangle in a plane where one axis is in model coordinates and one axis is in image coordinates. For l_{11}, the triangle points for the numerator are:

$$a_t = \begin{bmatrix} a_{i_x} \\ a_{m_y} \end{bmatrix} \quad b_t = \begin{bmatrix} b_{i_x} \\ b_{m_y} \end{bmatrix} \quad c_t = \begin{bmatrix} c_{i_x} \\ c_{m_y} \end{bmatrix}$$

$$l_{11} = \pm \frac{\mathrm{area}(\triangle a_t b_t c_t)}{\mathrm{area}(\triangle a_m b_m c_m)}$$

If the denominator area is small, this will have the effect of magnifying the errors present in the numerator and the denominator. So, we can eliminate a model group k if it satisfies

$$\mathrm{area}(\triangle a_{m_k} b_{m_k} c_{m_k}) < K \max_i \mathrm{area}(\triangle a_{m_i} b_{m_i} c_{m_i})$$

where a_{m_i} refers to the first point of the ith model group and $K \leq 1.0$ is an empirically determined constant.

5.7 Experimental results

These techniques have been tested on both randomly generated and real data. This section presents those experiments and results.

5.7.1 Synthetic data

To determine the efficacy of the affine transformation as an approximation to the perspective transformation and to determine the best parameters for eliminating unlikely matches, tests were carried out on random point sets. All of the experiments were conducted on sets of ten points with x, y, and z coordinates randomly distributed between -100 and 100. For each point set a random transformation was generated as follows:

1. Three random angles, uniformly distributed between 0 and 360 were generated and the points were rotated about the x, y and z axes by these angles.

2. Three random displacements, uniformly distributed between -100 and 100 were generated, and the points were displaced by these values.

3. The points were projected by the full perspective projection onto the image plane using subpixel accuracy. Various object distances were used to test the efficacy of the affine approximation to the perspective projection. In these experiments the focal length was varied with the distance to keep the size of the images approximately the same.

4. Gaussian noise was added to the each image point's x and y coordinates. Various standard deviations were used in the tests of the affine approximation to the perspective project. A standard deviation of $\sigma = 2.0$ was used for the remaining tests.

Each experiment counted the number of transformations for which the closest transformed model point to each image point is the correct match. Thus, a transformation was only considered successful if each corresponding model and image point were brought close together. For these experiments each model point was projected onto the image and no extraneous image points were included. Since we examine every possible group of three image points, and each has a model group in actual correspondence with it, there are

$$\sum_{i=1}^{10} \sum_{j=i+1}^{10} \sum_{k=j+1}^{10} 1 = 120$$

actual correspondences examined for each random object. Each experiment was performed on 100 random objects.

Figure 5.2 shows the fraction of actual correspondences which produced successful transformations using objects at varying ratios of object distance to object depth. The three plots are for images with no noise, images with gaussian noise of standard deviation 2.0 added to the image feature coordinates, and images with gaussian noise of standard deviation 5.0 added to the image feature coordinates. The x-axis is the ratio of object distance to the thickness of the object. We see that for each of the three plots serious degradation of the fraction correct begins at approximately a ratio of 8, which I conclude is the maximum ratio for which the affine approximation to the perspective projection is accurate for use with the alignment method. Since there are usually many correct image groups, the alignment method can find correct transformations even if this ratio is less than 8.

For the remainder of the experiments, I use gaussian noise with standard deviation 2.0, and objects at a distance such that the ratio of object distance to object depth is 10. Figure 5.3 shows the fraction of total matches examined (not eliminated) and the fraction of correct matches examined for various values of each of the elimination parameters. The most powerful parameter is the peaking probability, as seen by the large distance between the plots. The condition number and model group area parameters appear to be useful for elimination as well, while the image group norm appears to be of little use.

Figure 5.4 displays the ratio of incorrect matches examined to correct matches examined for various values of each of the elimination parameters. The total remaining ratio and the marginal change from the previous point are shown. Each of the parameters appears to have a graceful degradation, except for the norm of image points (Figure 5.4.d), which is also the only parameter where the marginal ratio falls below the remaining ratio (indicating that the total remaining ratio rose for that data point.)

Using all of the above parameters at reasonable values (peaking probability: 0.08, condition number: 6.0, fraction of max area: 0.3, fraction of max norm: 0.3), 3.1% of all possible matches were examined while 17.9% of correct matches were examined. So, the algorithm examined less than 1/30 of the total matches, yet still examined more than enough correct matches to produce a correct transformation.

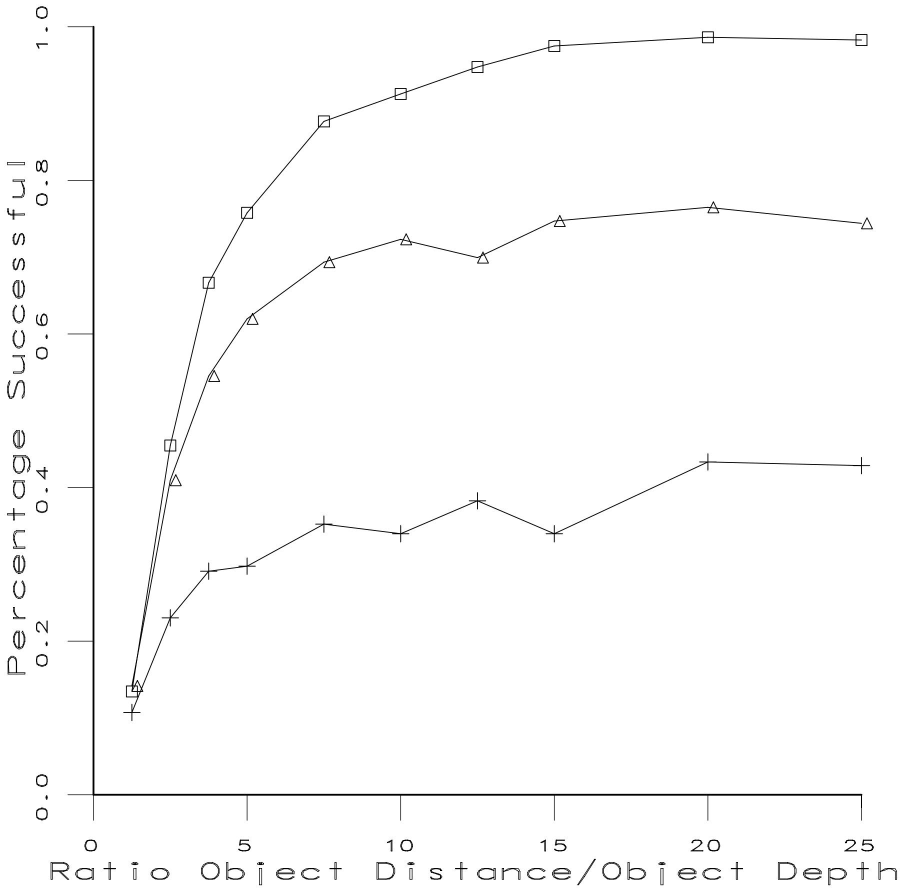

Figure 5.2: Percentage of correct hypotheses resulting in successful transformations at various object distances. □: no noise, △: noise ($\sigma = 2.0$), +: noise ($\sigma = 5.0$)

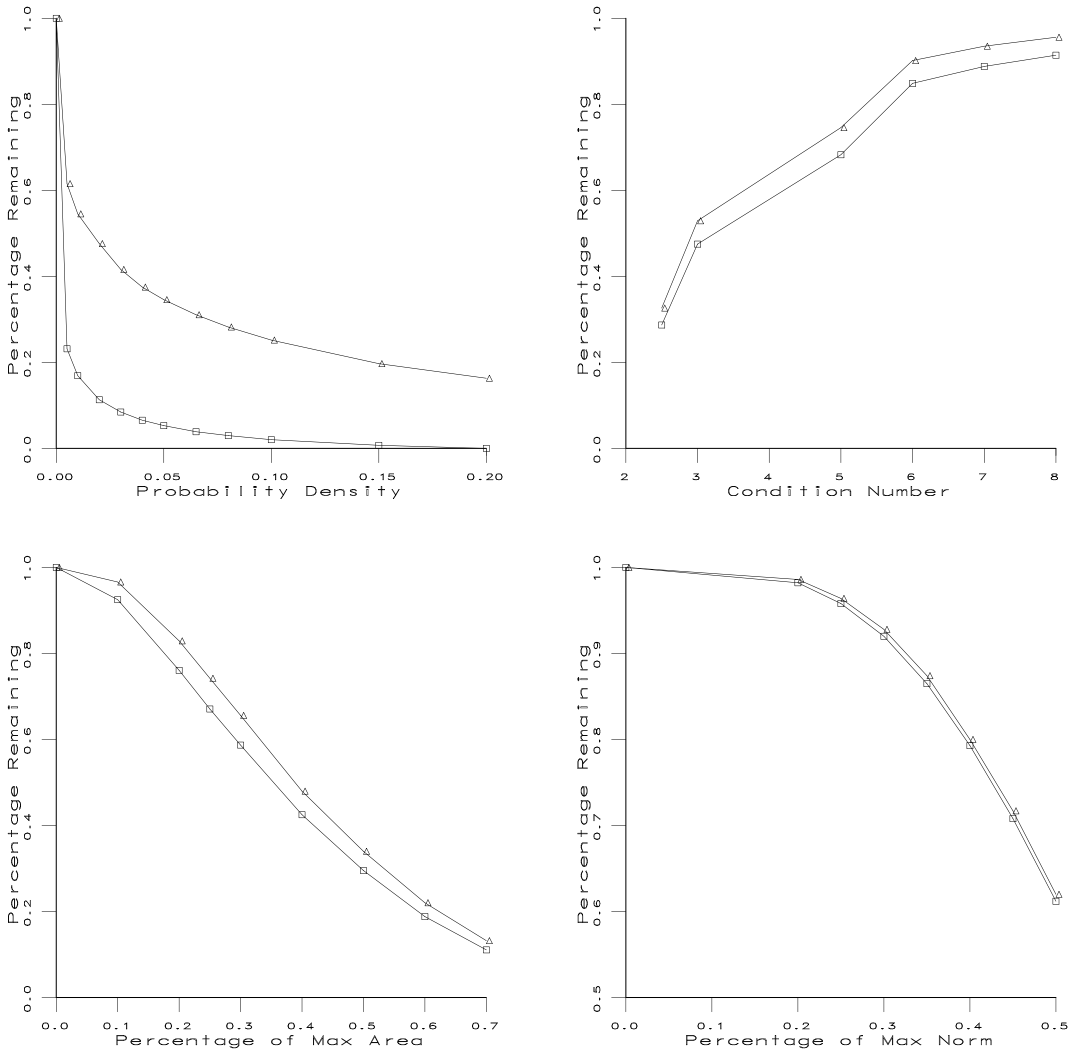

Figure 5.3: Percentage of total and correct matches examined for various values of each elimination parameter. □: total, △: correct.

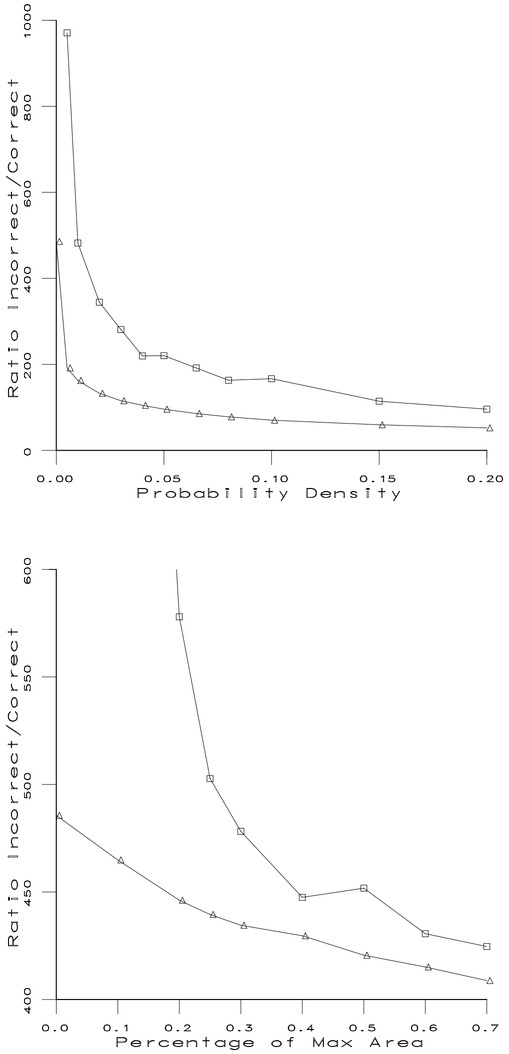

Figure 5.4: Marginal and remaining ratios of incorrect matches to correct matches for various values of each elimination parameter. □: marginal, △: remaining.

5.7.2 Real images

The techniques were tested on real images (see Figure 5.5 for an example.) The procedure to recognize the model was as follows:

1. Selected model points were measured by hand.

2. The edges were found using a Canny [1986] edge detector.

3. Corners were selected by hand.

4. For each group not eliminated, the alignment transformation was found.

5. A quick verification routine determined if the rest of model points were transformed close to image points.

6. If the quick verification routine scored high enough, a more detailed verification routine was executed. This routine determined if edges present in the model were also present in the image.

7. For the best scoring transformation, additional matches were determined and the least-squared error affine transformation was determined using a method similar to that of Stein and Medioni [1992].

Figure 5.5 (top left) shows an image of a stapler that was used to test the elimination techniques. Figure 5.5 (top right) shows the Canny edges of this image, and 5.5 (bottom left) shows the model points as transformed by the highest scoring transformation with some edges drawn in. Figure 5.5 (bottom right) is the transformed model overlaid on the Canny edges. For this image, the algorithm using elimination examined 1.8% of the total matches and 37.0% of the correct matches (compared with 3.1% and 17.9% for the random points.) The explanation for the better performance for the real image points sets lies in the fact that the model groups that would appear in unlikely positions (i.e. such that they are highly foreshortened) very often have one or more points occluded by the object itself, while our experiments on random point sets assume no self-occlusion.

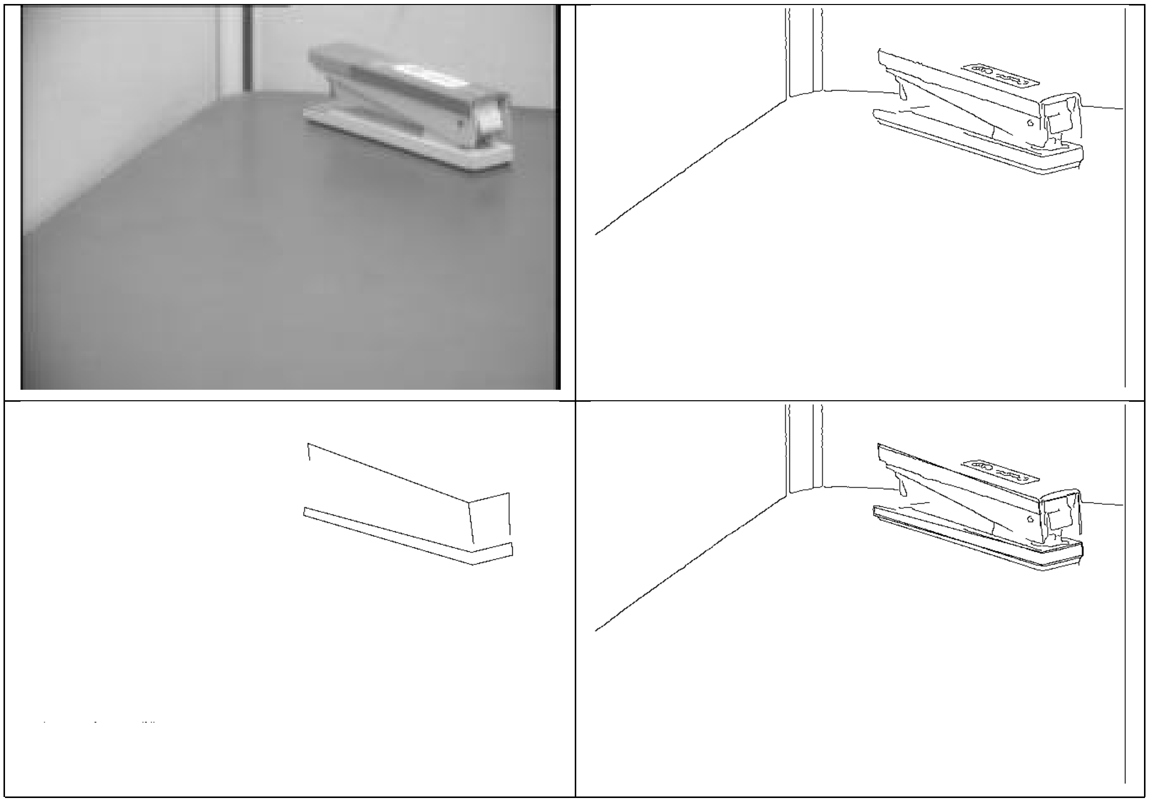

Figure 5.5: Testing fast alignment techniques. (Top left) Input image, (Top right) Canny edges, (Bottom left) Recognized model with edges drawn in, (Bottom right) Recognized model overlaid on Canny edges

5.8 Discussion

I examine the speedup and probability of a false negative produced by these techniques under three recognition models:

1. Each possible match between an image group and a model group that is not eliminated receives a score from some verification process. The best scoring match is accepted as correct if the score meets some criterion.

2. Matches are examined in some order. Matches that are not eliminated receive a score from the verification process. As soon as the score for one of the matches meets some criterion, it is accepted as correct and the remainder of the matches are not examined.

3. Matches are examined in some order. Only enough matches (that are not eliminated) are examined until the probability of having missed an object is arbitrarily low using the randomization technique described in Chapter 3.

The speedup will be defined as the expected number matches that must be verified by algorithm without using the elimination techniques divided by the expected number when using the techniques. I do not consider the overhead necessary to determine if a match is eliminated in the speedup since this process is $O(1)$ per match and the verification step is $O(m \log n)$. Let h be the total number of matches examined, p be the fraction of total matches not eliminated, η be the total number of matches examined that produce a correct transformation, and ρ be the fraction of these matches not eliminated.

In the first model, we examine hp matches when using these techniques and h matches when not using these techniques, so the speedup is simply $\frac{1}{p}$.

In the second model, if we assume that there is an equal chance of each possible match being chosen, we have a hypergeometric distribution. For large values of h, this can be approximated by the binomial distribution. The expected number of matches that must be verified when not using the elimination techniques is then approximately $\frac{h}{\eta}$. When using the elimination techniques the expected number of matches that must be verified is approximately $\frac{ph}{\rho\eta}$. The expected speedup is thus $\frac{\rho}{p}$. This analysis assumes that $\eta > 0$ (that a correct match exists to be found.) If $\eta = 0$ either because the model is not present in the image or because none of the matches in actual correspondence produces a good enough transformation, then the speedup is the same as for model 1 ($\frac{1}{p}$).

When randomization is used (the third model) the number of hypotheses that must be examined to achieve probability $1 - \delta$ of finding a correct match is $\frac{n^3}{(fm)^3} \log \frac{1}{\delta}$ where f is the fraction of model points appearing in the image. When probabilistic indexing is used the probability that each examined match is correct increases by a factor of $\frac{\rho}{p}$. When this is used to limit the bound on the number of matches we must examined we obtain a speedup of $\frac{\rho}{p}$.

Table 5.1 shows the expected speedup for some values of the elimination parameters. Impressive speedups are attained for recognition model 1 and for model 2 when $\eta = 0$. The speedups for recognition model 2 when $\eta > 0$ and model 3 are more modest.

Assuming that the probability of a correct match being eliminated is independent of whether other correct matches have been eliminated (this assumption will be discussed below), the probability of a false negative as a result of eliminating correct matches is $(1 - \rho)^\eta$ for the first two models. In the third model, the probability of a false negative remains δ (assuming independence,) so these techniques should not adversely affect this case.

Table 5.2 shows the probability of a false negative resulting from eliminated matches for the values of ρ in Table 5.1 and various values of η. As might be expected, for small values of η there is a non-negligible probability of our techniques causing a false negative. But, it is expected that a large number of correct matches will be found since, if even 10 points from a model are present in an image, there are 120 model groups. Grouping algorithms should be able to find a significant number of these model groups. For reasonable values of η, the probability of a false negative from eliminating correct matches becomes negligible.

peaking prob.	condition number	model area	image norm	ρ	p	$\frac{1}{p}$	$\frac{\rho}{p}$
0.20	4	0.5	0.5	.054	.006	179.81	9.71
0.15	5	0.4	0.4	.092	.012	83.47	7.70
0.10	6	0.3	0.3	.118	.018	55.24	6.52
0.05	7	0.2	0.2	.279	.058	17.17	4.79

Table 5.1: The speedups for various elimination parameters

ρ	η =50	η =75	η =100
.054	.062	.016	3.88x10^{-3}
.092	8.02x10^{-3}	7.18x10^{-4}	6.44x10^{-5}
.118	1.88x10^{-3}	8.13x10^{-5}	3.52x10^{-6}
.279	7.88x10^{-8}	2.21x10^{-11}	6.22x10^{-15}

Table 5.2: The probability of a false negative for various values of ρ (the fraction of correct matches indexed) and η (the number of correct model groups appearing in the image.)

Let's now consider the question of the independence of the probability of correct matches being eliminated. More specifically, we want to know if it is possible for some object to be in an orientation for which all model groups appear in unlikely configurations in an image. A model group appears in an unlikely configuration when it is viewed from a position with some proximity to the plane in which the points in the group lie. Objects that are not nearly flat cannot have the viewing direction nearly coplanar with each model group, so these objects are not expected to be a problem. A nearly flat object rotated such that it is very foreshortened in the image may produce angles and/or distance ratios far from the probability peaks. Such images would not benefit much from the elimination of matches based on the peaking parameter, but problems with such images are common to most object recognition systems including the human visual system. Two methods that could help alleviate this problem are to continue relaxing the parameters until most matches have been examined or using special case techniques for recognizing flat [Lamdan *et al.*, 1988] or nearly-flat objects [Arbter *et al.*, 1990].

5.9 Summary

This chapter has presented techniques that greatly reduce the number of matches that must be examined in the alignment method. The probabilistic peaking effect and error criteria have been used to eliminate unlikely hypotheses, greatly increasing the speed at which objects can be recognized. Experimental results were given that showed that these techniques work in practice and still result in a correct transformation being found.

Chapter 6

A Connectionist Approach to Model-Based Object Recognition

This chapter discusses how model-based object recognition techniques can be implemented in a connectionist fashion to run extremely quickly, assuming a sufficient number of simple processing elements are available.

6.1 Introduction

Connectionist algorithms have been an interesting area of study due to the massive parallelism possible and the similarity to biological processing of information. Such biological systems provide evidence that such a system is capable of extremely powerful computation. Connectionism has been applied to many interesting problems. This chapter presents an approach to recognize objects in the model-based paradigm.

Techniques using groups of image points to index groups of model points that may match them [Lamdan *et al.*, 1990, Clemens and Jacobs, 1991, Jacobs, 1992, Weinshall, 1993, Olson, 1993b] are promising for object recognition because of their ability to eliminate many point groups from consideration without expending much computation. This work will exploit another beneficial property of these indexing systems. Indexing systems have inherently parallel structure and extremely fast connectionist implementations are possible. In fact, in a connectionist implementation, eliminating groups from consideration is no longer necessary for quick recognition, since this work on separate groups is done in parallel. Matches are simply given varying levels of likelihood, which are then used to determine which objects are present in the image.

This approach assumes that feature points have already been extracted from the image, and it uses feature points as the primary tool for recognition. Determining feature points in a connectionist framework is not difficult, but is outside the scope of the work. The use of feature points as the primary means of recognition is somewhat limiting on the performance of a recognition system (see Chapter 3.) A conservative recognition system will find false positives in complex images, but active verification techniques can be used to discard incorrect hypotheses. In addition, these techniques can be extended to features other than points.

Each indexing system requires some fixed size of point groups to generate keys for indexing. I will call this size k. For all of the systems examined in this work $k \leq 5$, so I will treat k as a constant when possible. I will show that as much accuracy as desired (up to the limits of using feature points to generate hypotheses) can be achieved using $O(lmn^k)$ processing elements, where l is the number of objects the system recognizes, m is the number of feature points per model, and n is the number of image points. Alternately, we can use $O((l+m+n)n^{k-1}+I)$ elements, where I is the number of elements necessary to cover the indexing space at the level of fineness required to obtain accurate indexing.

The running time of the system is $O(\log n)$ if we use fixed fan-in and fan-out processing elements. If processing elements are allowed to broadcast a value on $O(n)$ dedicated connections in $O(1)$ time and sum $O(n)$ inputs on dedicated connections in $O(1)$ time then the running time is $O(1)$.

Rigoutsos and Hummel [1992] have previously described a parallel implementation of the geometric hashing system [Lamdan *et al.*, 1988]. While the connectionist version of their algorithm shares some similarities to the work described here, their work concentrated on general-purpose parallel computers. Such computers are typically limited in their scalability and their speed, two qualities that are key to the implementation of a real-time object recognition system. Special-purpose parallel computers can avoid these limitations, since they can be specifically tailored for the relevant application. In this work, I consider special-purpose connectionist units exclusively. The work of Rigoutsos and Hummel was also specific to the geometric hashing system, while this work treats a general class of algorithms of which geometric hashing is a member.

Next, I'll give an overview of connectionist algorithms. Then, I'll describe how indexing system are used for object recognition. The connectionist implementation of these concepts is then discussed.

6.2 Connectionist processing

Connectionist systems typically use a huge number of simple processing elements operating in parallel. I will use unit and processing element interchangeably to describe this basic processor. These processing elements communicate on fixed connections forming a graph structure, where the processing elements are viewed as vertices of the graph and the communication connections are directed edges. Often in a feedforward network, the processing elements are arranged in layers such that each processing element receives input only from processing elements in the previous layer and send output only to processing elements in the following layer (see, for example, Figure 6.1.)

The processing elements that I will use are defined by the following values:

i : a vector of inputs from other units
$P(i)$: a function on the inputs determining the excitation of the unit

Each unit receives an input value from each of the units specified by the input vector. The excitation of the unit is then a function of these input values. Other units receive this value as input according to their input vectors. See Figure 6.2.

Some researchers in this area have used considerably different units. For example, in Feldman's [1985] framework, the inputs and output are time varying signals. I consider the case of a single static input image, so the inputs and outputs take on a

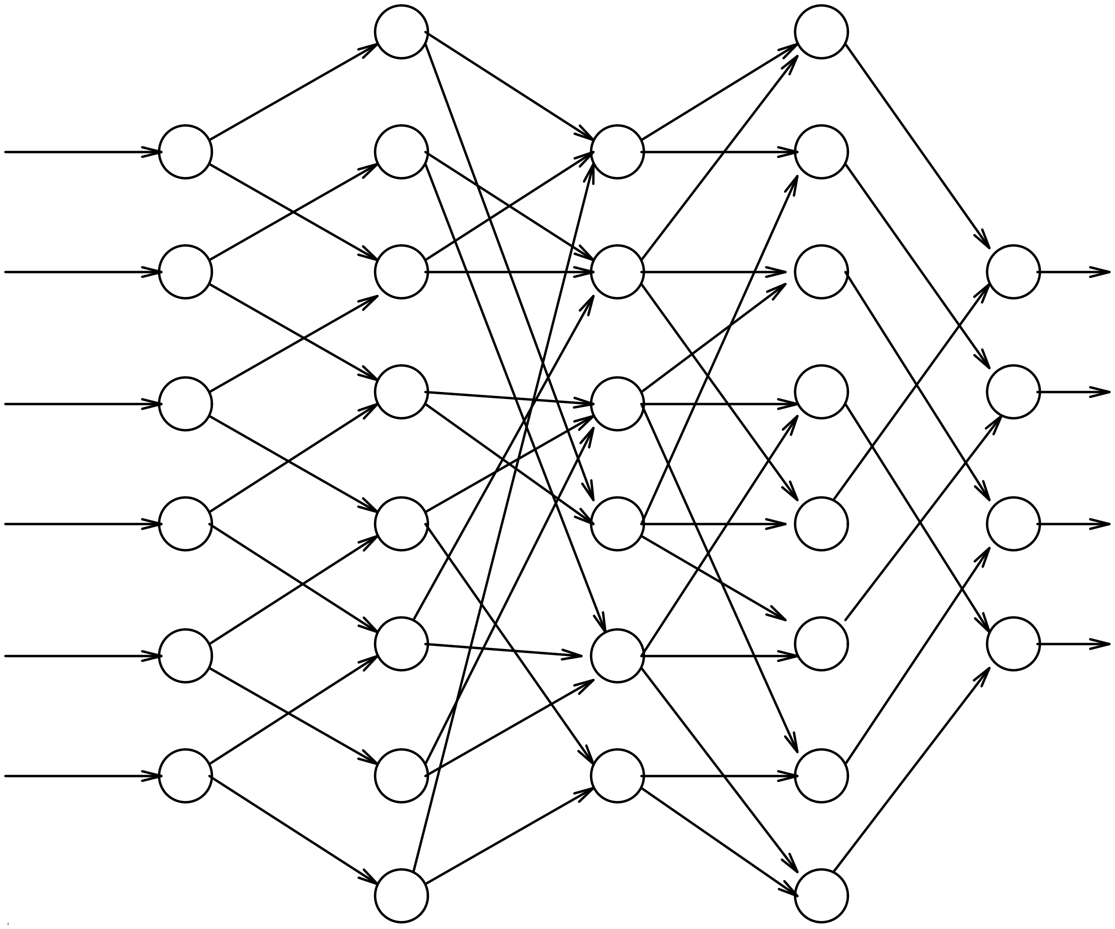

Figure 6.1: An example connectionist network.

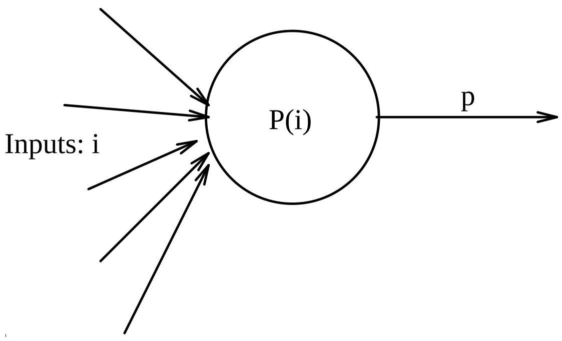

Figure 6.2: The conceptual operation of a single processing element.

single value until a new input image is considered. Of course, if we were recognizing objects in an image sequence, we could modify this system to work with time varying signals. Also, in an effort to follow biological models, Feldman limits his units to on the order of 10 discrete output values. I have not constrained my approach in this manner. While biological systems provide an example of highly powerful parallel computation that connectionist systems can strive to attain and understanding those limitations may help understand how such computation is possible, we need not limit computer systems in this manner.

6.3 Object recognition using an election

Indexing can be used with many recognition strategies. Any algorithm that uses hypothesized matches between groups of image features and model features is a prime candidate. For example, alignment methods [Fischler and Bolles, 1981, Lowe, 1987, Huttenlocher and Ullman, 1990] and pose clustering methods [Linnainmaa *et al.*, 1988, Stockman, 1987, Thompson and Mundy, 1987, Olson, 1994b] use such hypotheses to determine the transformation aligning the matches. These transformations are then used in varying ways to localize the object. Indexing can be used to determine which matches to use to determine possible transformations. In addition, alignment methods can use indexing to determine which additional point matches are brought into correspondence given the initial group match.

Since we will need a large number of processing elements, we wish to make them as simple as possible. Alignment and pose clustering are not ideal for connectionist algorithms, since they depend on the determination of the transformation aligning groups of feature points, which is a more complicated task than we wish to require our processing elements to perform. It is possible to use precomputation to shift a significant amount of the work off-line. The following subsection describes a recognition system that does this, making it more suitable for connectionist implementation.

6.3.1 Election methods

Lamdan *et al.* [1990] describe a recognition strategy called geometric hashing. The basic idea is to index the model groups that could have projected to a number of image groups and use these model groups to vote for the models from which they come. If the image groups that are used are chosen in the correct manner, then the voting can determine which objects are present in the image.

Their system recognizes objects as follows:

1. An index table is generated. This is done by examining each basis of three model points. For each basis, the relative coordinates of every other model point are determined and a record describing which points generated these coordinates is stored in the appropriate bin of the index table corresponding to the relative coordinates. These relative coordinates are invariant to affine transformations and orthographic projection. This step is performed off-line, prior to recognition time.

2. Feature points in the image are determined.

3. A basis of three image points is chosen at random.

4. For every other image point, the relative coordinates with respect to the image basis are determined.

5. For the current image basis and each of the additional image points, possible matching model groups are determined by indexing the table using the relative

coordinates.

6. A vote is recorded for each model basis indexed.

7. If some model basis receives enough votes, the process is stopped and the object is considered recognized. Otherwise Steps 3-7 are repeated, until we have examined enough image bases to rule out the presence of the object in the image.

A voting method of this type could be used with any indexing system. Indeed, if we wish to recognize general three-dimensional point sets, an alternate indexing method will be required. I shall call algorithms that combine an indexing system and voting in this manner *election methods.* In this chapter, a set of model or image points of minimal size to constrain the pose to a finite set of points will be called a *model basis* or an *image basis.* Model groups and image groups contain one more point than model bases or image bases. For example, to recognize three-dimensional models, three points are necessary to constrain the pose, so model and image bases consist of three points and model and image groups consist of four points.

6.3.2 Analysis of election methods

The analysis of geometric hashing by Grimson *et al.* [1990b] can be easily generalized to apply to general elections methods. Let m be the number of model points, n be the number of image points, and k be the number of points used to generate an indexing key. Grimson *et al.* call the average probability that a specific image group indexes a random model group the selectivity $\overline{\mu}$.

For a specific image basis, the probability that a particular image point will index a model group containing some specific model basis and any other model point is:

$$p = 1 - (1 - \overline{\mu})^{m-k+1}$$

since there are $m - k + 1$ model groups containing some model basis of $k - 1$ points. So, the probability of having a false match of size x between specific model and image bases is:

$$q_x = 1 - \sum_{i=0}^{x-1} \begin{pmatrix} n-k+1 \\ i \end{pmatrix} p^i (1-p)^{n-k-i+1}$$

We are primarily concerned with $x = fm - k + 1$, where f is the fraction of model points appearing in the image, since a conservative indexing system will find each of the additional $fm - k + 1$ matches for any correct image basis. The probability of a false match of size x will be found for a specific image basis over the entire model is:

$$r_x = 1 - (1 - q_x)^{\binom{m}{k-1}}$$

The probability of a false match over the entire set of image bases is difficult to calculate due to correlation in the number of additional matches that can be brought into alignment between image bases, but clearly a match of at least size $fm - k + 1$ will result from a conservative indexing system if there exists a transformation that aligns fm model points with distinct image points. So, the probability of a false match of size $fm - k + 1$ for any image basis and model basis must be at least as large as the probability that fm points can be brought into alignment up to the error bounds by some transformation, which can be significant in complex images. Chapter 3 discussed why this is a fundamental bound on all systems using only feature points to generate hypotheses.

It is important to note that in all of the election methods, we tally votes for model groups indexed only by image groups sharing a common basis. This is very important to the accuracy of election methods. Assume that we have an indexing system that indexes all of the correct model groups and, on average, some percentage p of the incorrect model groups for any specific image group. If there are fm correct model points appearing in the image, when using a basis of size κ, we expect a correct basis to accumulate

$$v_g = \begin{pmatrix} fm - \kappa \\ k - \kappa \end{pmatrix} = O((fm)^{k-\kappa})$$

correct votes and an incorrect basis to accumulate

$$v_b = p \begin{pmatrix} n - \kappa \\ k - \kappa \end{pmatrix} = O(n^{k-\kappa})$$

random votes. The second increases faster than the first as $k - \kappa$ increases, since $n > fm$. Thus, we wish to make $k - \kappa$ as small as possible (but it must be at least one to be useful,) which occurs when $\kappa = k - 1$.

6.4 Connectionist implementation

In a connectionist implementation, we can examine each image basis simultaneously. Massive parallelism is thus possible. This implementation requires a small amount of memory per unit. In particular, only the excitation functions must be known. (The inputs are known implicitly by the connections to the unit.) In general, the implementation can be broken into layers of processing elements representing the following conceptual entities:

Layer 1: Image feature points
Layer 2: Image point groups
Layer 3: Matches between image groups and model groups
Layer 4: Matches between image bases and model bases
Layer 5: Models

The indexing process consists of the first three layers of the network and the election is performed by the final two layers. The computations that take place in each of these layers and the communication pattern between them is as follows:

Layer 1: Image feature points. The coordinates of the feature points comprise the input layer. This layer does not perform any computation, the units just feed values to the correct processing elements in the following layer.

Layer 2: Image point groups. The units in the second layer correspond to groups of image features of size k (which is the number of points used to perform indexing.) These units generate the indexing parameters. For various indexing systems these parameters are easily computed functions of the feature point coordinates (see Chapter 2). Each of the units in this layer receives input from the units in the previous layer corresponding to the k feature points that make up the image group and sends output to each of the group matches that match the image group to a model group.

Layer 3: Group matches. The units at the third layer represent matches between image groups and model groups. These units are assumed to know the parameters of the model group in the match they are evaluating, although this may be implicit in the function they are performing. These units determine the likelihood that the match they represent is correct. This is done by calculating how far the image group parameters are from the model group parameters using some distance metric. Each of these units receives input from only the unit in the previous layer corresponding to the image group in the group match and sends output to the units corresponding to the k basis matches contained in the group match.

Layer 4: Basis matches. The fourth layer corresponds to the voting for each possible model basis, except that unlike the sequential algorithm, each of the image bases is considered simultaneously. Each unit at this layer receives output from each of the units at the previous layer that includes the match of the given image and model basis. These likelihoods are combined to determine a score for the basis match. A large score at this stage indicates that the basis match is likely to be correct. Each of these units receive input from all of the $O(mn)$ group matches contain the basis match and sends output to the correct model unit.

Layer 5: Models. Finally, the last layer determines which objects are present in the image. If more than one instance of a model may be present, this layer may be omitted and the information on basis matches from the previous level can be used to indicate which objects may be present in the image. This layer compresses the information by performing a maximum on the inputs. A large score at this stage indicates a hypothesis that the model is present in the image. These units receive input from each of the basis matches for this model.

Figure 6.3 shows how the information flows in this process. The image point locations feed into image group units, which determine the indexing parameters. These are then passed to the group match units, which give each model group a likelihood of projecting to the image group. The basis match units take the scores from each of the group matches that contain the appropriate basis match and perform a combining operation to get the score for the basis match. Finally, the model units

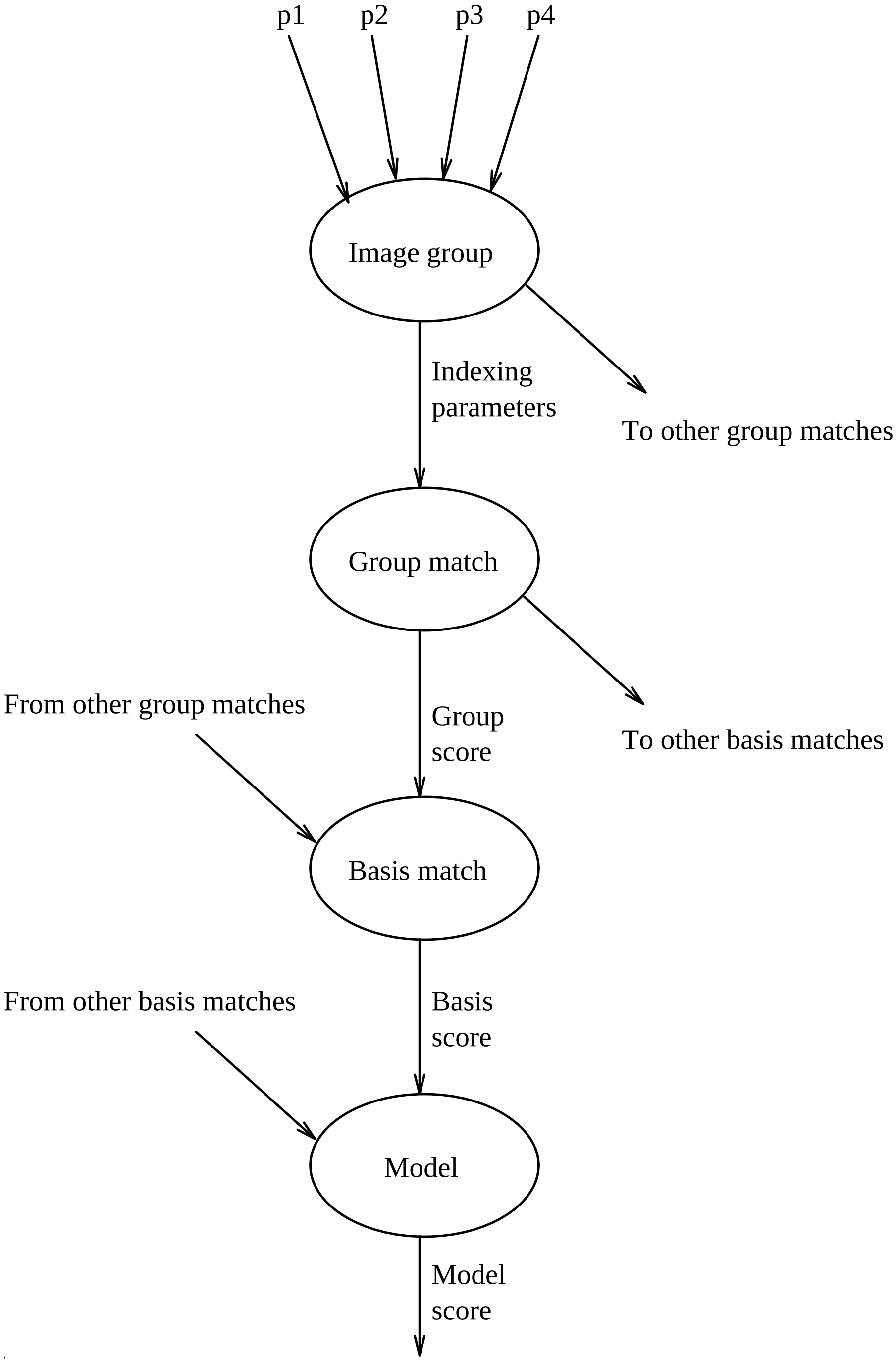

Figure 6.3: The flow of information in the connectionist indexing process.

output a high score if any of their basis matches have a high score.

As formulated above, this system would require $O(m^k n^k)$ processing elements per object since there are $O(m^k n^k)$ possible matches between an image group and a model group, but we can reduce this considerably using randomization and still achieve accurate object recognition. If we optimistically assume that our units will have the necessary fan-in and fan-out, the running time of the system is the sum of the times required at each the of levels and the communication times between the layers. Since this is bounded by a constant, the running time would thus be $O(1)$ and would be very fast in practice. The time required when we have limited constant fan-in and fan-out units, as well as the overall number of units required will be discussed further in the following section.

6.4.1 Number of processing elements required

Modifying the analysis of Lamdan *et al.* [1990] we can see that we can achieve any fixed probability of examining a correct model basis while examining much less than $O(m^k n^k)$ matches. If we choose some number x of image bases of size $k-1$ at random and examine only matches that involve those bases, we can see that the probability of not examining any correct bases is:

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

where f is the smallest fraction of model points that must appear in the image to obtain accurate recognition, since the probability of any particular point being a correct model point is at least $\frac{fm}{n}$ in this case. If we require this probability to be less that some small constant δ we get:

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

Taking the logarithm of both sides yields:

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

Solving for x we get:

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

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

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

So, we can examine $O(\frac{n^{k-1}}{m^{k-1}})$ image bases and achieve high accuracy. For each image basis, we consider each of the $O(m^{k-1})$ model bases as possible matches and for each possible match we examine each of the $(n-k+1)(m-k+1) = O(mn)$ additional point matches to determine if the match between the bases is correct. In total we must now examine $O(\frac{n^{k-1}}{m^{k-1}}) \cdot O(m^{k-1}) \cdot O(mn) = O(mn^k)$ group matches per object.

For the connectionist implementation, we can simply select the appropriate number of random bases to examine in parallel and we can achieve probability $1-\delta$ of examining a correct image basis. The processing elements corresponding to the remainder of the matches are unnecessary. We use only $O(mn^k)$ units per object in this case. Of course, in practice, the number of processing elements available will set some limit on the number of image features that can be handled with this accuracy. Performance will be good on images of this complexity or less and will degrade gracefully as the image complexity rises past this level.

It is now possible to describe the number of units required at each level, as well as the fan-in and fan-out required at each unit. We need approximately $\alpha(k-1)!n^{k-1}$ basis matches (where $\alpha = \frac{\ln \frac{1}{\delta}}{f^{k-1}}$, and thus is constant when f and δ are set,) since for each image basis we examine we must examine each of the $(k-1)!$ orderings of each of the $\binom{m}{k-1}$ model bases that may match it. This implies that the number of group matches that must be examined is approximately $\alpha k!(m-k+1)(n-k+1)n^{k-1}$. The required fan-in and fan-out of each can be determined from the connection pattern specified in the previous section. See Table 6.1. Overall, $O(mn^k)$ units are required per object with a maximum fan-in and fan-out of $O(n^{k-1})$. Alternately, we could implement this system using a constant number of layers with $O(n)$ fan-in/fan-out

Layer	Description	Number	Fan-In	Fan-Out
1	Image points	n	1	$\binom{n-1}{k-1}$
2	Image groups	$\binom{n}{k}$	k	$\alpha k!(m-k+1)$
3	Group matches	$O(mn^k)$	1	k
4	Basis matches	$O(n^{k-1})$	$(m-k+1)(n-k+1)$	1
5	Models	1	$\alpha(k-1)!\binom{n}{k-1}$	1

Table 6.1: The number of units and fan-in/fan-out necessary at each layer. Note that α is a function of δ and f, but is constant for when δ and f are set. See the text for details.

units, or we could use units with a constant fan-in/fan-out and build up the required number of outputs using a tree with $O(\log n)$ layers.

This analysis of the number of units required is on a per object basis. As the number of objects in the database increases, the number of units required increases linearly. It is possible to reduce the number of units required at the group match stage by combining units for each match between an image group and model groups that share the same (or very similar) indexing parameters. These units receive the same inputs since they share the same image group and perform the same function since the model groups have the same indexing parameters. These units would then send output to each of the basis match units that the original units output to. The formulation would then require $O((l+m+n)n^{k-1}+I)$ units where I is the number of units necessary to cover the space of indexing parameters.

6.4.2 Geometric hashing

The geometric hashing system [Lamdan *et al.*, 1990] can be easily implemented in the framework discussed above. The units corresponding to the image point groups simply determine the relative coordinates of the fourth point in terms of the first three. The units corresponding to the group matches are assumed to know the values corresponding to the correct model group. These units output a high value if the relative coordinates input to them are close to the model group relative coordinates held there.

While Lamdan and Wolfson perform binning to determine which image and model groups may match, we need not follow their algorithm exactly. A much better way to determine the output for the units corresponding to group matches would be use to a Bayesian formulation as in Costa *et al.* [1989] or Rigoutsos and Hummel [1993]. In these works, the geometric hashing problem is recast in the realm of Bayesian probability. Rather than indexing being a discrete event where each image group either indexes or doesn't index each specific model group, each possible match is given a score related to the probability of the image parameters being generated by a group of image features given that the model appears in the image. These scores are used to determine the probability of each basis match being correct. In these formulations, only the matches with large scores are examined. The rest of the matches contribute small values to the total score, so these are ignored to gain an indexing speedup. In a connectionist implementation, we do not need to ignore these matches, and so we can gain additional accuracy.

In the connectionist approach, the units corresponding to the group matches determine the Bayesian probability of matching. The next layer (corresponding to basis matches) performs a summing operation on the outputs of the units at the previous level that include each basis match. Finally, we can perform a maximum operation on the basis matches from each model to determine if the model is present in the image.

6.4.3 Indexing 3D models (Clemens and Jacobs)

The work of Clemens and Jacobs [1991] can also be put in this framework. While much of the implementation is very similar to that for geometric hashing, this indexing technique involves the additional complication that they represent each model group from each direction on a discretized viewing sphere in their index table. This can be handled by including in Layer 3, a unit corresponding to each of the viewing directions for each match. An additional layer is included (Layer 3.5) in which the maximum value over all of the viewing directions for the matches is determined as the output for each group match.

Jacobs [1992] has modified this work for affine transformations such that sampling the viewing sphere is not necessary. In this system, each group is represented by a line in each of two two-dimensional parameter spaces. To index model groups that may match an image group, the image group parameters are determined and model groups are looked up in both spaces. The intersection of the two sets of model groups correspond to the possibly matching points. Implementing this modification in the connectionist framework does not require a unit corresponding to each viewing direction for each group match. For this case, units are required to store the description of two lines in a two-dimensional space and be able to calculate some distance function of their input from these lines. We can then use one unit for each match. The unit determines the distance of the image group parameters from the lines in the the parameter spaces. The larger of these two distances is used to determine the score for the match. Since Jacobs' system uses point groups of size five, $O(mn^5)$ units are required to implement this system while $O(mn^4)$ were required for the previous systems.

6.4.4 Indexing 3D models (Weinshall)

Weinshall's [1993] formulation of the indexing problem can yield additional benefits for a connectionist implementation. Weinshall describes model-based invariants that can be used to determine if a model group can be brought into alignment with an image group. Rather than viewing each model group as representing some space in an index table, we can simply compute the model-based invariant that Weinshall describes. A single unit can thus easily determine the score for a group match.

The model-based invariant is given directly in terms of image feature point coordinates, so no parameters specific to an image group are generated. The connectionist implementation of these ideas thus does not require a layer of units corresponding to the image point groups. The units corresponding to the group matches compute the invariant functions and output a high value if they are close to zero (the value a noiseless correct match would yield.)

An alternative metric that could be used is the transformation metric defined in [Basri and Weinshall, 1992] which is very similar to the model-based invariants. I have found this metric to be more accurate in discriminating between correct and incorrect matches.

Due to the simplicity of representing group matches, this method appears to be the best to use for a connectionist election method to recognize 3-d objects.

6.4.5 Probabilistic indexing

The connectionist implementation of the probabilistic indexing system in this framework is also straightforward. While probabilistic indexing systems are capable of indexing using groups of size three, this is probably not suitable for an election method, due to the significant percentage of false positive matches indexed for this case. Probabilistic indexing has been extended to handle point groups larger than three points. When using point groups of size four, we can index many fewer false positives while still indexing a significant fraction of the correct groups. To perform probabilistic indexing on groups of larger than three points, we perform indexing on each of the subgroups of three points. If a model group is indexed by enough of the subgroups then the group is considered indexed by the entire group. For four point groups we thus need either four units for each group match to perform the probabilistic indexing on the four subgroups of three points or a single unit capable of performing all four calculations. An additional calculation must then be performed to combine the results of the previous calculations. An additional unit can be used to perform this task, if desired.

Due to the false negatives obtained as a result of using the probabilistic indexing system, we must reduce the vote total necessary to consider a model basis a possible match. The units corresponding to the basis matches still receive information from each of the group matches that contain that basis match, but these units will now output a high value based on a lower number of high input values, since not all of the correct matches will be indexed. This is usually acceptable, since a considerably smaller percentage of the incorrect matches will also be indexed for most cases.

6.5 Alternate recognition methods

While election methods are ideal for connectionist implementations since they shift much of the work offline, other recognition methods can be implemented in this framework.

6.5.1 Alignment methods

While alignment methods [Huttenlocher and Ullman, 1990] perform a similar operation to election methods, we can consider a connectionist formulation that closely follows the ideas in sequential alignment methods. Conceptually, the difference is that election methods pre-process the model groups so that less work needs to be done at run-time. Indexing is then used in the election methods to retrieve this information efficiently at run-time.

In a connectionist formulation of alignment, the basis matches are considered both before and after the group matches. They must be examined before the group matches to determine the transformation that brings the points into alignment. They are then examined again after the group matches to determine how many additional matches can be brought into alignment with each of them.

Alignment can be implemented using the following layers of processing elements:

1. Image points: This is the input layer. Units here simply feed the values to the correct units in Layer 2.

2. Image bases: This layer assembles the image bases and computes relevant values.

3. Basis matches (first): These units determine the transformation that aligns the corresponding basis match.

4. Group matches: These units determine if the transformation aligning the basis match align each possible additional match.

5. Basis matches (second): These units count how many additional matches were brought into alignment for each basis at the previous layer.

6. Models: These units determine if any bases for a model received enough votes to merit further verification.

This formulation still assumes that the units in Layers 3 and 4 have model feature point locations stored locally, but this could be avoided by treating the model feature point locations as additional input values that feed directly to Layers 3 and 4. In this implementation, units must compute the transformation in the third layer. All of parameters of this transformation must then be passed on to the appropriate units in the next level. Using randomization, this method requires $O(mn^4)$ processing elements for the problem of recognizing three-dimensional objects from intensity images.

6.5.2 Pose clustering

Pose clustering techniques [Ballard, 1981, Thompson and Mundy, 1987, Stockman, 1987, Linnainmaa *et al.*, 1988, Olson, 1993c] determine the transformations that align many possible matches between small sets of model and image features. Instances of objects correspond to a cluster of transformations in pose space.

A connectionist implementation of pose clustering methods requires implementation of the clustering stage, which can be problematic. One possibility is to sample the pose space and use a unit for each sampled point. For systems recognizing objects undergoing rigid 3D transformation, we have a 6D pose space, so this will required a large number of units unless we sample coarsely.

The connectionist layers necessary to perform object recognition using this approach are:

1. Image points: The input layer.

2. Group matches: These units at this layer determine the transformation aligning each of the group matches.

3. Pose space clusters: These units determine the clusters of transformations in the transformation space.

4. Models: These units determine if a model had any clusters of sufficient size to indicate presence of the model in the image.

In Chapter 3, I showed that randomization can be used to limit the number of poses that need be examined in a pose clustering system. Thus, pose clustering on 3-d objects using this method could be accomplished using $O(mn^3)$ units plus the units to perform the clustering.

6.6 Discussion

This section discusses the overall running time of the system and the possibility of learning the object models.

6.6.1 Running time

Previous recognition algorithms on sequential systems require (in general) a significant amount of processing time. By using a large number of simple units, the system I have presented can reduce this processing time immensely.

If we have sufficient units with $O(n)$ fan-in and fan-out capabilities, the running time is $O(1)$, since each the computation required at each layer can be computed in constant time and there are a constant number of layers (the $O(n^{k-1})$ fan-in and fan-out capabilities can be simulated using a tree of units of constant height.) If we are constrained to use units with limited fan-in and fan-out capabilities, we can chain units in a tree with $O(\log n)$ height to provide the necessary capability. Thus, in this case the running time of the system is $O(\log n)$.

Feldman [1985] suggests that since human neurons operate in the millisecond range but our reaction times are on the order of a few hundred milliseconds, we must perform about only about 100 computational steps in our object recognition processes. It is heartening to note that this system can thus meet the limitation of 100 computational steps, since even in the worst case of small fan-in/fan-out units the system requires $O(\log n)$ time (with reasonably small constant factors.)

6.6.2 Learning

In the implementations described above, I have assumed that the necessary indexing parameters are loaded into the correct processing element by some method prior to recognition time. An interesting possibility to consider is whether these parameters could be learned in a supervised or unsupervised manner through the examination of examples.

Many connectionist systems use linear functions to combine the inputs at each unit and then output some nonlinear function of the result. Such systems can be trained using the backpropagation algorithm [Rumelhart *et al.*, 1986]. I let my processing elements perform arbitrary (although usually simple) non-linear functions and there are no weights associated with the edges in the connection graph. In addition, given the number of model and image points that we wish to handle, the communication pattern is completely fixed, regardless of the object and image to recognize. The parameters that vary in this system are the indexing parameters stored at the units corresponding to group matches. These parameters can also be trained by a backpropagation algorithm.

While it is conceivable that the indexing parameters for an entire object could be learned concurrently using backpropagation techniques, a much superior learning strategy would be to isolate each of the units corresponding to the group matches and train them separately. If this can be achieved, learning the parameters would be simple for most indexing systems. Otherwise learning the parameters would be a formidable task.

At this time I have made no attempt to train such a system.

6.7 Summary

I have demonstrated that extremely fast object recognition is possible using a connectionist network if we have enough processing elements. A framework for the connectionist implementation of various object recognition strategies has been given. This framework uses a large number of simple processing elements connected in a feedforward manner to determine which objects are present in an image. The implementation of election algorithms using several specific indexing systems in this framework has been discussed. The time required by such an implementation has been shown to be $O(1)$ or $O(\log n)$ depending on the fan-in/fan-out capabilities of the processing elements.

Chapter 7

Conclusions

This thesis has considered methods of improving the efficiency of model-based object recognition techniques. In particular, I have been interested in the case of recognizing three-dimensional objects from monocular intensity images. The primary idea that has unified these techniques is that we need not examine all of the huge number of hypotheses that could be correct. Selective examination of such hypotheses can lead to greater efficiency in the recognition of objects, while not significantly reducing the accuracy of the recognition.

One of the outcomes of this research has been the development of a framework for the pose clustering method of object recognition similar to that which has been done for the alignment method of object recognition, where hypotheses are determined and then verified. While, the alignment method is one of the leading methods of object recognition, it is my hope that this work on pose clustering will help convince people that it is as good as (probably better than) the alignment method as a strategy of determining which hypotheses are probably correct from sets of local features, due to its efficiency and accuracy.

In general, this work can be used with any set of local features from which the pose of the object can be determined. If we require matches between k of these features to estimate the pose, then the time required by these techniques is proportional to the kth power of number of image features and is linear in the number of model features. Furthermore, the space required by this system is linear in both the number of model features and the number of image features.

My analysis has demonstrated that every system using only feature points to generate hypotheses is limited in the accuracy it can achieve. These limitations can be generalized to any set of features. (The limitations imposed by a more informative set of features would be less restrictive.) While my pose clustering system does not achieve the best accuracy possible under this limitation, it provides a reasonable approximation, and can be used as a method of finding good hypotheses for further verification. It should be noted that the alignment method inherently cannot achieve the accuracy upper bound because it finds those points that can be brought into alignment with each basis match independently (unless additional work is done, e.g. [Jacobs, 1991].) Since these additional points are examined separately, it does not guarantee that all of these points can be brought into alignment simultaneously by a single pose. This causes alignment methods to find extra false positives.

While this work on pose clustering shows that we do not need to examine all of the possible hypotheses for this problem to achieve accurate recognition, it does not provide us with an a priori means of determining which of the hypotheses are the best to examine. This thesis has also developed the idea that the small sets of matches between image and model features that are used to generate hypotheses can be selected intelligently based on probabilistic information yielded by the probabilistic peaking effect. I have used this effect to build an indexing system that is capable of indexing feature point sets of size three, where previous systems required the sets to be of at least four points. This reduction in the number of points required allows us to gain an indexing speedup in techniques that were previously unable to use indexing.

These probabilistic indexing techniques have been used with error criteria to achieve a large speed up in the alignment method, although when these techniques are used in conjunction with randomization to limit the number of hypotheses that must be examined, the speedups are smaller. While an analysis of the use of probabilistic indexing techniques with pose clustering has not yet been completed, it is expected that they will also yield a significant speedup when applied to this system.

Finally, this work shows that many model-based object recognition techniques can be implemented simply in a connectionist framework. The use of randomization is key to limiting the number of processing elements that are required in this framework. Given a sufficient number of processing elements, this approach allows the model-based object recognition techniques to be implemented in real time. While parallel implementations have been considered by many researchers, this work shows how model-based recognition techniques can be implemented using fine-grain parallelism at an unprecedented scale.

These techniques have been bound together by a common purpose, to make object recognition techniques more efficient. This has been achieved by examining selected hypotheses in an intelligent manner. Grouping techniques [Ahuja and Tuceryan, 1989, Huttenlocher and Wayner, 1992, Lowe, 1985, Mohan and Nevatia, 1992] give us means of determining which image features are most likely to come from the same object and thus help limit the search necessary, but comparatively little work has been done on which of the matches between image features and model features are most likely to be correct prior to the time consuming verification steps. It is my hope that this work will encourage further research in this area, since the improvements that can be gained in efficiency are substantial.

7.1 Future directions

In this section, I'll describe where I expect this research to lead in the near future, but I will also comment on future directions for the field of object recognition.

There are several aspects of the pose clustering techniques that have been described here that I would like to investigate further. The application of the probabilistic indexing techniques to pose clustering is one of these. A complete study of the theoretical and practical inaccuracies implied by the use of alignment, geometric hashing, and pose clustering is planned. Since alignment and geometric hashing do not consider all of the feature matches in the context of the same pose (hypotheses are bases on which additional matches can be brought into alignment at the same time as some basis, but the additional matches are considered separately,) it is my hope that analysis will show that this fundamental limitation of alignment and geometric hashing significantly limits the accuracy of recognition in comparison with pose clustering techniques.

In addition, pose clustering techniques are very similar to Hough transform techniques for the detection of curves such as lines, circles, or even arbitrary parameterized curves [Hough, 1962, Duda and Hart, 1972, Ballard, 1981]. Some of the techniques that I use with pose clustering have already been used in a limited form with such methods, but I expect that a more complete formulation of such techniques will lead to a curve detection method with near optimal accuracy and high speed.

Another area that merits further research is a study of which features are useful for indexing (probabilistic or otherwise). Current research has focused on feature points. A study of how indexing should be performed with more informative features (particularly in the three-dimensional case) would be interesting. Generalizing these techniques to apply to curved three-dimensional objects appears difficult but would be very useful. Along these lines, an examination of probabilistic viewing effects of curved objects should also be useful as vision researchers explore the best methods of recognizing such objects.

7.1.1 A broader perspective

While there are still efficiency issues, it is now possible to recognize polyhedral objects from intensity images. The majority of the work on recognizing these objects has been using feature points and sometimes line segments. While the problem of recognizing curved objects in intensity images has been examined (e.g. [Basri and Ullman, 1988],) much more work needs to be done in this direction. The current reliance of systems on feature points is very limiting in this regard, since general curved objects do not have stable feature points as they are rotated in space.

As algorithms become better, the reliance on weak-perspective as an imaging model will probably become too restrictive. My own experiments indicate that the use of the weak-perspective model is causing significant error in the pose estimation step, which is key to both the alignment and pose clustering methods of object recognition. The use of the full perspective projection or a more accurate approximation appears very desirable. Overcoming the problems we face in using these imaging models is important.

Obviously, massive parallelism will be necessary to implement vision systems in real time. Many authors have commented on the inherent parallelism in their algorithms. Furthermore, the connectionist approach to object recognition presented in this thesis shows that such parallelism is possible even at a very fine level. It is encouraging that most of the current object recognition techniques yield nicely to parallel implementations, but such considerations must continually be kept in mind.

The recognition of curved three-dimensional objects appears to be the final frontier for CAD-based vision systems. A significant improvement that will be required if we are to achieve general object recognition is to go beyond the geometric information present in the object and image features. Steps in this direction have been taken by several researchers (e.g. [Stark and Bowyer, 1991, Strat and Fischler, 1991],) but a general solution to this problem appears to require vast knowledge and reasoning capabilities far beyond what is currently the state of the art for computers. This additional frontier for general object recognition systems will include hard problems from artificial intelligence and other fields, such as retrieval from vast databases (text and image,) common sense reasoning, massively parallel computing, context-based reasoning, and many others. It will be interesting to see the future directions research in object recognition take.

Bibliography

[Ahuja and Tuceryan, 1989] N. Ahuja and M. Tuceryan. Extraction of early perceptual structure in dot patterns: Integrating region, boundary, and component gestalt. *Computer Vision, Graphics, and Image Processing*, 48:304–356, 1989.

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

[Alter, 1992] T. D. Alter. 3D pose from 3 corresponding points under weak-perspective projection. A. I. Memo 1378, Massachusetts Institute of Technology, July 1992.

[Arbter *et al.*, 1990] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger. Application of affine-invariant Fourier descriptors to recognition of 3-d objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):640–647, July 1990.

[Ayache and Faugeras, 1986] N. Ayache and O. D. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54, January 1986.

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

[Barrow and Tenenbaum, 1981] H. G. Barrow and J. M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116, August 1981.

[Basri and Ullman, 1988] R. Basri and S. Ullman. The alignment of objects with smooth surfaces. In *Proceedings of the International Conference on Computer Vision*, pages 482–488, 1988.

[Basri and Weinshall, 1992] R. Basri and D. Weinshall. Distance metric between 3d models and 2d images for recognition and classification. A.I. Memo 1373, Massachusetts Institute of Technology, July 1992.

[Basri, 1992] R. Basri. The alignment of objects with smooth surfaces: Error analysis of the curvature method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 341–346, 1992.

[Ben-Arie and Meiri, 1987] J. Ben-Arie and A. Z. Meiri. 3d objects recognition by optimal matching search of multinary relations graphs. *Computer Vision, Graphics, and Image Processing*, 37(3):345–361, March 1987.

[Ben-Arie, 1990] J. Ben-Arie. The probabilistic peaking effect of viewed angles and distances with application to 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):760–774, August 1990.

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

[Biederman, 1985] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.

[Binford *et al.*, 1989] T. O. Binford, T. S. Levitt, and W. B. Mann. Bayesian inference in model-based machine vision. In L. N. Kanal, T. S. Levitt, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*, pages 73–95. Elsevier Science Publishers B.V. (North-Holland), 1989.

[Binford, 1982] T. O. Binford. Survey of model-based image analysis systems. *International Journal of Robotics Research*, 1(1):18–64, 1982.

[Breuel, 1992] 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.

[Burns *et al.*, 1990] J. B. Burns, R. Weiss, and E. M. Riseman. View variation of point set and line segment features. In *Proceedings of the DARPA Image Understanding Workshop*, pages 650–659, 1990.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–697, November 1986.

[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*, pages 879–884, 1988.

[Cass, 1990] 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.

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

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

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

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

[Clowes, 1971] M. B. Clowes. On seeing things. *Artificial Intelligence*, 2:79–116, 1971.

[Costa *et al.*, 1989] M. Costa, R. Haralick, T. Phillips, and L. Shapiro. Optimal affine-invariant point matching. In *Applications of Artificial Intelligence VII, Proc. SPIE 1095*, pages 515–530, 1989.

[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, 1972] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15:11–15, 1972.

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

[Faugeras and Hebert, 1986] O. D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *International Journal of Robotics Research*, 5(3):27–52, 1986.

[Feldman, 1985] J. A. Feldman. Connectionist models and parallelism in high level vision. *Computer Vision, Graphics, and Image Processing*, 31:178–200, 1985.

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

[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.

[Flynn and Jain, 1991] P. J. Flynn and A. K. Jain. BONSAI: 3d object recognition using constrained search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1066–1075, October 1991.

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

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

[Forsyth *et al.*, 1991] D. Forsyth, J. L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3-d object recognition and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):971–991, October 1991.

[Gaston and Lozano-Pérez, 1984] P. C. Gaston and T. Lozano-Pérez. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3):257–265, May 1984.

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

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

[Grimson and Lozano-Pérez, 1984] W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[Grimson and Lozano-Pérez, 1987] W. E. L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.

[Grimson *et al.*, 1992a] 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.

[Grimson *et al.*, 1992b] W. E. L. Grimson, D. P. Huttenlocher, and D. W. Jacobs. A study of affine matching with bounded sensor error. In *Proceedings of the European Conference on Computer Vision*, pages 291–306, 1992.

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

[Grimson, 1991] W. E. L. Grimson. The combinatorics of heuristic search termination for object recognition in cluttered environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):920–935, September 1991.

[Haralick and Shapiro, 1979] R. M. Haralick and L. G. Shapiro. The consistent labeling problem: Part 1. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:173–184, April 1979.

[Haralick *et al.*, 1991] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions of the three point perspective pose estimations problem. In *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 592–598, 1991.

[Hart *et al.*, 1968] P. E. Hart, N. J. Nillson, and B. Raphael. A formal basis to the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

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

[Huffman, 1971] D. A. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 6:295–323, 1971.

[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, 1987] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proceedings of the International Conference on Computer Vision*, pages 102–111, 1987.

[Huttenlocher and Ullman, 1988] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment. In *Proceedings of the DARPA Image Understanding Workshop*, pages 1114–1124, 1988.

[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.

[Huttenlocher and Wayner, 1992] D. P. Huttenlocher and P. C. Wayner. Finding convex edge groupings in an image. *International Journal of Computer Vision*, 8(1):7–27, 1992.

[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.

[Jacobs, 1992] D. W. Jacobs. Space efficient 3d model indexing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 439–444, 1992.

[Koenderink, 1984] J. J. Koenderink. What does the occluding contour tell us about solid shape. *Perception*, 13:321–330, 1984.

[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.

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

[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):634–647, September 1988.

[Lowe, 1985] D. G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic, 1985.

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

[Mackworth, 1973] A. K. Mackworth. Interpreting pictures of polyhedral scenes. *Artificial Intelligence*, 4:121–137, 1973.

[Malik and Maydan, 1989] J. Malik and D. Maydan. Recovering three-dimensional shape from a single image of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):555–566, June 1989.

[Malik, 1987] J. Malik. Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1(1):73–103, 1987.

[Mohan and Nevatia, 1992] R. Mohan and R. Nevatia. Perceptual organization for scene segmentation and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):616–635, June 1992.

[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.

[Mundy *et al.*, 1988] J. L. Mundy, A. J. Heller, and D. W. Thompson. The concept of an effective viewpoint. In *Proceedings of the DARPA Image Understanding Workshop*, pages 651–659, 1988.

[Olson, 1992] C. F. Olson. Fast alignment by eliminating unlikely matches. Technical Report UCB//CSD-92-704, Computer Science Division, University of California at Berkeley, November 1992.

[Olson, 1993a] C. F. Olson. Fast alignment using probabilistic indexing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 387–392, 1993.

[Olson, 1993b] C. F. Olson. Probabilistic indexing: Recognizing 3d objects from 2d images using the probabilistic peaking effect. Technical Report UCB//CSD-93-733, Computer Science Division, University of California at Berkeley, May 1993.

[Olson, 1993c] 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.

[Olson, 1994a] C. F. Olson. Probabilistic indexing: A new method of indexing 3d model data from 2d image data. In *Proceedings of the Second CAD-Based Vision Workshop*, pages 2–8, 1994.

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

[Rigoutsos and Hummel, 1992] I. Rigoutsos and R. Hummel. Massively parallel model matching: Geometric hashing on the connection machine. *Computer*, pages 33–41, February 1992.

[Rigoutsos and Hummel, 1993] I. Rigoutsos and R. Hummel. Distributed Bayesian object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 180–186, 1993.

[Roberts, 1963] L. G. Roberts. *Machine Perception of Three-Dimensional Solids.* PhD thesis, Massachusetts Institute of Technology, May 1963.

[Rosenfeld *et al.*, 1976] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6):420–433, June 1976.

[Rothwell *et al.*, 1993] C. A. Rothwell, D. A. Forsyth, A. Zisserman, and J. L. Mundy. Extracting projective structure from single perspective views of 3d point sets. In *Proceedings of the International Conference on Computer Vision*, pages 573–582, 1993.

[Rumelhart *et al.*, 1986] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press, 1986.

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

[Stark and Bowyer, 1991] L. Stark and K. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1097–1104, October 1991.

[Stein and Medioni, 1990] F. Stein and G. Medioni. Efficient two dimensional object recognition. In *Proceedings of the IAPR International Conference on Pattern Recognition*, volume 1, pages 13–17, 1990.

[Stein and Medioni, 1992] F. Stein and G. Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, February 1992.

[Stevens, 1981] K. A. Stevens. The visual interpretation of surface contours. *Artificial Intelligence*, 17:47–73, August 1981.

[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.

[Strat and Fischler, 1991] T. M. Strat and M. A. Fischler. Context-based vision: Recognizing objects using information from both 2-d and 3-d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, October 1991.

[Thomas and Finney, 1984] G. B. Thomas and R. L. Finney. *Calculus and Analytic Geometry*. Addison-Wesley, sixth edition, 1984.

[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*, volume 1, pages 208–220, 1987.

[Ullman and Basri, 1991] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, 1991.

[Watkins, 1991] D. S. Watkins. *Fundamentals of Matrix Computations*. John Wiley and Sons, 1991.

[Weinshall and Basri, 1993] D. Weinshall and R. Basri. Distance metric between 3d models and 2d images for recognition and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 220–225, 1993.

[Weinshall, 1993] D. Weinshall. Model-based invariants for 3-d vision. *International Journal of Computer Vision*, 10(1):27–42, 1993.