Technical Report 1374

Geometric Aspects of Visual Object Recognition

Thomas M. Breuel

MIT Artificial Intelligence Laboratory

Geometric Aspects of Visual Object Recognition

by

Thomas M. Breuel

A.B., M.A. Biochemistry and Molecular Biology, Harvard University
(1986)

Submitted to the Department of Brain and Cognitive Sciences
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

May 1992

©1992 Massachusetts Institute of Technology
All rights reserved.

To my Parents.

Geometric Aspects of Visual Object Recognition

Thomas M. Breuel

Submitted to the Department of Brain and Cognitive Sciences on April 28, 1992, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy.

Abstract

Systems (artificial or natural) for visual object recognition are faced with three fundamental problems: the correspondence problem, the problem of representing 3D shape, and the problem of defining a robust similarity measure between images and views of objects.

In this thesis, I address each of these problems:

- I present a recognition algorithm (RAST) that works efficiently even when no correspondence or grouping information is given; that is, it works in the presence of large amounts of clutter and with very primitive features.
- I discuss representations of 3D objects as collections of 2D views for the purposes of visual object recognition. Such representations greatly simplify the problems of model acquisition and representing complex shapes. I present theoretical and empirical evidence that this "view-based approximation" is an efficient, robust, and reliable approach to 3D visual object recognition.
- I present Bayesian and MDL approaches to the similarity problem that may help us build more robust recognition systems.

These results form the basis for a simple, efficient, and robust approach to the geometric aspects of 3D object recognition from 2D image. The results presented in this thesis also strongly suggest that future research directed towards building reliable recognition systems for real world environments must focus on the non-geometric aspects of visual object recognition, such as statistics and scene interpretation.

Thesis Supervisor: Tomaso Poggio
Uncas and Helen Whitaker Professor, Department of Brain and Cognitive Sciences and Artificial Intelligence Laboratory

Acknowledgements

I would like to thank my thesis advisor, Tomaso Poggio, for his support and encouragement over the last six years. I also want to thank my other two committee members, Chris Atkeson and Sandy Pentland, for taking the time to read and evaluate my thesis, and for useful discussion and valuable insights.

Michael Riley has been a good friend over the years. His down-to-earth approach to research has been a significant influence on me. He has also reminded me that there is life after and outside the MIT AI lab, which is important, lest one get lost between Tech square and the infinite corridor.

Robert Thau has not only contributed and helped me through many discussions and comments on the thesis, but he has also tolerated my excentricities for the last few years in our shared office.

I would also like to thank Ronen Basri and Eric Grimson for their reading of parts of the thesis, and the many discussions and debates we have had.

Many other friends and colleagues at the AI lab and LCS have contributed to this thesis, through discussion and friendship. I want to name in particular Tao Alter, David Beymer, Michael Bolotski, Todd Cass, Ellen Hildreth, Anya Hurlbert, Dan Huttenlocher, David Jacobs, David Michael, Ron Rivest, Amnon Sha'ashua, Brian Subirana, Ali Taalebi, and Shimon Ullman.

Without the help and guidance of Manfred Eigen, it would have been difficult for me to realize my plans and scientific aspirations. He has also permitted me to attend the yearly Winterseminar in Klosters, which has been an important source of inspiration (as well as a chance to take a break). Through Manfred Eigen, I made many interesting and significant scientific contacts; among them, I met Christoph von der Malsburg, whose work has been a significant inspiration for the work reported in this thesis.

Matthew Meselson has been my undergraduate thesis supervisor. While I may have strayed from the path of pure biological research, I am grateful to him for helping me become a scientist.

I also want to thank those people who keep the MIT AI lab running. Liz Highleyman and Jeanne Speckman have been patient, kind, and helpful, even when the stress of my thesis has been getting to me.

Jan Ellertsen has been of tremendous help overcoming the difficulties and administrative obstacles of an MIT graduate student career and submitting a thesis from far away.

The GNU project of the Free Software Foundation and the SML of NJ project at AT&T, CMU, and Princeton have provided excellent software tools without which this thesis would have been very difficult to complete.

But most of all, I want to thank my parents for their love, support, and endless patience.

This thesis describes research done within the Center for Biological Information Processing, in the Department of Brain and Cognitive Sciences, and at the Artificial Intelligence Laboratory. This research is sponsored by a grant from the Office of Naval Research, Cognitive and Neural Sciences Division; by the Alfred P. Sloan Foundation; by the National Science Foundation; by the Artificial Intelligence Center of Hughes Aircraft Corporation (LJ90-074); by E.I. DuPont DeNemours and Company; by the NATO Scientific Affairs Division (0403/87), and by Office of Naval Research contract N00014–89–J–3139 under the DARPA Artificial Neural Network Technology Program. Support for the A.I. Laboratory's artificial intelligence research is provided by the Advanced Research Projects Agency of the Department of Defense under Army contract DACA76–85–C–0010, and in part by ONR contract N00014–85–K–0124.

Contents

Chapter 1

Introduction

The human visual system allows us to identify and distinguish thousands or possibly millions of objects quickly and effortlessly, even in complex, cluttered surroundings and in the presence of partial occlusions. Duplicating this ability with computers has many practical applications. Being able to build artificial systems that can recognize objects may also help us in the formulation and testing of hypotheses about the human visual system.

In this thesis, we will be studying one particularly important aspect of visual object recognition: geometry. By geometric aspects of visual object recognition, I mean those aspects of visual object recognition that are based on the shape of an object, and those procedures that a recognition system (artificial or natural) can use to take advantage of the shape of objects.

Before even embarking on the study of geometry in visual object recognition, a valid question to ask is how related the shape of objects is to their identity. The answer to this question is very complex, and we will defer a discussion until later. For the time being, suffice it to say that shape does play a significant role in many applications (in industry and robotics) of visual object recognition, and that the human visual system has a demonstrable and significant ability to recognize and distinguish objects based on their shape alone.

On the other hand, the use of shape is not a panacea. Many logistic and economic constraints prevent us from measuring shape exactly, and many instances of classes objects vary significantly in their shape from one another. Because of such variations, highly accurate modeling and use of shape is usually neither possible nor advantageous for recognition.

Therefore, for the purposes of this thesis, we consider the problem of taking advantage of shape for recognition in a coarse, approximate, and (as it will turn out) robust way.

Some applications of visual object recognition in industry do require high-precision measurements of shape. However, even for such applications, we will see results that support the notion that the coarse and robust strategies presented in this thesis are a useful first step of processing and solve most of the difficult aspects of recognition. The human visual system also seems to solve some high-precision recognition tasks (e.g., face recognition), and, again, it appears that coarse recognition is an important first processing step.

Research on building systems for visual object recognition has been faced with three major obstacles:

- the correspondence problem
- representations of 3D objects
- robust similarity measures for shapes

Let us examine these in more detail.

1.1 The Correspondence Problem

Given an image, there is a large number of places and a large number of different ways in which an object could match that image. This fact may be surprising to someone who has not actually tried to implement a system for visual object recognition on a computer. We will be discussing the nature of the correspondence problem in more detail in Chapter 2 and review algorithms that have been designed for addressing it.

In Chapter 3, we will discuss a particular algorithm, the RAST algorithm, that addresses the correspondence problem in a novel and particularly efficient way. Unlike previous recognition algorithms, the RAST algorithm performs sufficiently well even in the absence of any correspondence information that it has allowed us to use very simple and robust representations of images.

1.2 Representations of 3D Objects

Because a significant part of visual object recognition deals with 3D shape, a natural approach towards 3D object recognition has been to represent 3D shape explicitly and use such representations for the recognition of objects in images.

Unfortunately, even apparently simple 3D objects are very complicated mathematical objects. Describing their structure formally and in a way suitable for algorithmic processing has been a very hard research problem on which significant progress has only been made recently.

Because of this, many recognition systems have used heuristic simplifications for the representation of 3D shapes. In particular, the appearance of an object in an image in relation to its shape can be divided into two main components: aspects and metric properties. The aspects of an object describe what is visible from a particular viewpoint and what is occluded by the object itself (e.g., we cannot see the tail light of a car when looking at the car from the front). The metric properties describe the exact mutual distances and locations of visible parts of an object in an image.

It has generally been accepted for many years that the aspect structure of an object is very difficult to compute and use for visual object recognition. Therefore, many recognition systems have relied on multi-view representations of objects, in which each aspect of an object is represented separately.

Metric properties, on the other hand, have always been considered as being very easily accessible to mathematical description. After all, the relationship between points in 3D and points in an image is described by little more than a matrix multiplication. Therefore, most researchers have attempted to take explicit mathematical advantage of metric properties for recognition.

However, the mathematical description of metric properties is deceptively simple. Once we start taking into account issues like model acquisition, representation of partial knowledge about 3D objects, curved objects, inaccuracies and measurement errors in images, the effects of lighting, and the variability of the shape of objects, the mathematical description of the metric properties of objects in images becomes a very difficult problem. On the other hand, if we ignore these complexities and simply assume that the world consists of rigid objects with rigidly attached features that can be located with high precision in images, our recognition systems will make frequent errors; unfortunately, this is the assumption that many current recognition systems make.

In response to such problems, a few systems have taken a strictly view-based approach: 3D objects are heuristically represented by collections of representations of 2D views. This is the representation of 3D objects for visual object recognition that I will argue for in this thesis. However, unlike previous uses of view-based recognition, this thesis presents theoretical and empirical results that carefully analyze the tradeoffs between computational complexity, accuracy, reliability, and amount of memory required.

1.3 The Similarity Problem

Most of the current research in visual object recognition has been on addressing the two key implementational issues of visual object recognition: the correspondence problem and the question of 3D representations. However, a much more important question than this question of "how" is the question of "what": what do we actually mean by finding an object in an image.

Current recognition systems fail to recognize simple 3D objects reliably even in most seemingly simple scenes.

Many recognition systems fail because they make unsound simplifications of the recognition problem in order to get around the correspondence problem or the representational issues. Examples of such unsound simplifications are the use of extended straight-line approximations to edge maps or the assumption of objects that can be represented using CAD-like models.

But even systems that are careful about such issues (or restrict their domain suitably) are usually still unreliable in the presence of clutter and occlusions. This failure is due to the simplistic similarity measures used in such recognition systems.

Similarity measures used in current object recognition systems generally simply determine "how much" of an object is visible in an image. Some more sophisticated systems assign different weights (or saliency) to different parts of an object or of an image, and others take into account prior probabilities on the presence or absence of object parts.

However, all these similarity measures still do not take sufficient advantage of large amounts of information about objects, the world, and the image. In a Chapter 7, I will give examples supporting this assertion and outline one approach towards developing better similarity measures.

The RAST algorithm and the paradigm of view-based recognition mostly free us from concerns about the implementational issues of visual object recognition and let us finally address the much more fundamental question of similarity measures.

Chapter 2

Bounded Error Recognition

2.1 Introduction

In this section, we examine in detail the problem of recognition under bounded error.

Bounded error models are interesting because they are amenable to complexity analysis, and they are robust models of error for real applications (see Norton, 1986, for a general discussion). In many applications, we are only concerned with the question of whether some value does not exceed certain tolerances, but the exact error is unimportant. Bounded error models also tend to be more robust and easier to apply than estimation techniques based on more specific models of error.

We define the problem of *object recognition under bounded error* as follows. Let images and models consist of features (e.g., points). Find a transformation (e.g., among all translations) that will map as many points of the model to within given error bounds of image points.

The following are key concepts in understanding bounded error recognition:

- **features** Features are discrete, localizable properties of images. Common features include vertices, edges, lines, corners, intersections, blobs, etc.

- **transformation space** Abstractly, we can view object recognition as the problem of identifying an equivalence class of feature locations under some geometric transformation. The set of all geometric transformations that we allow is called transformation space.

- **evaluation functions** Real images contain noise, occlusions, and context. The

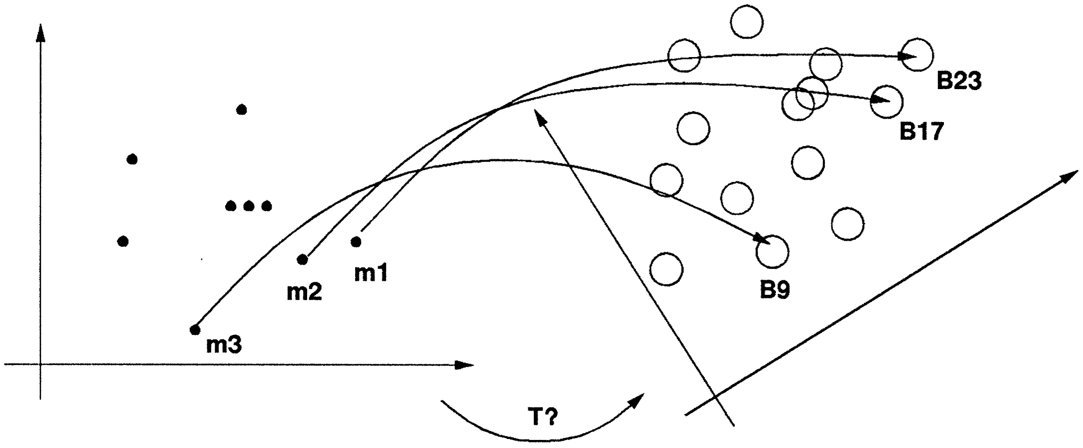

Figure 2-1: A formalization of the recognition problem with bounded error: Find the largest subset of points m_i on the left such that there exists a transformation T (translation, rotation, scale) of the plane that maps the points into the error bounds $B_j = b_j + E_j$ given by image points b_j together with error bounds given as sets E_j on the right.

presence of these means that we cannot expect that an image matches a model perfectly, even if we choose the same geometric transformation that gave rise to the image in the first place. To quantify how well an image is matched by a given object transformed by some given transform, we use an evaluation function that returns an indication of the quality of match between the image and the model.

- **verification of matchings** Under a bounded error model of recognition, picking a particular transformation will bring only some object features in correspondence with image features under the error bounds. The collection of pairs of image features and object features that are brought into correspondence by a transformation is its corresponding matching. The verification problem for matchings is to determine whether there exist a transformation consistent with some given matching.

Evaluation functions find applications primarily in transformation space based recognition algorithms, while verification of matchings is used in depth-first search algorithms (see below).

2.2 Features

For the purpose of this thesis, a *feature* is a localizable property of images. Common examples of features are edges, ridges, lines, arcs, vertices, weighted averages of point feature locations, centroids of image regions, and specularities.

Using features as the basis for recognition has two major advantages:

- it normalizes the input data with respect to many kinds of changes in illumination
- it greatly reduces that amount of information that needs to be processed by the recognition algorithm

Despite these advantages, the use of features for recognition implies a kind of *early commitment*: if the initial stages of processing (feature extraction) fail to detect a feature in the image, subsequent processing cannot recover the missing information easily. Some authors (e.g., Bichsel, 1991) have argued that this kind of early commitment makes feature-based algorithms too unreliable for practical applications. This may be true for some domains. In many domains, however, feature-based methods do work reasonably well.

It is likely that, ultimately, vision systems will use a combination of feature-based and non feature-based approaches. In this thesis, we will limit ourselves to feature-based methods, mostly for practical reasons. The basic techniques and conclusions, however, can be generalized to non feature-based approaches.

The methods for feature extraction used in this paper are described in Appendix A.

2.3 Error on Location

2.3.1 Sensing Error

A vision system cannot hope to recover the location of features in a scene exactly. Some of the reasons for this fact are:

- Optical systems are imperfect and subject an image to various kinds of convolutions before it reaches the image sensor
- Image sensors themselves are subject to noise and quantization
- Any particular "feature" only exists at certain scales; if we try to examine, say, an edge, at too fine a scale, we will find that it is a complicated curve (see, e.g., Falconer, 1990)
- Edge detectors are sensitive to variations in lighting (this is, in fact, probably the most significant source of sensing errors in current vision systems).

Vision systems must be prepared to handle the resulting uncertainty in feature locations.

We may try to model the sensing error parametrically, for example, using Gaussian distributions (e.g., Wells, 1992). However, experience in other fields has shown that such models are often not very robust and are easy to update only for a few special kinds of distributions.

An alternative approach is the following. Instead of trying to model the sensing error, we simply bound it. For example, instead of saying that the true location of some feature is distributed like a Gaussian around the measured location with variance σ, we say that it must be within a distance of ϵ of the true location, where ϵ is of the order of, say, 3σ. This is called a bounded error model.

Bounded error techniques have proven to be very robust and efficient for identification and control (Norton, 1986) and have been used extensively in computer vision (Grimson and Lozano-Perez, 1983, Baird, 1985, Ayache and Faugeras, 1986, to name just a few). Note that bounded error models differ from statistical models in subtle ways.

First, if we believe that the true error distribution is a Gaussian distribution, then a bounded error model is strictly speaking incorrect, since a Gaussian distribution has tails that extend to infinity. However, if we choose error bounds carefully, this distinction is usually unimportant in practice, since even if the true distribution were a Gaussian distribution, the bounded error model would still be correct in almost all the cases, since it contains almost all the probability mass of the Gaussian distribution.

Second, a bounded error model is not the same as assuming uniform error on location within the error bounds. While it is true that bounded error algorithms are often useful in situations where the distribution of errors can be modeled as uniform, the converse is not true. Assuming uniformly distributed error within error bounds is a much stronger assumption than assuming bounded error, and algorithms based on the assumption of uniform error distributions may not be as robust as bounded error algorithms.

Most researchers implicitly assume that sensing errors for different features are uncorrelated. This assumption is perhaps justifiable because correlations between sensing errors are usually due to incomplete modeling of the geometric aspects of the image formation process. However, a bounded error model does not require such assumptions to be made. Empirically, sensing error appears to be modeled well by a bounded error model.

2.3.2 Model Variation

Another important source of error is model variation. There are several causes of model variation: Real objects, even rigid ones, are subject to slight deviations from their idealized geometric shape. Such errors are similar in character to sensing errors: they are small and bounded, and usually uncorrelated.

In fact, if the transformations that we allow do not include changes of scale, then model variation is often well approximated by error bounds on the location of features in the image; that is, model error will appear. In particular, if bounds on the amount of 3D model error are circular, then these bounds will be transformed into circular bounds by the imaging process (projection).

In the presence of changes of scale, error bounds on the object scale with the model, while sensing error is independent of the scale of the object. However, in practice, this change of scale is limited (because the object must fit into the image sensor–CCD or retina) and, on the other hand, cannot be too small, since otherwise feature extraction and other processing steps will fail. Therefore, it is often appropriate to represent both model and sensing error by error bounds on the location of features in the image.

There are other kinds of model variability that cannot be modeled as sensing errors. For example, some objects are *articulated*, meaning that the are composed of several rigid parts that can move relative to one another under certain constraints. Obviously, a single rigid model with small error bounds is a poor model of this case. However, the assumption of rigidity still applies to the object parts (see Brooks, 1983, Connell, 1985, Ettinger, 1987, Grimson, 1989c). Other objects may be allowed to undergo more general, non-rigid transformations.

2.4 Transformation space

2.4.1 What is Transformation Space?

Apart from the error model, another fundamental question for recognition based on geometry is what kinds of transformations we allow. This question is often answered by considering the process of image formation.

For the recognition of rigid 3D objects from 2D images, objects are usually thought to be able to translate and rotate freely in 3D space. For the recognition of rigid 2D objects (e.g., "flat widgets", laminar or planar objects), objects are allowed to undergo translation and rotation in 2D.

The set of all transformations that we allow the object to undergo is usually referred to as *transformation space* (sometimes also called *pose space* or *registration space*). A point in transformation space is also called a *pose*.

After feature extraction, a transformation, i.e., a point in transformation space, uniquely determines the image of a given model.

It is usually the case that "nearby" points in transformation space give rise to very similar images, and this simple fact forms the basis for most efficient methods of visual object recognition. This additional structure is the reason why the set of all transformations is referred to as a "space" (rather than a "set").

There are several important cases of transformation spaces; they are listed in Figure 2-2.

Note that the case of 3D recognition from 2D images is different in some important respects from the case of 2D recognition from 2D images (or, "2D recognition") because of the fact that the transformation of the object is followed by orthogonal or central projection. Because of this projection, some information about the object is irretrievably lost, and an infinity of different objects can give rise to the same image, even in the absence of errors.

The complete view of error bounds and transformations in the 3D case is illustrated schematically in Figure 2-3. An object is considered an equivalence class (under 3D rigid body transformation) of feature locations in 3D together with an error model. A particular element of that equivalence class together with model error is a *scene*. The scene becomes an image through a projection and the addition of sensing error.

As we stated above, a point in transformation space determines the image that corresponds to a given model. However, we can turn this around and ask: given some

	Dim	Problem	Applications	Comments
T2	2	2D translation	OCR, cartoons, speech recognition, primitive for TR2/TRS2	simplest, most efficient case, transformation space is easy to visualize
TR2	3	2D translation and rotation	industrial parts recognition, cartoons	can be reduced to T2 by sampling rotations or derived from TRS2
TRS2	4	2D translation, rotation, and scale	industrial parts recognition, cartoons; solving TR2	linear relationship between error bounds and subsets of transformation space (Baird, 1985)
A2	6	2D affine transformations	laminar objects in 3-space can be modeled exactly as 2D objects under affine transformations (Huttenlocher and Ullman, 1987)	linear relationship between error bounds and subsets of transformation space (Baird, 1985)
TRS3	6 or 7	3D translation, rotation, and scale	recognition of 3D rigid objects	non-linear
A3	12	3D affine transformations	solving TRS3	
TRSN2	∞	2D translation, rotation, scale, and smooth non-rigid deformation	recognition of natural objects	no known efficient recognition algorithms

Figure 2-2: Different kinds of transformation spaces useful for visual object recognition.

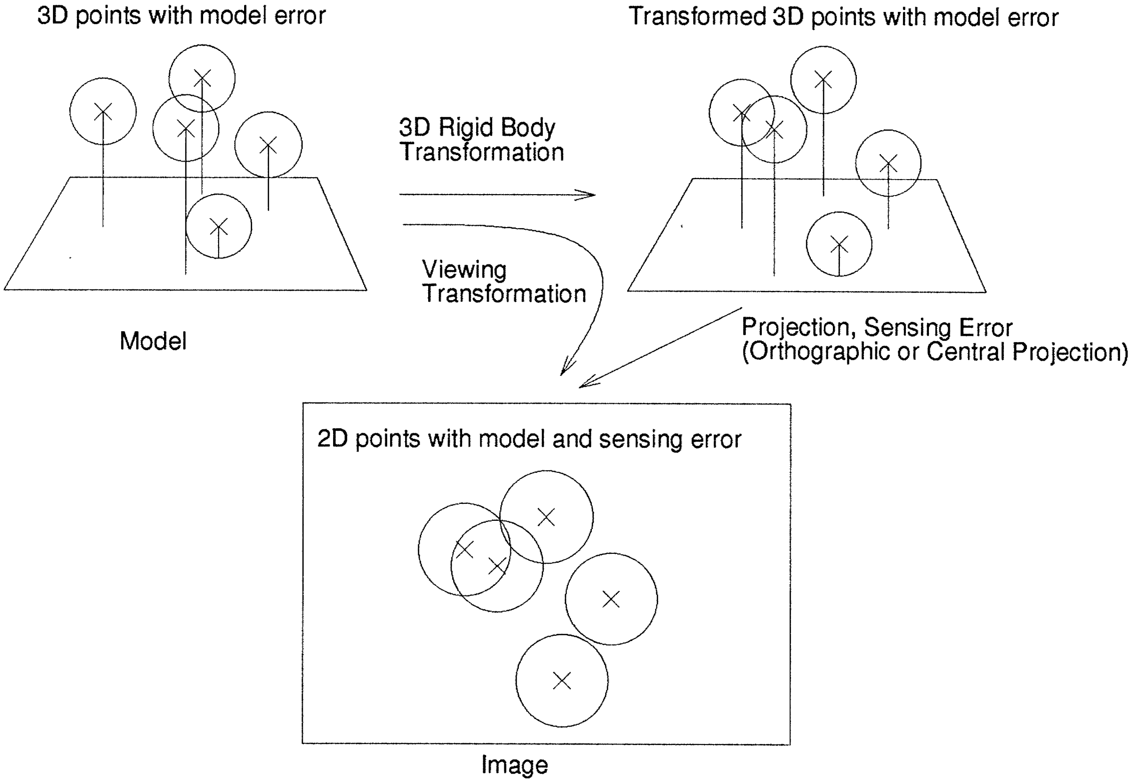
3D points with model error
Transformed 3D points with model error
3D Rigid Body Transformation
Viewing Transformation
Model
Projection, Sensing Error (Orthographic or Central Projection)
2D points with model and sensing error
Image

Figure 2-3: Transformation and errors in 3D object recognition from 2D images.

constraints on the image, what points in transformation space (i.e., what transformations) will transform the model in such a way that the constraints on the image are satisfied.

In particular, for bounded error recognition, the constraints are of the form of a matching, i.e., a collection of pairs (or correspondences) of image and model points. A transformation that is consistent with this matching must map model points to within the given error bounds of their corresponding image points. The set of all such transformations (for a given matching) is the *feasible set* for that matching.

This view of bounded error recognition in transformation space provides the basis for many object recognition algorithms, and for the analysis of many other algorithms (see Section 2.7.1). However, most researchers have used rather unwieldy representations of transformations; if the representation of transformations is not chosen properly, feasible sets can be non-linear or even non-convex. Since most operations in bounded error recognition algorithms require intersections and unions of such sets to be computed, a representation that supports such operations efficiently is very important.

It is known that in several important special cases, simple constraints (error bounds) on the location of image or model features give rise to simple subsets of transformation space. These three special cases are 2D translation, 2D translation, rotation and scale with convex polygonal error bounds on location (Baird, 1985), and affine transformations with convex polygonal error bounds on location. Below, we will review these representations and also discuss how other kinds of transformation space can be represented in terms of these.

Note that some correspondences may be excluded on non-geometric grounds. For example, a red dot on an object can never give rise to a green dot in the image, so no correspondence between them is possible under any transformation. We refer to such non-geometric information as *feature labels.*

2.4.2 2D Translation (T2)

The case of recognition under 2D translation is particularly important, because of its potential applications, because of its efficiency, and because it is easy to visualize.

Assume that image and model points are contained in $\mathbb{R}^2$. Now, consider a particular model point m and a particular image point b. We are interested in the set of all translations T of the plane that map m to within a given error ϵ of the given image point b.

A natural representation of the translation T is as a vector t in $\mathbb{R}^2$. The constraint on T arising from the correspondence between m and b under error ϵ can then be expressed as

$$||T(m) - b|| = ||(t + m) - b|| < \epsilon \tag{2.1}$$

This is illustrated in Figure 2-4.

Note that in this case, error bounds on model points are circular, and that the feasible set in transformation space is also circular. In fact, the 2D translational case is very special because error bounds that take on any shape will give rise to a feasible set of similar shape in transformation space.

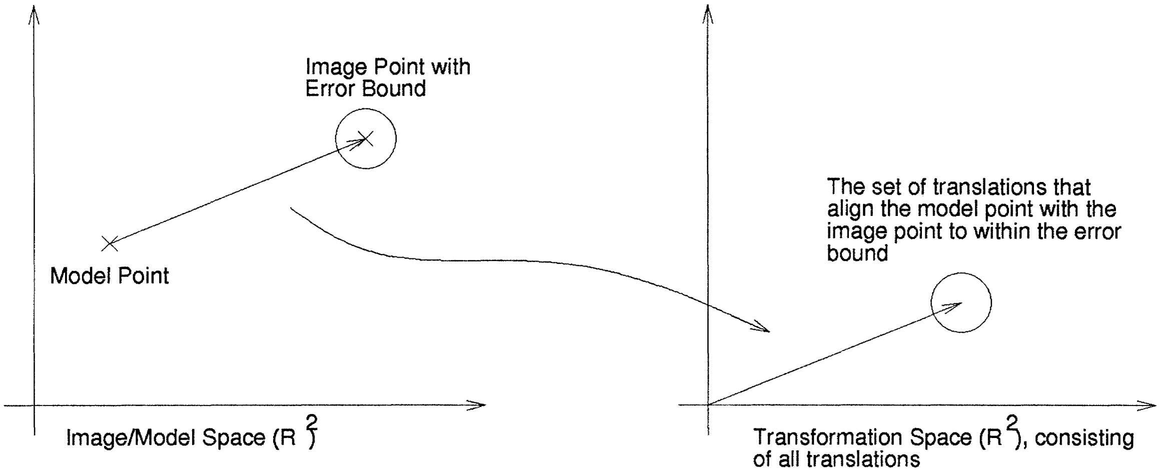
Image Point with
Error Bound
Model Point
Image/Model Space (R 2)
The set of translations that align the model point with the image point to within the error bound
Transformation Space (R 2), consisting of all translations

Figure 2-4: A simple case of transformation space: the space of all translations.

2.4.3 2D Translation, Rotation, Scale (TRS2)

Let us consider the case in which transformation space consists of translations, rotations, and changes of scale (this is sometimes called the set of *equiform* transformations), and in which error bounds are given by convex polygons around model points. The derivations in this section follow Baird, 1985.

Transformations are given by a rotation/scaling R and translation t, i.e., the transformation of the model point m into the image point b_m is given by: $b_m = Rm + t$.

Let us use complex numbers for representing locations. This means that we can represent the rotations and translations themselves as complex numbers. Thus, we can consider the space of transformations either as isomorphic to $\mathbb{C}^2$ or to $\mathbb{R}^4$.

Consider now a point b in the image, and a point m on the model. By pairing the image point b with the model point m under an error bound E, we define a set of geometrically consistent transformations

$$\mathcal{T}_E(m, b) = \{(R, t) : Rm + t \in b + E\} \tag{2.2}$$

This representation has two important (equivalent) properties:

- convex polygonal error bounds E give rise to convex polyhedral sets $\mathcal{T}_E$

- convex error bounds E give rise to convex sets $\mathcal{T}_E$

To see this, assume that error bounds on the location of image points are given as linear constraints. If the linear constraints are given by a vector u and a scalar d, we then require for a transformation that is geometrically consistent with the pairing of a given image and model point that:

$$u \cdot (b - b(r)) \leq d$$

After a little manipulation, for certain kinds of transformations, this can be rewritten as a linear constraint on the components of the transformation:

$$\tilde{C}(u, b) \cdot (t, R) \leq \tilde{d}(d, u, r)$$

Here, (t, R) is a vector formed from the components of the vector representing the translation t and the components of the matrix or complex number representing the rotation R. If t is a vector and R a matrix, then the constraints have the form:

$$\tilde{C}(u, b) = [(u^k)_{k=1,\ldots,m}, (u^i b^j)_{i=1,\ldots,m; j=1,\ldots,n}]$$

$$\tilde{d}(d, u, r) = d + u \cdot r$$

If t is a 2-vector and R is a rotation specified as a complex number, the constraints take the form:

$$\tilde{C}(u, b) = [u^0, u^1, r \cdot u, r \times u]$$

$$\tilde{d}(d, u, r) = d + u \cdot r$$

This is illustrated in Figure 2.4.3.

Observing that convex polygonal error bounds can be written as the intersection of linear constraints, we can therefore state the following theorem:

Theorem 1 *Let $\mathcal{T}_E(b, m)$ be the set of transformations (in TRS2) that are geometrically consistent with the pairing between an image point b and a model point m under a convex polygonal bounded error constraint E. Then $\mathcal{T}_E(b, m)$ is a convex polyhedron.*

A similar result also holds for bounds on orientations of features.

An equivalent result is that if E is a convex set (considered as a subset of $\mathbb{R}^2$), then so is $\mathcal{T}_E(m, b)$ (considered as a subset of $\mathbb{R}^4$).

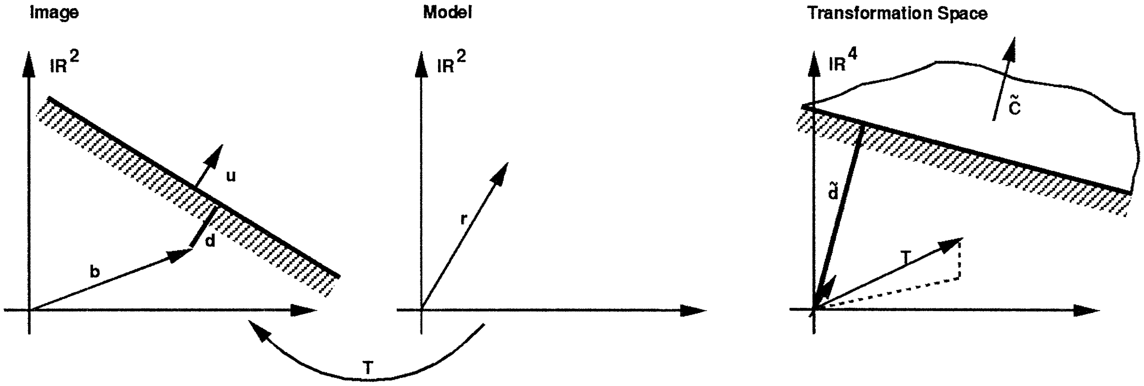

Figure 2-5: Linear constraints on location give rise to linear constraints on transformations.

Theorem 2 *Let $\mathcal{T}_E(b,m)$ be the set of transformations (in TRS2) that are geometrically consistent with the pairing between an image point b and a model point m under a convex bounded error constraint E. Then $\mathcal{T}_E(b,m)$ is convex.*

While this kind of relation between convexity and linearity is actually more generally true, it is still instructive to look at a concrete proof:

Proof. Consider two transformations, (R,t) and (R',t') that are geometrically consistent with the pairing of an image point b and a model point m. We have to show that any linear combination of the two of the form $(1-\lambda)(R,t) + \lambda(R',t')$, where $\lambda \in [0,1]$ is also a geometrically consistent transformation.

$$\begin{aligned} b_m &= ((1-\lambda)(R,t) + \lambda(R',t'))(m) \\ &= (1-\lambda)Rm + \lambda R'm + (1-\lambda)t + \lambda t' \\ &= (1-\lambda)(Rm+t) + \lambda(R'm+t') \end{aligned}$$

But by assumption, $(Rm+t)$ and $(R'm+t')$ are contained in $b+E$. Since $b+E$ is a convex set, for all λ, b_m must therefore also be contained in $b+E$, which proves that $\mathcal{T}(m,b)$ itself is convex. □

2.4.4 The Affine Cases (A2 and A3)

An affine transformation T in d dimensions is given by a non-singular $d \times d$ matrix M_T and a d-dimensional vector t_T:

$$T(x) = M_T x + t_T \tag{2.3}$$

Representations analogous to those given in the previous section also exist for affine transformations; the derivation of this fact is similar.

2.4.5 The "Non-Linear" Cases (TR2 and TRS3)

For several important cases (given in Figure 2-2 as TR2 and TRS3), there are no representations such that linear constraints on location correspond to linear constraints on transformations.

For these problems, most researchers have used "non-linear" representations with a minimal number of parameters (e.g., Cass, 1988a, Grimson, 1990). Such representations make the computation of intersections between feasible sets rather complicated (we will see below why we need to be able to compute such intersections).

A simpler approach is to embed the manifold of all transformations inside a larger space (in the case of TR2, this larger space is the space TRS2, and in the case of TRS3, it is A3). The manifold of transformations is a quadratic surface inside this larger space (Figure 2-6). Now, when we consider correspondences under convex polygonal error bounds, feasible sets are represented by the intersection of polyhedra in the embedding space with the quadratic surface.

2.5 Evaluation of Transformations

Let us now go back for a moment and consider a single transformation (i.e., a pose or a single point in transformation space); the goal of a recognition algorithm is to find an "optimal" transformation. Let us think about what we mean by "optimal".

Together with a model, any transformation determines the locations that model features will assume in the image (Figure 2-7). As we can see, some model features will fall within the given error bounds of some image features. But the same model feature may fall within the error bounds of several image features and vice versa.

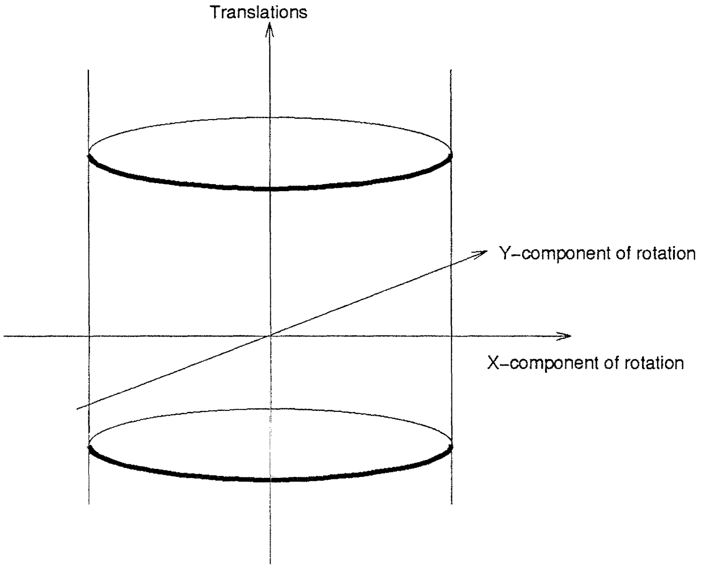

Figure 2-6: The transformation spaces TR2 and TRS3 are quadratic surfaces that can be embedded in TRS2 and A3. Shown in the figure is TR2 as a subset of TRS2, where the two components of the translation have been collapsed onto the Z-axis.

Cass, 1988b, has pointed out that this relationship can be represented conveniently in the form of a bipartite graph; for any given transformation, all the information we need to evaluate a match under a bounded error criterion is summarized in this graph.

How we evaluate this graph depends on the nature of features and scenes that we observe and the kind of recognition problem we are trying to solve.

A particularly common view is that there should be a one-to-one correspondence between features on the object and features in the image.

This can be justified as follows. Consider, for example, the vertices of a cube, in the image of an unoccluded general view, there will be four vertices visible at occlusion boundaries, and one vertex internal to the 2D image of the cube. These should be represented distinctly in the image. Furthermore, it is intuitively appealing to call "better matches" those images in which more of the cube's vertices are visible (see Chapter 7 for a more detailed discussion of these issues).

Under such a model of image formation, we therefore use an evaluation function for a given transformation that computes the largest number of correspondences between

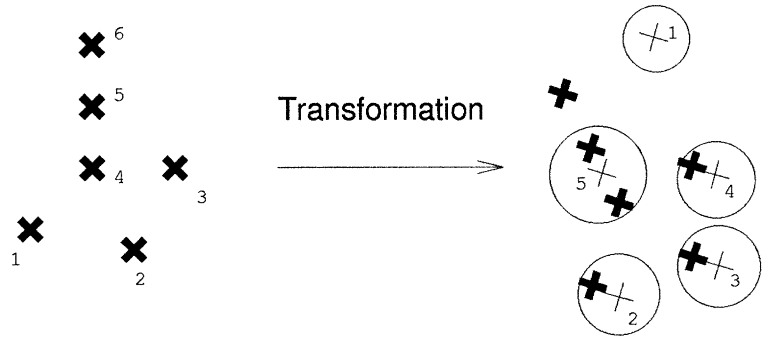

Figure 2-7: The relationship between transformations and matchings.

image and model points that can be established without using either a model or an image feature twice ("unique correspondences"); formally, this idea can be expressed as a *maximal bipartite graph matching* (Cass, 1988b).

The computation of maximal bipartite graph matchings, however, while of low-order polynomial time complexity, is not cheap. In fact, there are a number of reasonable and cheap alternatives to requiring unique correspondence between image features and model features. Here is a list of commonly used evaluation functions:

1. **total-correspondences**: the quality of match is the total number of correspondences between image and model points (this is the quality of match measure usually used by the Hough transform)

2. **unique-correspondences**: the quality of match is the maximal number of correspondences between image and model points that can be established without using any image or model point twice (this is the quality of match measure usually used by correspondence search-based recognition)

3. **min-model**: the quality of match is the number of model features that are within the error bound of some (suitable) image feature

4. **min-image**: the quality of match is the number of image features that are within the error bound of some (suitable) model feature

5. **min-min**: the quality of match is the minimum of the quality of match computed in items 3 and 4

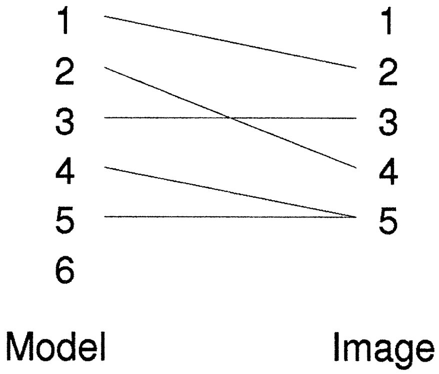

Figure 2-8: The transformation in Figure 2-7 is represented by this bipartite graph.

6. **monotonic**: the quality of match is some other monotonic function of the bipartite graph ("monotonic" in the sense that adding edges to the bipartite graph does not decrease the quality-of-match measure)

For the empirical evaluation of the complexity of the recognition algorithms presented in subsequent chapters, we have used the min-model and min-min evaluation functions. In many cases, the min-min evaluation function is, in fact, equivalent to the unique-correspondences evaluation (this observation has also been made in Huttenlocher and Cass, 1992).

These evaluation functions do allow different model features to match the same image feature or different image features to match the same model features. Other systems (e.g., Grimson, 1990) also allow these kinds of matches as a means to deal with edge fragmentation arising from occlusions.

For the applications in this thesis, allowing multiple matches of a single feature can be justified by the nature of the features themselves. Features derived from the fixed-scale polygonal chain approximations used in this thesis (see Appendix A), are not preserved under changes of scale, unlike, say, a vertex; changes in sampling and matching at different scales causes multiple chain segments in the image to correspond to a single chain segment on the model and vice versa.

Experience with such evaluation functions shows that they seem to work particularly well for industrial parts recognition problems. However, they are based on implicit independence assumptions are work poorly for objects that have less-well defined geometrical shapes; we will examine these issues more closely in Chapter 7.

For more complex recognition tasks than recognition of industrial parts (i.e., tasks in which objects exhibit significant variability, and in which realistic occlusions are present), we will argue in Chapter 7 that the basic model of image formation that underlies the assumption that counting the number unique correspondences is a good measure of the quality of match between a model and an image is inappropriate. Therefore, more general evaluation functions are needed. Some of them will be introduced later. While these measures are more complex, they still satisfy the monotonicity condition (see item 6), which simplifies the recognition problem.

2.6 Verification of Matchings

Above, we looked at the question of what correspondences are implied by a particular transformation (pose). Complementary is the question of whether, given a matching (a set of correspondences) between image and model points, there exists a transformation that is *consistent* with the matching. Consistency here means that the transformation maps each model point to within the given error bound of its corresponding (in the sense of the matching) image point.

The set of transformations consistent with a given correspondence or with all the correspondences contained in a matching is called the *feasible set.*

The problem of verifying a matching is then the following. Each correspondence in the matching implies a feasible set. A transformation that is in the intersection of all the feasible sets is consistent with each correspondence. Hence, we verify a matching by testing whether the intersection of its feasible sets is non-empty.

We have seen above that in several important cases, polygonal error bounds on location give rise to polyhedral feasible sets. Then the problem of determining whether the intersection of a number of polyhedra (represented as intersections of half spaces) is non-empty is a simple form of linear programming. Baird, 1985, was the first to discover this fact and use it for the verification of matchings.

Using the result of Megiddo, 1984, on the linear time complexity of the linear programming problem independent of dimension, this means that we can carry out verification in time $O(n)$ independent of the dimension of transformation space (where n is the number of correspondences).

There is a simple related result. For a collection of convex feasible sets (though not necessarily polyhedral), we can determine whether they have a non-empty intersection using Helly's theorem (cited here from Edelsbrunner, 1987):

Theorem 3 (Helly) *Let $S_1, S_2, \ldots, S_n$ be $n \geq d+1$ convex sets in $\mathbb{R}^d$. If any $d+1$ of the sets have a non-empty common intersection, the common intersection of all sets is non-empty.*

In the case of 2D translation, rotation, and scale (TRS2), this means that given only a boolean function that tells us for any set of five correspondences between image model features whether they are geometrically consistent, we can determine whether a set of an arbitrary number of pairings is geometrically consistent by checking all the subsets of size 5. Since we can test for geometric consistency of a set of 5 correspondences using an $O(1)$ algorithm (in an arithmetic model of complexity), this means that for arbitrary convex algebraic error bounds of bounded degree, we can carry out verification in time $O(n^5)$.

2.7 Recognition

At the beginning of this chapter, we defined recognition as the problem of finding a transformation that brought a "maximum" number of model and image features into correspondence under given error bounds, and in Section 2.5, we have made more precise what we mean by this.

There are two basic strategies for organizing the search for such an optimal transformation: those based on correspondences, and those based on transformation space.

As we saw above, picking a correspondence between a model feature and an image feature gives rise to a feasible set of transformations. If we consider all correspondences between M model features and N image features, we obtain MN feasible sets. These sets partition transformation space into a finite number R discrete regions; for all transformations within each such region, the set of correspondences implied by that transformation remains constant, and hence we can assign to each such region a quality-of-match measure (see Section 2.5). Furthermore, any particular recognition problem under bounded error is completely represented by the collection of matchings corresponding to such regions.

In the language of computational geometry, the structure generated by the intersections of the collection of feasible sets is called an arrangement (see Edelsbrunner, 1987).

For each of the cells in this arrangement, an evaluation function based only on correspondences will be constant; we can find the optimal answer to the recognition problem by examining each cell and calling the evaluation function (Cass, 1990). For

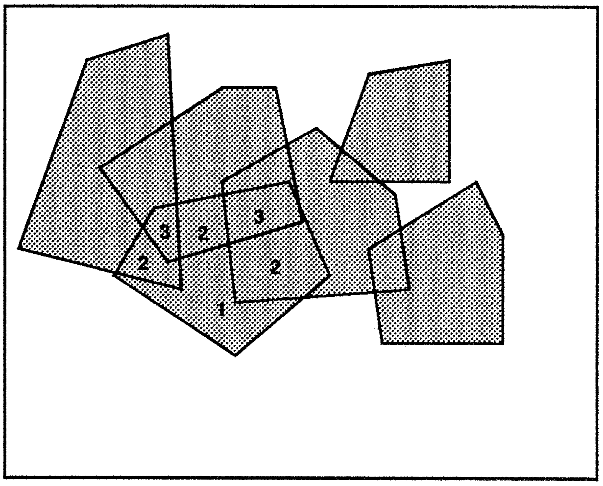

Figure 2-9: A recognition algorithm searches for regions where a maximum number of constraint regions overlap, possibly subject to additional combinatorial constraints.

the usual classes of feasible sets, it is not difficult to establish that the number R of regions is polynomial in MN.

The following is an important simplification of the recognition problem that can be used with monotonic evaluation functions. For monotonic evaluation functions, it can be seen easily that it is sufficient for a recognition algorithm to consider only matchings (i.e., sets of correspondences) such that every possible matching in the arrangement is a subset of one of the matchings considered. The PCS2 algorithm described in Breuel, 1991, takes advantage of this simplification.

Note that it does not make much difference whether a recognition algorithm returns a matching (as correspondence-based methods do), or a transformation: a transformation or pose can be turned into a matching by applying it to the model and actually determining which image and model points satisfy the error bounds, while a matching can be converted into a pose using a (constructive) verification algorithm.

2.7.1 Correspondence-based Methods

Depth-first search methods, alignment, and the PCS algorithm described in this thesis are based on correspondences; such methods consider different sets of matchings and use verification functions to direct the search.

What makes correspondence-based methods potentially inefficient is the fact that such methods are based on a combinatorial view of the recognition problem: a correspondence-based method is faced with finding subsets of image and model points (because of the presence of clutter and occlusions), and correspondences between them, that maximize some evaluation function. Therefore, any simple correspondence-based approach will have exponential time complexity even in the average case.

Correspondence-based methods also seem to be less flexible in the implementation of complicated pose evaluation functions like those discussed in Chapter 7.

Depth-First Search

The basic idea behind using depth-first search for object recognition is to start with an empty matching and non-deterministically add correspondences to it until it becomes inconsistent; the maximal matching obtained in this way is a solution to the recognition problem.

Depth-first search has been used by a number of researchers. (e.g., Grimson and Lozano-Perez, 1983, Baird, 1985, Ayache and Faugeras, 1986), often combined with heuristic methods for speeding up the search. While such algorithms have exponential worst case and (in some cases) exponential average case running times (Grimson, 1989a), they are very popular because they are easy to implement, can take advantage of diverse kinds of geometric and non-geometric constraints, and offer good performance in many practical situations.

Alignment

Alignment (Fischler and Bolles, 1981, Huttenlocher and Ullman, 1987) is a special kind of depth-first search algorithm; effectively, it is depth-first search that has been truncated at a depth that brings the minimum number of image and model points into correspondence needed for determining a unique pose in the error-free case.

In different words, the basic idea behind alignment is to pick the minimum number of correspondences between model and image features necessary to determine a transformation uniquely in the error free case. This transformation is then used to transform the model and to evaluate the match between the model and the image.

Alignment does not solve the bounded error recognition problem exactly. The reason is that there is error on the location of the points used to determine the initial pose. However, an alignment within some error bound ϵ usually implies the existence of a bounded error match with some error bound $c\epsilon$ larger by a constant factor c (see Huttenlocher, 1989, Breuel, 1990, and Grimson *et al.*, 1990). This makes alignment a *weak* recognition algorithm for bounded error (we will examine the effects of weakness in later chapters).

Often, this problem is circumvented by using alignment to establish correspondences between a model and an image, and to optimize the transformation used for the alignment to achieve minimum least-square error.

In practice, alignment seems to be slower than some of the transformation space based algorithms. The reason is that alignment will blindly re-explore geometrically similar situations. That is, even though alignment is substantially weak (i.e., represents only an approximation to bounded error recognition), it does not take sufficient advantage of this simplification to speed up recognition.

Alignment can be speeded up somewhat by using constant-time point location structures. Furthermore, by pre-computing the locations of features in the image for different choices of alignment points, significant constant factor speedups can be achieved. This is the basic idea behind geometric hashing and configuration-based indexing and (described independently in Breuel, 1987, Lamdan and Wolfson, 1988, and Breuel, 1990).

2.7.2 Transformation Space Algorithms

Transformation space algorithms partition transformation space in various ways and use evaluation functions for points or subsets of transformation space to guide the search towards different regions in transformation space. Hough transform methods, template matching, sweeps, and the RAST algorithm (described later in this thesis) are transformation space algorithms.

Template Matching

The simplest form of transformation space based algorithms are template matching algorithms. Such algorithms simply try a large number of transformations, and evaluate the evaluation function each transformation that has been tried.

When the evaluation functions are based on grey level values of individual pixels, such approaches are usually referred to as *correlation* or *template matching.* When the evaluation function is feature based, the approach has been called *transformation space sampling* and *generate-and-test.* But either form can be transformed into the other, and they are based on the same principle.

This approach has several problems. One is that it becomes infeasible when the transformation space becomes high-dimensional, as it usually does when transformations include rotations and scaling, and when models become 3-dimensional. In general, the complexity of template matching is exponential in the dimension of transformation space. Another problem is that solutions to the recognition problem may be very small and may be missed by the sampling scheme.

Hough Transforms

A Hough transform algorithm represents transformation space with a coarse grid of bins. It considers each correspondence between a model feature and an image feature and increments a counter for each bin that overlaps the feasible set implied by the correspondence. Bins that contain large counts at the end are candidates for good matches. Many variations of Hough transforms have been used for object recognition in practice (Ballard, 1981, Stockman *et al.*, 1982, Stockman, 1987, Mundy and Heller, 1990).

There are two main problems with Hough transform methods. The first is geometric in nature: storage limitations usually make it necessary to choose rather large bins. Furthermore, the bins usually have simple, fixed shapes. A second, more significant problem is that for Hough transform methods, the evaluation function implicitly simply evaluates to a count the total number of correspondences that are consistent with a particular set of transformations rather than one of the other evaluation functions describe in Section 2.5. Figure 2-10 shows that this can easily lead to incorrect matches. Note that the reason for failure of the Hough transform is not the quantization of transformation space (as suggested in Grimson and Huttenlocher, 1988), but the choice of evaluation function.

Sweeps

Cass, 1990, has introduced and implemented an algorithm that he calls Critical Point Sampling (CPS). In the language of computational geometry (Edelsbrunner, 1987), this algorithm is a sweep of the arrangement generated by the feasible sets implied by all correspondences between model and image points (see Section 2.7). While guaranteed to find an optimal solution (in fact, all optimal solutions) in polynomial time, sweep algorithms does not have very good average case complexity for the common cases in visual object recognition. It remains to be seen whether taking advantage of

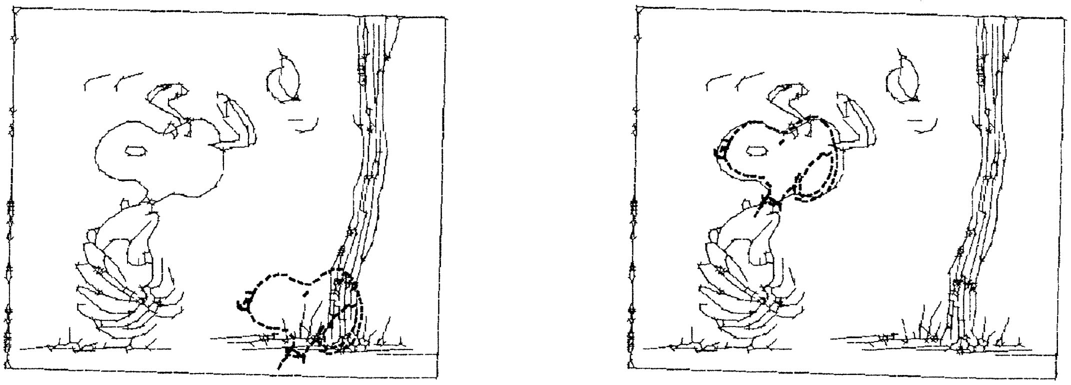

Figure 2-10: The picture on the left shows the optimal match under the Hough transform, the picture on the right shows the optimal bounded error match as found by the RAST algorithm. All geometric parameters for the two matches were identical, but in the Hough transform case, the evaluation function counted the total number of correspondences under the given error bound, while in the bounded error recognition case, the evaluation function counted the number of model features that matched at least one image feature. (Cartoons are taken from Schulz, 1967.)

average-case properties for actual recognition problems is as effective and simple as it is for depth-first search or the RAST algorithm (see Chapter 3).

2.8 The Complexity of Recognition

An important question about recognition is what the theoretical limitations are on the speed of recognition algorithms.

An upper bound on the complexity of recognition is given by the polynomial time recognition algorithms given in Breuel, 1991. Cass, 1990, has presented a similar result based on sweeps. The complexity of such algorithms is determined by the complexity of the arrangement generated by the feasible sets in transformation space.

In the case of polyhedral constraints, there are NM polyhedral constraint sets in transformation space (where N is the number of image points, and M is the number of model points). In the case of 2D recognition under translation, rotation, and changes of scale (TRS2), transformation space is 4-dimensional. Standard results from computational geometry show that the arrangement can be enumerated in time

$\Theta(N^4M^4)$. To this, we need to add the amount of time necessary to evaluate individual transformations, which requires time $\Omega(M)$ using current techniques.

If we define 2D object recognition as the problem of finding all maximal sets of correspondences (all maximal matchings) between image and model points, then a lower bound on the complexity of recognition is given by $\Omega(N^2M^2)$. This is true because the output from the recognition algorithm itself can have size $\Omega(N^2M^2)$ (consider, for example, the trivial case in which there are no solutions larger than 2 points; in that case, the maximal solutions consist of all $\binom{N}{2}\binom{M}{2}$ pairs of compatible correspondences between image and model points).

Since we expect that there are only a few good matches in the image, this bound may not be very interesting. We can relax the problem somewhat by looking at a related decision problem: given N image points and M model points, we ask whether there exists solution to the recognition problem of size at least k, for some given k. Such a definition does not penalize the recognition algorithm for potentially large output.

Even for this simpler decision problem, the lower bound on time complexity seems to be $\Omega(N^2M^2)$ in an algebraic decision tree model, although a proof of this fact appears to be non-trivial.

An question of some theoretical interest is whether recognition problems for which there is a significant gap between the quality of the optimal solution and other, poorer, solutions have a better worst-case bound. For the RAST algorithm (Chapter 3), for the average case, this seems to be true.

The case of 3D recognition from 2D images is similar. However, the set TRS3 of transformations in this case does not give rise to the same kind of linear representations that exist in the case of 2D matching, which complicates the recognition problem somewhat. One approach is to embed the quadratic surface TRS3 in the larger space A3. Using such an approach, we can see that enumerating the possible intersections of feasible sets has complexity $O(N^{12}M^{12})$. A lower bound, on the other hand, is given by the complexity of alignment, $\Omega(N^3M^3)$.

These bounds on the complexity of the recognition problem are of rather high polynomial complexity, in particular for the case of 3D recognition. They provide a strong argument that we must take advantage of other constraints on the object recognition problem in order to obtain practical algorithms.

Chapter 3

Adaptive Subdivision

3.1 Introduction

In the previous chapter, we have reviewed the geometry of the recognition problem under bounded error. In this chapter, we examine a particular algorithm for solving such problems: the RAST algorithm (Recognition by Adaptive Subdivisions of Transformation Space).

Other approaches to the problem (e.g., generalized Hough transformations, Ballard, 1981) are fast, but can miss good solutions because they can take advantage of geometrical constraints only in an approximate fashion. They also expend significant resources on regions of transformation space that could easily be determined not to contain a solution.

Only recently have worst case polynomial time algorithms become available that solve the recognition problem exactly (Cass, 1990, Breuel, 1991). These polynomial time algorithms so far have not yielded truly competitive, practical recognition algorithms, however. Their use has been limited to situations where absolute geometric correctness is more important than speed and occasional suboptimal solutions or mistakes.

This chapter presents an efficient transformation space based algorithm for bounded error recognition. It has low-order polynomial time complexity in the average case even in the presence of spurious data and occlusions, it never loses solutions, and its storage requirements are very modest.

Recall from Section 2.4.3 that assuming a correspondence between a particular image point and a particular model point under a linear error constraint gives rise to a linear constraint on the set of allowable ("feasible") transformations. This observation will be very important for actually implementing the RAST algorithm, since it lets us test quickly whether a constraint set resulting from some correspondence between image and model features has a non-empty intersection with a query box in transformation space.

In a geometric view, the linear constraint on the transformation arising from a correspondence is a half-space in the four-dimensional space of transformations. If we impose several linear constraints on the location of an image point, the set of transformations compatible with a correspondence of a model point with that image point will form a convex polyhedron in transformation space, given by the intersection of the halfspaces corresponding to each individual linear constraint. We will refer to these convex polyhedra as "constraint polyhedra".

Now, consider the set of all possible correspondences between image and model points (there are at most MN of these, where M is the number of model points and N is the number of image points). Correspondences that are compatible with one another, i.e., correspondences for which a transformation exists that allows all of them to be satisfied within the given error bounds, will imply constraint polyhedra that have a non-empty intersection.

The problem of finding maximal sets of correspondences is then the problem of finding the region in transformation space where the largest number of convex constraint polyhedra intersect. The same idea still works when commonly used additional combinatorial constraints are imposed. For example, usually, we count each model point only once, even if it matches several image points. Alternatively, we may wish to associate weights with model points or image points, or we may want to insist on a matching (one-to-one assignment) between image and model points.

Regions of maximal overlap can be very small, and can be missed by coarse sampling techniques like the Hough transform. However, it is possible to bound quickly the maximal overlap of a set of constraint regions inside some query region. For example, assume that we want to determine a bound on the number of convex polyhedra that can intersect inside a given box. A simple upper bound on this number is the number of convex polyhedra that intersect the box itself.

3.2 The RAST Algorithm

We can now formulate a recognition algorithm based on these ideas. The algorithm starts out with a box in transformation space that contains all the transformations

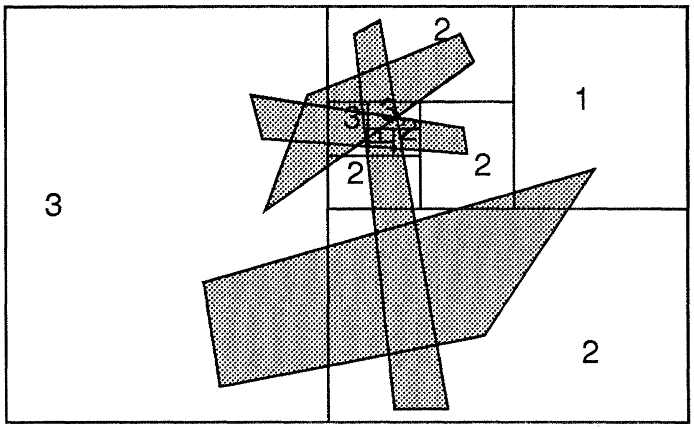

Figure 3-1: Regions in transformation space where many constraints are satisfied can be identified quickly using an adaptive subdivision of transformation space. Each grey polygon represents a region in transformation space arising from a single correspondence between a model feature and an image feature. The rectangles show a subdivision of transformation space computed by the RAST algorithm. The numbers in each rectangle indicate an upper bound on the size of a match in that rectangle.

we would like to consider. It then finds all the correspondences between image and model features that imply convex constraint polyhedra intersecting this box, and evaluates the quality of the match resulting from this set of correspondences; usually this evaluation consists of summing the weight or counting the number of distinct model features represented by the set of correspondences.

If the upper bound on the best possible match is smaller than the minimum quality that we require for a match, or if it is smaller than the best solution found so far, we simply return and abandon the exploration of this part of transformation space. Otherwise, we subdivide the current box into smaller regions and repeat the same process recursively.

Eventually, the boxes will get so small that they either must contain a true intersection of all the polyhedra that overlap the box itself (we can test for this by asking whether the box itself is contained in all the polyhedra that intersect it), or the number of polyhedra that intersect it will fall below the threshold. Pseudocode describing this algorithm is shown in Figure 3.2.

It is interesting to note that this algorithm can be viewed as a kind of multiresolution matching: the RAST algorithm first looks at a large scale for potential solutions and refines and verifies them at successively finer scales.

3.3 Worst-Case Complexity

The worst case running time of the unmodified RAST algorithm is easily seen to be exponential in the problem size. If two constraint regions have large overlap, the improved solution solution contained in their intersection will be found quickly by the RAST algorithm by discovering some box that is contained in their intersection. Only if two constraint regions approach closely but do not overlap, the RAST algorithm may have to expand the search tree very deeply in order to find boxes that fit between the two constraint regions. In principle, such regions of close approach can require a very large number of small boxes to cover them.

Fortunately, this is not a problem in practice, for two reasons. First, on the average, regions like this occur rarely. Second, in practice, we can usually limit the depth of the search tree, effectively changing the recognition algorithm into a "weak algorithm" (see below) and guaranteeing worst-case polynomial time complexity.

3.4 Average-Case Complexity

Let us carry out an *informal* analysis of how an algorithm that works by recursive subdivisions of transformation space will perform on the average. Throughout this section, we will use the symbol "$\mathcal{O}$" to indicate "order of magnitude".

3.4.1 Shape of Constraint Regions

Consider again the case of recognition of a 2D model from a 2D image under translation, rotation, and scaling. As we saw above, in this case, transformation space is 4-dimensional. If image and model features consist of points without associated information about orientation and scale, then the constraint polyhedra in 4-dimensional transformation space implied by their correspondences are cone-like objects with two infinite and two "short" dimensions. If image and model features are line segments that are somewhat larger than the error bounds on location, then a correspondence between individual image and model features constrains translations, rotations, and scaling simultaneously, and the constraint polyhedron in transformation space is a small bounded object. This is the case we will analyze below. For the analysis, we

```
function RAST(constraints,maxDepth,minQuality) =
    let
        bestQuality = minQuality
        bestBox = none
        function searchBox(box,depth,candidates) =
        let
            intersecting =
                all candidates that intersect box
            containing =
                all candidates that contain box
            axis = depth mod 4
        in
            if evaluate(intersecting) <= bestQuality then
                return
            else if candidates = containing
                    orelse depth > maxDepth then
                bestQuality := evaluate(intersecting)
                bestBox := box
                return
            else
                searchBox(left half of box along axis,
                          depth+1,
                          intersecting)
                searchBox(right half of box along axis,
                          depth+1,
                          intersecting)
        end
    in
        searchBox(box containing all transformations,
                  0,
                  constraints)
        return bestBox
    end
```

Figure 3-2: The RAST algorithm. The function `RAST` is called with a list (`constraints`) of all constraint polyhedra, as well as a bound on the maximum depth of the search (or ∞), and a minimum requirement for the size of a solution. The function `searchBox` is given a box (`box`) in transformation space and recursively searches it for a solution to the recognition problem.

will define the quality of a match between an image and a model for a given transformation simply as the number of correspondences compatible with that transformation (i.e., we will impose no combinatorial constraints).

3.4.2 Independence Assumption

We will also assume that the regions in transformation space implied by correspondences between image and model points are distributed independently and "uniformly", and are oriented randomly; this assumption is not strictly satisfied in the case under consideration, since there are MN (M is the number of model points and N is the number of image points) constraint regions, but only $4(M+N)$ parameters determining their location. However, these violations of independence will work for us, since they tend reduce rather than increase the number of intersections between the regions in transformation space.

3.4.3 Small Error compared to Box Size

The RAST algorithm begins by considering a large box in transformation space, much larger than the constraint regions resulting from individual correspondences. The RAST algorithm subdivides this box into smaller boxes until each box contains no more constraint regions than are required for a match of minimum size. As long as the current box in the search algorithm is large compared to the individual constraint regions, the subdivision of transformation space carried out by the RAST algorithm is similar to the construction of a k-D tree, and is therefore approximately $O(n \log n)$ in the number of constraint regions present in transformation space.

3.4.4 Large Error compared to Box Size

Now, let us consider what happens once the linear dimension ϵ of the current box becomes small compared to the linear dimension D of the region of intersection of the constraint polyhedra. We can see that the algorithm in this case will spend most of its time in those parts of transformation space where the surfaces of two constraint regions meet at a small angle. For if they meet at an angle that is large (relative to D and ϵ), the algorithm will quickly find a box that is completely contained in the intersection of the two regions (Figure 3.4.3:2). Assuming random orientations of the constraint regions, and keeping in mind that transformation space is 4-dimensional, the probability of finding two surfaces at such an angle is $\mathcal{O}(\frac{\epsilon^3}{D^3})$ (Figure 3.4.3:3). In

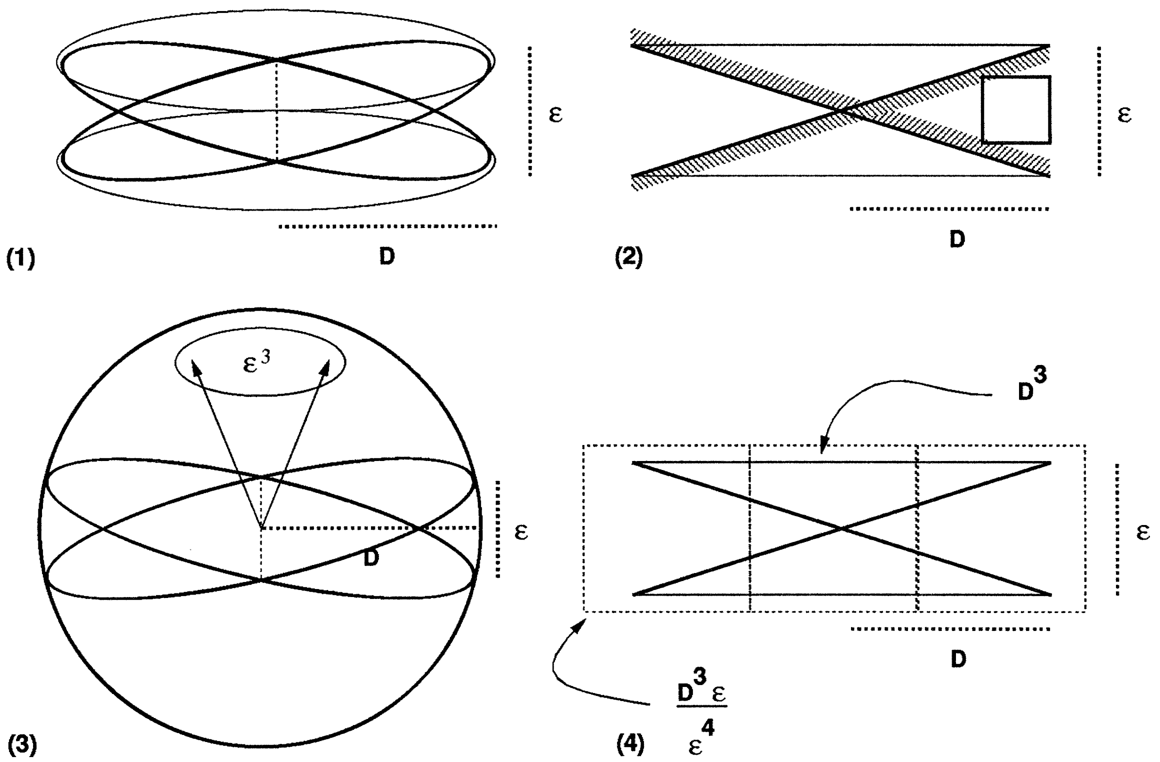

Figure 3-3: (1) Constraint regions in transformation space are bounded by flat faces; let their linear dimensions be $\mathcal{O}(D)$. (2) If the angle between the faces is such that their maximum perpendicular distance is $\mathcal{O}(\epsilon)$ or larger, then we can fit a box of linear dimensions $\mathcal{O}(\epsilon)$ between them. (3) Assuming random orientations, the probability that the surface normals are such that no box of linear dimensions $\mathcal{O}(\epsilon)$ can be fit between the planes is $\mathcal{O}(\frac{\epsilon^3}{D^3})$. (4) To cover the volume enclosed by two such planes, we need $N = \mathcal{O}(\frac{D^3}{\epsilon^3})$ boxes of linear dimension $\mathcal{O}(\epsilon)$.

the worst case, the number of boxes of linear dimension ϵ needed to cover the whole region of approach between the two surfaces is $\mathcal{O}(\frac{D^3}{\epsilon^3})$ (Figure 3.4.3:4). Altogether, the expected number of boxes that are required to cover any intersection between two regions in transformation space is therefore $\mathcal{O}(1)$ (notice that the parameter D has canceled out).

If we assume that there are NM total constraint regions in transformation space, the expected number of intersections is $O((NM)^2)$. However, intersections only need to be considered by the algorithm in regions where more than `bestQuality` constraint regions already overlap, and the fraction of transformation space where this is true becomes very small quickly as `bestQuality` grows, so we expect the quadratic component of the complexity of the algorithm to be small, and diminish rapidly as we set larger minimum thresholds for the weight or quality of a solution.

3.4.5 Higher Dimensions

The above analysis has relied on the assumption that correspondences between model and image features give rise to bounded sets in transformation space. If this assumption is satisfied, then the same analysis applies to higher-dimensional transformation spaces. Of course, in general, for a transformation space of dimension $2k$ and point features, we need k correspondences between model and image features in order to obtain a bounded set in transformation space. For random point sets in the absence of grouping information, this makes the complexity of the RAST algorithm slightly better than that of alignment; for more realistic feature distributions, the RAST algorithm can (and does) perform significantly better, since it automatically takes advantage of non-uniform distributions of features and of imprecise geometric information (e.g., coarse estimates of orientation) that cannot be used by alignment methods.

The RAST algorithm is easily seen to have space complexity $\Theta(NM \log \delta)$, where δ is the dimension of the smallest query region considered by the algorithm. This is in sharp contrast to the Hough transform, which requires space at least $\Omega(NM \left(\frac{\epsilon}{\delta}\right)^d)$ for solving the same problem as the RAST algorithm (using Hash tables to represent Hough space; here, d is the dimensionality of transformation space, ϵ is the linear dimension of the error bounds, and δ is the size of the Hough buckets).

Most implementations of the Hough transform actually solve the simpler problem of only counting the total number of correspondences consistent with some transformation under error (we find that this simpler approach leads to a large number of spurious matches in practice); on the other hand, they use a multidimensional array to represent Hough space. In that case, the space complexity of the Hough transform becomes $\Theta(\left(\frac{L}{\delta}\right)^d)$, where L^d is the volume of transformation space. (For a related analysis, see Grimson, 1988.)

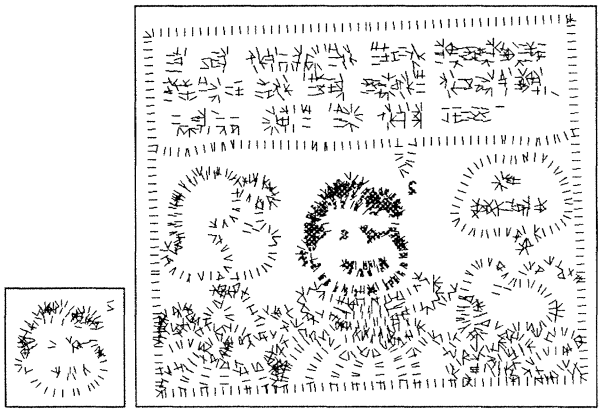

Figure 3-4: Examples of images used for testing the recognition algorithm. The model at the left consists of 150 features, the image on the right, of about 2000 features, and is 560 by 480 pixels large.

3.5 Performance

To gain some practical experience with the performance of the RAST algorithm, it has been applied to two matching problems.

3.5.1 Random Line Segments

Figure 3-7 shows the running time of the RAST algorithm applied to the problem of matching images and models of random line segments under translation, rotation, and scaling. The measure optimized by the RAST algorithm was the number of model features that could be brought into correspondence with some image feature. In each of the test images, half of the model was occluded, and there was a large excess of spurious features ("context") in many of the images. Altogether, the ratio of spurious to model-derived features was as large as 60:1, and the total number of constraint regions in transformation space was 10000.

3.5.2 Cartoon Line Drawings

Figure 3-5 shows the running time of the RAST algorithm applied to the problem of matching a cartoon figure against cartoon line drawings. Features were unlabeled

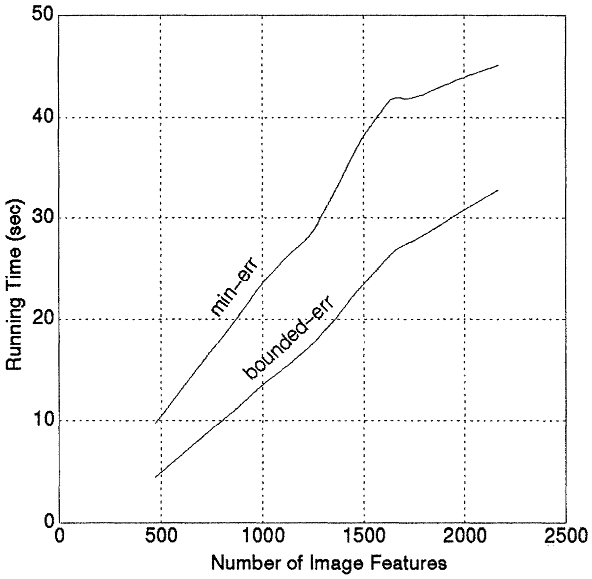

Figure 3-5: Performance of the algorithm on matching cartoon line drawings. A model consisting of 270 features (location and orientation) was used and matched against 192 different images. The recognition algorithm had to find the translation that either maximized the number of matching model features within an error of 10 pixels (bounded-err), or minimized the maximal error when required to match at least 50% of the model (min-err) Running times are in seconds on a SparcStation IPC.

points with their associated local orientation, sampled at regular intervals from the boundaries in the line drawings. The RAST algorithm was used to find the translation that maximized the number of model points matching image points at a given error bound (graph bounded-err).

3.5.3 Comparison to Other Methods

In comparisons with the alignment method (Huttenlocher and Ullman, 1987), the RAST algorithm performed 30 times faster for a model consisting of 166 features (tested for 231 images ranging in size from 434 to 1738 features). The alignment method used a trie data structure for efficient constant time feature lookup in the image; hence its asymptotic complexity is $O(NM^2)$. In contrast, empirically, the dependence of the running time of the RAST algorithm on the number of model features is linear. It is interesting to note that the complexity of the alignment method

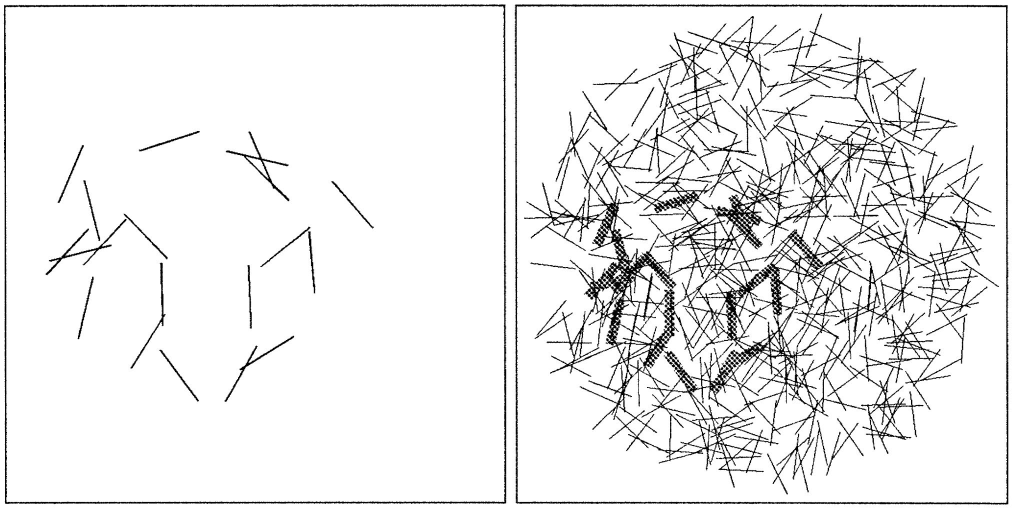

Figure 3-6: Examples of images used for testing the recognition algorithm. The image contains 600 segments in this case, and the model, 20 segments, 10 of which are present in the image.

is asymmetric in image and model size because it can utilize a point location data structure only for either the image or the model. RAST, on the other hand, builds an implicit point-location data structure for both image and model features. In comparisons with the Hough transform method (Ballard, 1981), the RAST algorithm was also slightly faster (but used much less space).

In all cases, the solutions to the bounded error recognition problem found by the alignment and Hough transform methods were significantly worse (in terms of number of features matched) than those found by the RAST algorithm.

Note that both for 2D matching under translation, rotation, and scale, and for 2D matching under translation alone, the RAST algorithm scales well even in the presence of a large excess of spurious features and significant occlusions and in the absence of grouping information.

3.5.4 Finding Solutions with Minimum Error

In a different set of experiments, the RAST algorithm has been modified to minimize the largest error at a given number of matches rather than to maximize the number of matches at a given largest error. We call this the *minimum error* evaluation

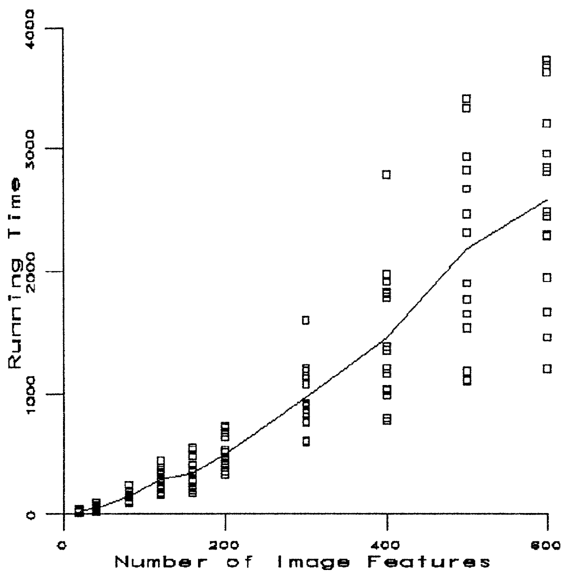

Figure 3-7: Each square represents one execution of the algorithm implemented in C on a Sun SparcStation IPC, time in seconds. The model consisted of 20 line segments, 10 of which were present in each image. The model underwent translation, limited scaling and rotation, and addition of 2% noise. Line segments had a length of 10% of the total width of the input image.

function. Graph min-err in Figure 3-5 shows the performance of the RAST algorithm for solving recognition problems under a minimum error evaluation function; running times marked with "·".

It has recently been argued (Huttenlocher *et al.*, 1991b) that the minimum error measure of similarity between a model and an image is preferable to the more standard evaluation function that counts the fraction of a model accounted for in an image under fixed error bounds. The RAST algorithm is probably currently the best available algorithm for solving problems of this kind.

The basic idea behind using the RAST algorithm for solving recognition problems under the minimum error criterion is the following. First, we parameterize the constraint polyhedra by a scale factor ϵ. Now, during the execution of the RAST algorithm, instead of keeping the error bound parameter constant and updating the `bestQuality` variable, we keep the `bestQuality` variable constant and decrease the error bound parameter ϵ every time a solution at the larger error bound has been found. This way, the RAST algorithm will find the smallest error ϵ at which a match of quality `bestQuality` exists.

Note that we could convert any bounded error recognition algorithm (including the RAST algorithm) into a minimum error recognition algorithm using binary search on the parameter ϵ. However, as some experiments have shown, adjusting ϵ during the execution of the RAST algorithm results in significantly better performance.

3.6 Discussion

The RAST algorithm is a fast algorithm for finding exact solutions to recognition problems under bounded error. As opposed to many previous algorithms for bounded error recognition, it is well-behaved, both theoretically and empirically, even for very large ratios of spurious features to model-derived features in the image.

RAST combines the ideas of the Hough transform, search-based recognition, multiresolution matching, and bounded error recognition, in a simple, efficient algorithm. We find that the performance of the RAST algorithm is better than that of alignment and Hough transform methods. And, as opposed to these methods, RAST finds solutions satisfying simple, well-defined bounded error criteria.

The RAST algorithm is based on the same ideas as voting and clustering schemes like the Hough transform (Ballard, 1981, Stockman, 1987; for a complete recognition system based on the approach, see, for example, Mundy and Heller, 1990). However, because of the dimensionality of transformation space and space limitations, such systems can subdivide transformation space only coarsely and may even have to rely on finding solutions in projections (decompositions) of transformation space. We could use RAST as a highly space-efficient implementation of the Hough transform.

Furthermore, voting schemes like the Hough transform usually only count the number of geometrically consistent transformations in some region of transformation space. The RAST algorithm, in contrast, allows us to use efficiently evaluations of the quality of match that take into account unique matching between model and image features as well as grouping information. In practice, we find that such information is crucial for accurate recognition (it is often only considered in a separate verification step), and that incorporating this kind of information early in the search leads to much faster recognition.

Stockman *et al.*, 1982, also use a subdivision of Hough space to reduce space requirements. Most closely related to the work presented here is the method described independently by Li *et al.*, 1986. Like the algorithm presented here, they use a recursive subdivision of transformation space, and their algorithm does not need to represent all of Hough space explicitly. However, they only apply their algorithm to feature detection, they do not consider the effect of error on location, and they only consider the case in which votes for Hough buckets are generated by linear subspaces. In contrast, the RAST algorithm is an algorithm for the recognition of arbitrary objects. It incorporates an explicit error model for matches between model and image features. Furthermore, the RAST algorithm evaluates matches while searching through transformation space. This is an important difference, since it means that the RAST algorithm can update the `bestQuality` parameter dynamically during the search (particularly important for the computation of the solution under a minerr quality-of-match measure), and that the RAST algorithm accumulates only very little state during the execution.

Compared with algorithms based on correspondences and search (alignment, Huttenlocher and Ullman, 1987, and depth-first search, reviewed in Grimson, 1990), the RAST algorithm has much better time complexity. Furthermore, it can more easily take advantage of geometric constraints with large error bounds for improving its performance; taking advantage of such constraints in a depth-first search algorithm requires constraint propagation, which appears to be costly (Grimson, 1990).

As described above, the RAST algorithm only returns a single, optimal solution. However, it can be modified to return the k-best solutions. A problem with such a formulation of the recognition problem is that it is usually the case that near-optimal solutions cluster (in transformation space) around the optimal solution. Therefore, we usually want solutions that are both well-separated and near-optimal. An inefficient approach to finding the k-best solutions would be to find the optimal solution to the recognition problem and then repeat the search process, excluding a region in transformation space around the optimal solutions already found. Maintaining a priority queue, the RAST algorithm could be modified to compute such solutions more efficiently in a single pass.

An interesting fact about the RAST algorithm is that it can easily be modified to take advantage of relaxed requirements on the matching problem:

- We are only interested in solutions that are better than some minimum quality.
- Suboptimal solutions that are within a given percentage of the optimal solution are acceptable.
- Instead of the optimal solution, the algorithm is allowed to return solutions for a matching problem with slightly larger error bounds (see Grötschel *et al.*, 1988).

Such simplifications of the matching problem are expressed in the shape, arrangement, or evaluation of the constraint regions. Empirically, all of these simplifications of the matching problem result in significant speedups for the RAST algorithm. It is possible to eliminate all floating point operations and multiplications from the inner loop of the RAST algorithm, possibly resulting in significant additional speedups.

The RAST algorithm has already proved to be a very useful tool for matching 2D images in practice. We will see applications for it in Chapters 5 and 6. The RAST algorithm could be used directly for 3D recognition by choosing a 6-dimensional or 12-dimensional transformation space (see Section 2.4.3). However, for reasons we will be discussing in Chapter 4, for the recognition of 3D objects, we are currently using the RAST algorithm for 2D matching in a view-based paradigm.

The RAST algorithm seems to be particularly useful compared with well-established methods for recognition when images consist of a large number of very simple features (e.g., fixed segment-length polygonal approximations to edges). When applied to more conventional problems, in which individual correspondences between features differ greatly in their contribution to the overall quality of match, it may be useful to modify the RAST algorithm to use lazy evaluation for the list of constraints, together with an A*-like heuristic. This would make the RAST algorithm a hybrid between search based methods and transformation space methods, potentially combining the advantages of both approaches.

[illegible]

Chapter 4

View Based Recognition

4.1 Introduction

In this and the subsequent two chapters, we study representations of 3D objects as collections of 2D views. In this chapter, we discuss the theoretical properties of such representations. In Chapter 5, we present simulations comparing the performance of both 3D model-based and view-based recognition algorithms on large databases of objects. Finally, in Chapter 6, we present results of experiments on the recognition of actual 3D objects from 2D images.

A system for object recognition based on visual information is faced with the following problem. The system receives visual input containing different objects. From this input, it has to build internal representations that allow it to detect, distinguish, and identify objects in new images.

There may be many aspects of visual input that help us with recognizing objects; in this chapter, we will study only the geometric aspects. Furthermore, let us adopt the following simplified, formal view of a recognition system.

Let an *image* be a collection of feature locations in $\mathbb{R}^2$, possibly with associated feature orientations and feature labels. These feature locations have been derived in some way from an *object* that may have undergone rigid-body transformations in front of the camera or eye. During an *acquisition phase*, a recognition system receives a collection of training images of 3D objects with associated information about which objects they contain. During a *recognition phase*, the system is presented with novel images and has to determine which of the previously presented objects are present in the image.

Most existing approaches to the visual object recognition problem have concentrated on the recognition phase (see Besl and Jain, 1985, Chin and Dyer, 1986, Grimson, 1990, for surveys). Only recently has there been significant progress in the area of automated 3D model acquisition from 2D views (Longuet-Higgins, 1981, Tomasi and Kanade, 1991, Clemens, 1991).

Thus, the problems of recognition and acquisition have traditionally been considered separately. 3D models have naturally been considered the "interface" between the acquisition and recognition phases. In different words, most workers in visual object recognition have implicitly assumed that there is some process that builds 3D representations of objects in the world, and that this 3D representation is then used by an algorithm to determine whether any given image contains evidence for the given 3D model.

This 3D model based approach has several serious theoretical and practical problems, relating to efficiency, robustness, and some pragmatic issues. We will examine these problems in more detail below.

4.1.1 View Based Recognition

In this chapter, we examine view-based recognition (VBR) an alternative approach to the recognition of 3D objects from 2D images (see Section 4.2 for a discussion of related work). The idea behind VBR is the following.

During the acquisition phase, the recognition system collects and stores different 2D views of the 3D objects it must later recognize. During the recognition phase, the recognition system then compares the given image containing view of the object to be recognized against every stored view of every known object using a 2D recognition algorithm.

The result of each such comparison is a quality of match value (e.g., giving the number of model features that could be brought into correspondence with some image feature under some fixed error bound and 2D translation, rotation, and changes of scale).

The stored view that best matches the given image then determines the hypothesis about the unknown object that the recognition algorithm returns.

The advantages of the view-based approach to the recognition of 3D objects from 2D images will be discussed in the following sections.

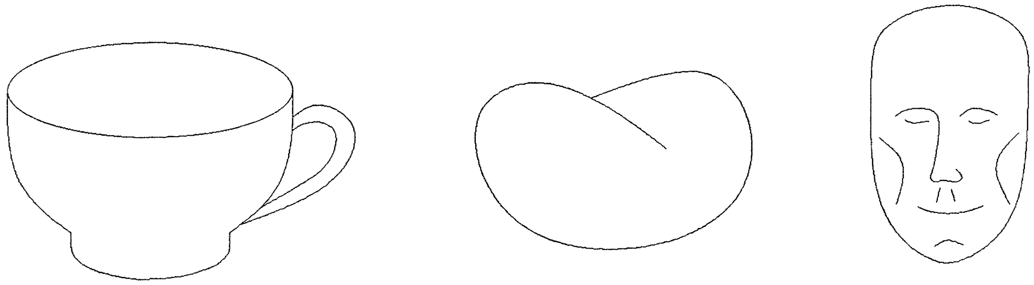

Figure 4-1: Examples of non-attached features.

4.1.2 Robustness

Probably the most significant problem with the 3D-model based approach to visual object recognition is that it is very difficult to model the shape of objects and the process of image formation accurately and generally enough for recognition. Building realistic models for image generation (e.g., by ray tracing) is a difficult enough problem in itself, but recognition actually requires a kind of "inverse" problem to be solved. Specifically, a recognition algorithm has to identify which parts of an object might have given rise to particular features in the image.

The most common assumption about image formation is that the locations of features (like edges or vertices) in an image are determined as follows. Locations of features are assumed to be attached rigidly to the surface of a 3D object, for example, in the form of surface markings or singularities in occluding boundaries (vertices, corners, etc.). When the object undergoes a rigid body transformation, the locations of these features in the image obey the same rigid body transformation composed with the viewing transformation (usually, orthographic or perspective projection).

But there are many kinds of very useful (from the point of view of object recognition) features that do not behave in this way. For example, consider the images of a coffee cup, bean, and face shown in Figure 4-1. In fact, nearly every feature in the images of these objects is not attached to the surface of the object and therefore does not obey the relationship to rigid body transformation that is usually assumed.

Such features are commonly referred to as "unstable features", but there is nothing unstable about them. For example, in the case of the coffee mug, many of those "unstable features" change their location less than features rigidly attached to the surface of the object. We will therefore simply refer to such features as *non-attached features*.

Of course, we can try and model the relationship between 3D rigid body transformations and the location of non-attached features in images. In fact, there has been extensive work on the subject (e.g., Ponce and Kriegman, 1989, Ponce *et al.*, 1991).

However, applying such techniques to actual recognition problems is difficult. There are many different kinds of non-attached features: self-occlusion, highlights, shadows, and curvature extrema on occlusion boundaries are just some examples. And it is not just sufficient to be able to predict where such features will end up in the image given an object shape, we also have to be able to acquire during the training phase, and later represent shapes accurately enough so that the locations of non-attached features can be predicted in images.

This is by no means trivial, since even shapes that are similar in terms of absolute 3D error bounds may differ significantly even in their qualitative behavior for generating non-attached features. Finally, even if the complete theory of non-attached features had been worked out, putting all this knowledge into software would still be difficult.

A view based approach avoids these problems very easily, as we will see below.

4.1.3 Indexing

A further argument for a view based approach is the following. Databases of object models used for real-world recognition problems are large. The human visual system, for example, can probably easily distinguish tens or hundreds of thousands of objects (whatever our definition of object may be). This means that it is very likely to be too inefficient to compare each model in a model base individually against an image. The problem of accessing large model bases efficiently has been called *indexing* (Marr, 1982).

Intuitively, indexing requires that a significant amount of processing must be done in a bottom-up fashion, without explicit reference to the database of models. The reason for this is that if there are N models in the database of models, indexing is going to effective only if the indexing algorithm needs to make reference to some representation derived from the model base that is bounded by a function that grows slowly in N. Note that this does not mean that the representation itself be small in N, but simply that the indexing algorithm access only a small part of the representation for any particular instance of the indexing problem.

Unfortunately, for general 3D shapes, many different 3D objects may have common views (Ullman, 1979; see also Burns *et al.*, 1990, Moses and Ullman, 1991, and Clemens and Jacobs, 1991, for stronger results). Furthermore, the shape of error bounds and their correlation depends on the 3D structure of an object. If indexing were based directly on a simple representation of 3D shape, an indexing algorithm would have to examine each of the N stored 3D shapes in order to discover which views are shared between objects and how error bounds are dependent on viewing parameters. This means that effective indexing algorithms are likely to need to precompute information about how 3D objects share views. A very simple approach to doing this is to represent 3D objects as collections of their 2D views.

4.1.4 Efficiency

As we saw in Section 2.8, using alignment methods, 2D recognition has complexity $\Theta(N^2M^3)$ (where N is the number of image points and M is the number of model points), whereas 3D recognition has complexity $\Theta(N^3M^4)$. A view based approach to recognition has complexity $\Theta(N^2M^3V)$, where V is the number of views per object. This means that 3D object recognition based on 2D views is faster by a factor of $\frac{NM}{V}$ than direct 3D object recognition. Furthermore, in practice, 2D object recognition methods seem to have significantly better constants than 3D methods.

Of course, view based recognition of 3D objects does not answer geometrical questions as precisely as true 3D matching. We will examine this question in more detail in Section 4.4. However, even if we demanded high-precision matching, 2D matching as a pre-processing stage is still a very efficient and useful method for establishing initial correspondences.

Finally, another kind of efficiency to be considered is the amount of effort required for implementing and debugging a recognition system. View based recognition systems are much easier to implement than recognition systems based on 3D shape.

4.2 Related Approaches

4.2.1 Recognition Systems

The idea of representing 3D objects by collections of 2D views is quite old (e.g., Russell, 1921) and has come up frequently in the literature on visual object recognition.

Korn and Dyer, 1987, review a number of recognition systems based on multiple views. Some of these systems have taken an approach very similar to the approach described here: they have computed image properties from a large number of sampled viewpoints (between 100 and 500) and performed recognition by trying to see whether such properties could be identified in images. Examples of such viewpoint specific properties have included silhouettes (e.g., Wallace and Wintz, 1980), visible features (see Section 4.2.4 below), the extended Gaussian image (Ikeuchi, 1981), and edges (Lamdan and Wolfson, 1988). Breuel, 1989, has described a view-based system for indexing that used a particularly compact representation of object views. Stein and Medioni, 1991, have described a similar indexing system that is based on the compact encoding of viewpoint-specific depth information.

Some neural-network based recognition systems (e.g., Edelman and Weinshall, 1989) that in one way or another have relied on view-based representations have also been described in the literature.

This frequent use of multi-view representations for recognition systems is not surprising. The recognition of 3D objects from 2D images does not require any information other than the comparison of viewer-centered information with viewer-centered visual input. Since, as we have seen above, view-based recognition is significantly easier to implement and can even be more efficient than recognition based on 3D models, it is not surprising that even systems that were intended as 3D recognition systems took the shortcuts of view-based representations.

4.2.2 Approximately Invariant Properties

Many researchers have tried to identify and use invariant or approximately invariant properties of objects for recognition.

There are many non-geometric invariants, like color or texture. While such cues are potentially very useful for object recognition, the human visual system has no trouble recognizing 3D objects in the absence of such cues.

There are also many non-metric invariants of objects, such as the topological relation between object parts. Recognition by components (Biederman, 1985) is based to a significant degree on topological relations. Topological and parts relations are likely to be very important for certain kinds of human recognition.

Actual implementations of this paradigm (e.g., Dickinson *et al.*, 1990) rely on parts extraction and on the fact that simple parts can be represented by relatively few views. Villalba, 1990, makes a similar argument for the use of qualitative, view-dependent properties for recognition. However, he claims (without proof) that the use of metric, view-dependent properties for the recognition of 3D objects is "not appropriate", presumably because he assumes that too many views would be needed to represent individual objects. We will see later that that this is not the case.

As far as metric invariants are concerned, several researchers have observed that there are no exact metric invariants even in the error-free case for 3D recognition (Burns *et al.*, 1990, Moses and Ullman, 1991, Clemens and Jacobs, 1991). Furthermore, it is trivial to see that there are no exact invariants even in the 2D case in the presence of bounded error. Moses and Ullman, 1991, additionally prove that even approximate invariants do not exist in general if they are required to apply to the whole viewing sphere.

However, there do exist useful, approximate invariants that apply to large parts of the viewing sphere. This has already been pointed out by Marr and Nishihara, 1978. As an example, they give data that demonstrates that the 2D angles between the projections of the spines of the generalized cylinders that make up simple animal stick figures are quite constrained.

Breuel, 1989, and Breuel, 1990, describe a view-based indexing and recognition system intended for 3D objects. That system used a very compact encoding of 2D views to allow for space-efficient storage and fast retrieval of the number of the multiple 2D views of a 3D object. Breuel, 1990, gives a bound on the viewpoint variation of relative angles and discusses the number of views needed to represent a 3D objects by 2D views. A similar analysis has been made by Burns *et al.*, 1990, and the authors have implemented a recognition system that takes explicit advantage of such approximate invariants. However, trying to take explicit advantage of approximate invariants greatly complicates the actual implementation of their recognition system.

4.2.3 Interpolation among Views

Another approach that has been called "view based" are the method of linear combinations of views proposed by Ullman and Basri, 1989. But unlike the strictly view-based method described here, linear combination of views relies on the geometric nature of the relationship between the attached features of a 3D object, the viewing transformation, and the image.

In fact, if supplemented with the necessary non-linear constraints, the linear combination of views approach can be viewed as building a 3D model of the object for each different aspect; the technique is then equivalent to the model building technique describe by Tomasi and Kanade, 1991. As such, recognition by linear combination of views really follows in the tradition of 3D model based recognition that utilize multi-view representations of the aspect graph.

The assumptions of attached features and affine transformations on which the linear combination of views method is based means that that the method can perform poorly for at least some classes of non-attached features. On the other hand, unless the additional non-linear constraints are enforced (which is costly in practice), the linear combination of views method overgeneralizes. This overgeneralization leads to less-than optimal performance even for the recognition of rigid 3D objects (compared to 3D alignment and strictly view-based methods, for example). (We will see evidence for these assertions in the next chapter.)

Non-linear interpolation schemes like radial basis functions (RBF's; Poggio and Edelman, 1990, Brunelli and Poggio, 1991) can model the behavior of non-attached features under rotation; such systems assume only smoothness of the viewing transformation.

The strictly view-based system described by Breuel, 1989 (like the bounded error methods discussed above), has been described as a covering of the view-manifold (see below) by small, bounded sets. It works for arbitrary view-manifolds of bounded Hausdorff dimension (in fact, the "view-manifold" need not even be a "manifold"). This property is satisfied by most kinds of attached and non-attached features.

The assumption of smoothness made by non-linear interpolation schemes, on the other hand, is a stronger assumption than the assumption about the Hausdorff dimension of the view-manifold. (In fact, assumptions about smoothness are strictly speaking not satisfied for object with non-trivial aspect graphs, due to the discontinuities between aspects.) However, making such stronger assumptions might result in a reduction in the number of views needed to represent an object.

Whether non-linear interpolation schemes actually perform significantly better than strictly view-based systems remains to be seen. The simulations of an RBF-based recognition system described in Poggio and Edelman, 1990, were carried out with a model base of only 10 objects, and they find that they need to store about 80–100 views for each object to achieve recognition of each object with high probability over the whole viewing sphere. In simulations similar to the ones described in Chapter 5, for a model base of the same size, 30 training views give an error rate of 5.6%, and 100 training views give an error rate of 0.7%. The experiments on RBF-based recognition described in Edelman and Buelthoff, 1990a, apply to a two-alternative forced-choice experiment. A strictly view-based approach under the same conditions achieves an average error rate over the whole viewing sphere of 5% using only 5 views, which indicates a significantly better ability of strictly view-based methods to generalize from a single view than either human subjects or RBF-based recognition.

Therefore, while error rates and number of training views are not strictly comparable for these three different simulations (because the objects used in the experiments were slightly different), these results do suggest that the supposed advantage in terms of number of training views needed by interpolation-based methods when compared with strictly view-based methods may be slight (Brunelli and Poggio, 1991, also report that RBF based methods are not significantly better than nearest-neighbor methods in terms of their ability to generalize to novel views; the HyberBF method seems to be able to generalize somewhat better).

Recognition methods based on interpolation (linear or non-linear) have one crucial disadvantage: they require that correspondences be established between multiple training views and the image, and attempting to solve this correspondence problem directly leads to very costly algorithms (with complexities significantly worse than even those of 3D alignment).

We can overcome this problem somewhat by using a strictly view-based approach for establishing correspondences between the image and each of the model views and perform interpolation using the resulting correspondences. This is essentially the approach taken in Chapter 5 for experiments `lcom-3-nearest`. As we will see, such an approach does allow for more precise modeling of the geometry of rigid objects with attached features, but can also make the recognition system less robust, since there is more potential for establishing incorrect correspondences, and since only features that are visible in three training views can be used for matching.

In contrast, for a strictly view-based approach to recognition, correspondences are found efficiently as part of the 2D matching process, for example, using the RAST algorithm, and need only be established between the image and a single model view.

The availability of efficient 2D bounded error matching algorithms that can establish correspondences has allowed us to apply view-based recognition without interpolation to realistic scenes of 3D objects (see Chapter 6).

4.2.4 Aspect Graphs

Aspect graphs (Koenderink and van Doorn, 1979) describe the combinatorial structure of the collection of different views of an object: in the presence of occlusions or non-attached features, different views of an object can differ not only in the location of features, but also qualitatively in the presence or absence of particular features.

Because of the high (even though polynomial time) complexity of known algorithms for computing aspect graphs (e.g., Ponce and Kriegman, 1990), it has been relatively well accepted in the literature that multi-view representations of the aspect graph structure of 3D objects are probably necessary for efficient 3D object recognition from 2D images. Several systems have used deterministic or random sampling to determine the different aspects of an object (see Besl and Jain, 1985, and Chin and Dyer, 1986, for examples), and the number of different aspects of an object has been analyzed theoretically (Ikeuchi and Kanade, 1988).

The question of how the locations of the features of an object behave under 3D transformations is complementary to the question of the complexity of the aspect graph. In fact, the paper clips considered in some of the theoretical analysis and simulations presented here have trivial aspect graphs, since there are no occlusions.

4.2.5 Summary

Almost all previous work on multi-view representations has been based on the premise that a recognition system must take explicit measures to take advantage of the 3D structure of objects lest the storage required to represent objects by collections of their views and the runtime required to match all those views against images become impractical.

In the case of work on approximately invariant geometric properties, such explicit measures have included the identification of specific sets of characteristic features. In the case of interpolation approaches, such explicit measures have been to find linear or non-linear combinations of views.

But taking explicit advantage of the 3D structure of objects has serious disadvantages: as we will see, it tends to make recognition systems less robust and less general. Furthermore, it requires knowledge of correspondences among training views.

In what follows, we will argue that we do not have to use invariants or interpolation for building practical systems for recognizing 3D objects from 2D images.

An important difference between the work on view-based recognition presented here and the simulations and analyses presented elsewhere is that we will explicitly study the effects of large model bases and the presence of noise on view-based recognition, and relate the performance of view-based algorithms to that of 3D model-based recognition algorithms.

4.3 The Number of Views of a Rigid 3D Object

Above, we argued that view-based recognition is an attractive approach to the recognition of 3D objects from 2D images because it handles many deviations from the special case of rigid 3D objects with attached features easily and robustly.

However, while being able to handle deviations from this special case is important for robust, general-purpose recognition systems, the special case of rigid 3D objects with attached features is important and common. Therefore, in this and the next section, we examine theoretically the number of views that are needed to solve the bounded error recognition problem for 3D objects using a view-based approximation, and we discuss the effects of the view-based approximation on the correctness and reliability of a recognition system.

The basic idea behind view-based recognition is that if there is a small change in relative position between viewer and object, the image of the object is not affected much. Since we can look at an object only from a limited set of directions (the technical statement is that the "viewing sphere is compact") and since each view of the object covers a range of directions, we only need to store a finite number of directions in order to cover all possible views of an object when we match under bounded error. We can formalize this idea by considering the modulus of continuity of the viewing transformation, and we can actually give bounds on the number of views needed to represent a 3D object as a collection of 2D views.

4.3.1 The Viewing Sphere

Consider a set of points in $\mathbb{R}^3$ that can undergo rigid body transformations and scaling. Such a transformation is given by 7 parameters: 3 parameters to specify a translation, 3 parameters to specify a rotation, and one parameter to specify scale (recall from Section 2.4.1 that a *viewing transformation* is such a rigid body transformation together with the projection).

Let us assume an orthographic projection model. Then it is easy to see that translation along the projection axis does not affect the projected image of the points. Furthermore, by symmetry, translations, rotations, and scale of the projection of the points in the image plane can compensate for 4 of the remaining 6 parameters describing the 3D pose of the set of points.

This leaves us with 2 rotation parameters (e.g., identifiable with slant and tilt) determining the actual location of points in the projection of the set of 3D points, up to 2D translation, rotation, and scale.

Note that this argument is independent of the relation between features and rotation. In particular, in addition to attached features, it works for all non-attached features whose location only depends on the relative position of the object and the viewer (most non-attached features fall into this category, but highlights, for example, also depend on the position of a light source relative to the object and viewer).

To make this idea more concrete, we can imagine the 2 viewing parameters parameters to be identified with points on a sphere. This sphere is called the *viewing sphere*. We can think of the object as being placed inside the sphere. As we move around the object, looking at the center of the viewing sphere, the optical axis of our eye intersects the sphere, and this point of intersection is a representation of the 2 viewing parameters.

4.3.2 Bounded Error

Above, we defined view based recognition in terms of a 2D recognition algorithm that returns a quality of match measure: the result of a view-based recognition algorithm is simply determined by the best-matching known view for some object.

However, for deriving bounds on the number of views of an object, such a definition is unwieldy, since it makes the total number of views needed to achieve a certain error rate dependent on the entire collection of objects.

Therefore, for the derivation below, we will use a simpler and stricter error measure: bounded error recognition. Under such an error measure, in order to declare a match between an image and a 3D model, we require that all features in some view of the 3D model match the features in the given 2D image to within some error bound δ. This is the most commonly used definition of 3D object recognition from 2D images under bounded error.

Under such a model of recognition, the question of the number of views of an object for a given δ can then be reduced to the question of how much the viewing parameters for an object can change before the location of features in the image determined by the viewing parameters move by more than δ. For each view in the collection of views representing the objects, this set of viewing parameters will correspond to a small patch on the surface of the viewing sphere, and we determine the number of views as the number of views needed to generate enough patches to cover the viewing sphere completely.

4.3.3 Arbitrary Features

Now, if we assume that (for each object) the relationship between the slant and tilt parameters of the viewing transformation and the location of features in the image is piecewise smooth (this is certainly true for attached features and also true for any reasonable kind of non-attached feature), then it will be true that small changes in slant and/or tilt will give rise to only small changes in the location of features in the image.

To get some idea of how smoothly the 2D views of an object change with changes in viewing transformations, we can plot the quality of match of different views of an object for each pair of possible slant/tilt values. Such a plot is called an *error sphere.* We will see examples of error spheres in Section 5.5.9. Error spheres let us draw conclusions about the ability of view based recognition methods to recognize objects from novel viewpoints (to "generalize" to novel viewpoints).

We can formalize the notion of smoothness of the viewing transformation as requiring piecewise uniform continuity over the viewing sphere. To do this, we make use of the modulus of continuity (see Davis, 1975, for more details):

$$w_f(\delta) = \sup_{|x_1 - x_2| \leq \delta} |f(x_1 - x_2)| \tag{4.1}$$

It is not difficult to see that if the maximum modulus of continuity of the viewing transformation is bounded as $\mu \geq \frac{w_f(\delta)}{\delta}$, then each individual view of an object will match a solid angle of approximately $\frac{\delta}{\mu}$ on the viewing sphere.

Note that we can often bound μ by considering the derivative of the location of a feature in the image by the parameters of the viewing transformation.

Since the viewing sphere is a 2D surface, and since we are covering it with patches of linear dimension $\frac{\delta}{\mu}$, we expect that we need $O(\left(\frac{\delta}{\mu}\right)^{-2})$ different patches (and hence, views) of an object. We will derive more specific bounds below.

4.3.4 Bounds for Attached Features

For attached features, it is not difficult to derive more concrete bounds. Recall that attached features are features that behave like the points of a rigid body under rigid body transformations of a 3D object. In this case, the viewing transformation is uniformly continuous over the whole viewing sphere, and we can determine a bound on the modulus of continuity as follows.

Let us assume that translations have already been accounted for.

Then, a viewing transformation consists of a rotation R, a change of scale S, and a projection P: $v_{\text{image}} = PSR(v_{\text{model}})$.

We know that for small rotations (say, of size ϵ) around an axis given by a unit vector r, the displacement of a vector v is given by (Goldstein, 1980):

$$\Delta v = \epsilon\, r \times v \tag{4.2}$$

Since for a unit vector r, it is true that $\|r \times v\| \leq \|v\|$ we know that $\|\Delta v\| \leq \|v\|$. Furthermore, since $P = \text{diag}(1, 1, 0)$, $\|P\, v\| \leq \|v\|$.

Therefore, we see that for any axis of rotation, scale factor s, and small angles of rotation ϵ, the projection of an attached feature v does not move by more than $\epsilon\|v\|$ (this bound actually also works for large ϵ).

Hence, the modulus of continuity of the viewing transformation with respect to any rotation is bounded as $\mu = s\|v\|$. Now, because images are formed on a sensor of finite diameter (retina, CCD array) $s\|v\|$ is bounded by a constant determined by the sensor hardware. So, if we assume that the sensor is bounded by a circle of radius D, then μ is simply D.

The case of perspective projection (weak perspective) is similar. In particular, it is known that the additional error in the location of image features introduced by approximating perspective projection by orthographic project is of the order of a few percent for realistic camera positions.

Note that the resulting bounds on the number of views of an object are independent of the complexity (number of features) of the object. Object complexity does have a slight influence on the number of different views in the presence of occlusions: objects with more features tend to have a larger number of aspects (a bound on the number of aspects in terms of the complexity of an object is given in Ikeuchi and Kanade, 1988).

4.3.5 Deterministic and Stochastic Sampling

Above, we have seen that for an individual view, for smooth viewing transformations, changes in slant/tilt of order ϵ will move the location of features in the image by less than $\epsilon\mu$. Since we require that $\delta \geq \epsilon\mu$, this means that for a given δ, ϵ is at least as large as $\frac{\delta}{\mu}$.

Now, allowing changes in slant/tilt by an amount of ϵ corresponds to an area of α on the viewing sphere:

$$\alpha = 2\pi(1 - \cos\epsilon) \geq \frac{23}{24}\pi\,\epsilon^2 \tag{4.3}$$

(the last inequality comes from the Taylor series expansion of cos).

The viewing sphere has total area 4π. The total number V of circular patches required to cover the viewing sphere, if we could choose their placement, is then bounded (including a factor of 2 to account for the fact that we cannot cover the viewing sphere without overlap using circular tiles):

$$V \leq \left\lceil 2\frac{4\pi}{\alpha} \right\rceil = \left\lceil 2\frac{4\pi}{\frac{23}{24}\pi\,\epsilon^2} \right\rceil \leq \left\lceil 9\epsilon^2 \right\rceil \leq \left\lceil 9\left(\frac{\delta}{\mu}\right)^2 \right\rceil = \left\lceil 9\left(\frac{\delta}{D}\right)^2 \right\rceil \tag{4.4}$$

This is the bound on the number of views of a 3D object under a bounded error recognition model and allowing the view based recognition algorithm to choose the individual views.

The ratio $\frac{\delta}{D}$ is the error that is tolerated by the recognition system relative to the size to the image of the object. In practice, this ratio will be somewhere between 1% and 10%, resulting in an *upper bound* on the number of views of between 90000 and 900. We will see below that the number of views required in an actual view-based system can be much smaller, for example, because of the existence of approximate invariants and because of the presence of characteristic non-metric information (topology, non-geometric information) in images.

In a realistic vision system, we do not get to choose the representative views that represent an object. Rather, we have to cover the viewing sphere stochastically. If we assume that views are uniformly chosen at random, we can apply a variety of results from stochastic geometry. It can be shown that the probability of complete coverage after V trials satisfies as $V \to \infty$ (see Hall, 1988):

$$1 - P_{\text{complete}} = P_{\text{incomplete}} = 1 - (1 + o(1))V^2\rho(1-\rho)^{V-1} \tag{4.5}$$

Here, ρ is the fraction of the surface covered by one patch, i.e., $\rho = \frac{\alpha}{4\pi}$. Note that $P_{\text{incomplete}}$ goes to zero exponentially in V.

Easier to calculate is the probability that a particular view is not covered after V trials, or, equivalently, the expected fraction of the viewing sphere that is not covered by V randomly chosen views. This is simply:

$$E_{\text{error}} = (1-\rho)^V \approx e^{-\rho V} \tag{4.6}$$

4.4 The View-Based Approximation

What we have seen in the previous sections is that if we are willing to accept a certain degree of "approximation" in the recognition of 3D objects, we can use a strictly view-based algorithm for solving this problem. Let us analyze what the nature of this approximation is and how it may affect the reliability of object recognition.

We saw above that, in order to allow an individual view to generalize across a small set of viewing directions, we can simply match it under a small error bound δ. In the presence of error on the location of model features (in 3D) of bounded magnitude δ_e, this means that we must perform bounded error matching under an error bound $\delta + \delta_e$. The case of error on the location of image features (sensing error) is similar, but there are some technical complications due to the fact that sensing error does not scale with the model. This means that a collection of features that are very close to one another in the image could match many models after normalization for scale (because of sensing error). This anomaly is eliminated if we limit the scaling of objects to some "reasonable" range.

Any match outside these new error bounds $\delta+\delta_e$ is then guaranteed not to be a correct 3D match under error bounds δ_e. However, a match within these error bounds might be either a correct 3D match under error bounds δ_e to the correct model, or it might be a false match of another 3D model that happens to match the particular view to within error bounds $\delta + \delta_e$, but would be detected as a false match by a recognition algorithm that solved the 3D recognition problem exactly for the given error bound of δ_e.

Therefore, increasing the error bounds slightly to allow individual views to generalize slightly over a patch on the viewing sphere allows false positive matches to occur. Hypothetically, this might lead to incorrect recognition. However, below, we will argue that the effect of this approximation is no more significant than the effect of many other approximations that have been made commonly and casually in visual object recognition.

In order to understand the effect of these approximations on the recognition problem, we need to understand the structure of 3D point sets under 3D rigid body transformations and orthographic projection.

4.4.1 The Structure of Point-Sets under Projection

Projection of an object consisting of k attached features in 3D gives rise to a view consisting of k feature locations (points) in 2D. We can represent the coordinates of the k feature locations in 3D as a single point in $\mathbb{R}^{3k}$ (object space), and we can represent the feature locations in the corresponding view as a point (the *view-vector*) in $\mathbb{R}^{2k}$ (*view space*).

In the view of 3D object recognition under bounded error described above, we consider objects equivalence classes under 3D rigid body transformations. Since a rigid body transformation is given by 6 parameters, this means that a 3D object is a 6-dimensional surface (in fact, a 6-dimensional quadratic variety) in $\mathbb{R}^{3k}$. Under projection, in general, this 6-dimensional surface will turn into a 6-dimensional surface, the *view-manifold*, in view-space.

In Section 4.3, we have used the fact that certain 3D rigid body transformations (under orthographic projection along the z-axis) can be compensated for directly by 2D rigid motions of the plane. This lets us account for 4 of the 6 dimensions of the surface representing the views of the object in view-space as follows.

It is trivial to see that 3D translations either do not affect the image at all (if they occur along the z-axis), or can be compensated for by 2D translations. To see that one of the three rotational components of the 3D rigid body transformation can be compensated for by 2D rotation, it suffices to write the 3D rigid body transformation in terms of Euler angles (Goldstein, 1980) and choose the last rotation to be around the z- (projection-) axis. That is, after accounting for 2D rigid transformations, the view-manifold is a 2D quadratic surface in the view-space $\mathbb{R}^{2k-4}$.

The view-manifold actually has a number of important properties owing to the fact that it is the projection of the orbit of a single point under the group of 3D rigid body transformations. For our purposes, it suffices to observe that the view-manifold has a relatively simple structure, since it is topologically equivalent to the viewing sphere in most cases.

4.4.2 Error Measures

The above considerations apply to the error-free case. In the presence of error, the view-vector for some view of an object will not lie directly on the view-manifold. Different error measures (bounded error, least square error) correspond to different ways of measuring the distance of the view-vector from the corresponding view-manifold.

In particular, let v be a view-vector, and let the x-coordinates be given by the even components v^{2i} of v $(i = 0, \ldots, 2k-1)$.

The error measure used with bounded error recognition corresponds to the expression:

$$d_{\text{be}}(u, v) = \max_{i=0,\ldots,k-1} \sqrt{(u^{2i} - v^{2i})^2 + (u^{2i+1} - v^{2i+1})^2} \tag{4.7}$$

It is not difficult to see that the bounded error measure is also a metric, which is similar to (but different from) the ∞-metric in view-space. In order to declare a match between an image and a model under bounded error ϵ, we ask that there exist some point on the view-manifold of the model whose distance (under metric d_{be}) from the given view is less than ϵ.

Another common error measure, used with least-square recognition, corresponds to the expression:

$$d_{\text{lsq}}(u, v) = \sqrt{\sum_{i=0}^{k-1} (u^{2i} - v^{2i})^2 + (u^{2i+1} - v^{2i+1})^2} \tag{4.8}$$

This is simply the Euclidean metric in view-space (it is perhaps a little surprising that the distinction between individual features, as well as x- and y-coordinates has completely disappeared).

Under the bounded error model of recognition, we can then reformulate the recognition problem as a point membership problem. The view-manifold for each object is dilated by the set $\mathcal{B}_\epsilon = \{p : d_{\text{be}}(0, p) < \epsilon\}$; if the view-manifold of some model is the set M, we denote this dilated set as $D(M, \mathcal{B}_\epsilon)$. The view-vector v of an image then matches a particular model M iff $v \in D(M, \mathcal{B}_\epsilon)$.

We note that using techniques like those described in Chazelle, 1985, and Clarkson, 1987, such a formulation of bounded error recognition as a set membership problem means that, for large model bases, exact bounded error recognition can be carried out in time $O(\log N)$, where N is the size of the model base. This is an interesting observation, since all previously known algorithms for this problem have had complexity $\Omega(N)$ in the number of models in the presence of error (Clemens and Jacobs, 1991, Weinshall, 1991).

In Section 4.1.1, we described view-based recognition as finding the *best* matching view of some object for a given image, rather than matching under bounded error, as we have considered in the analysis. It is interesting to analyze how this slightly different approach is expressed in view-space.

In view-space, the set of points that will be classified as belonging to a given view-manifold by nearest neighbor classification on samples from the view-manifold will simply correspond to the cells of a Voronoi diagram generated by the individual views. This has implications for the number of views needed to cover the view-manifold. Intuitively, in those regions where different view-manifolds are far apart from one another, only few views are needed for a correct assignment of points to the different view-manifolds, whereas in regions where view-manifolds approach closely, many more views may be needed. This particular intuition about view-based nearest-neighbor recognition of 3D objects from 2D images may be helpful for deriving better bounds on the number of views needed than those we have found in Section 4.3.

4.4.3 View-Based Recognition under Error

Armed with this mathematical intuition about view-space and view-manifolds, we can now analyze the behavior of view-based recognition under error in view-space.

For view-based recognition, each stored view is a sample from the view-manifold M, if there are no errors during training, or from the dilated view-manifold $D(M, \mathcal{B}_{\delta_e})$, if there is error on location during training.

Under a bounded error of $\epsilon = \delta + \delta_e$, an individual view v from the model M will match all views within the translated ϵ-ball $v + \mathcal{B}_\epsilon$. View-based recognition then approximates the view-manifold from the outside using ϵ-balls. (The analysis in Section 4.3 applies to the error free case in which we cover the undilated view-manifold M with δ-balls.) This is illustrated in Figure 4-2.

Note that by covering the view-manifold from the outside, we also allow for some non-matching views to be falsely declared as matches. This is, in fact, the nature of the view-based approximation to the 3D bounded error recognition problem.

Whether these false positives present a practical problem depends on our application. However, we can make several arguments that in most cases, they are likely to be insignificant.

Rejection of Random Images

A very simple and important case for analyzing the effects of the overgeneralization resulting from the view-based approximation is that of the rejection of random images. That is, our vision system confronted either with images containing an actual view of a known object, or with "random" views (see Grimson and Huttenlocher, 1988, for a similar analysis of the Hough transform). We can argue that for a reli-

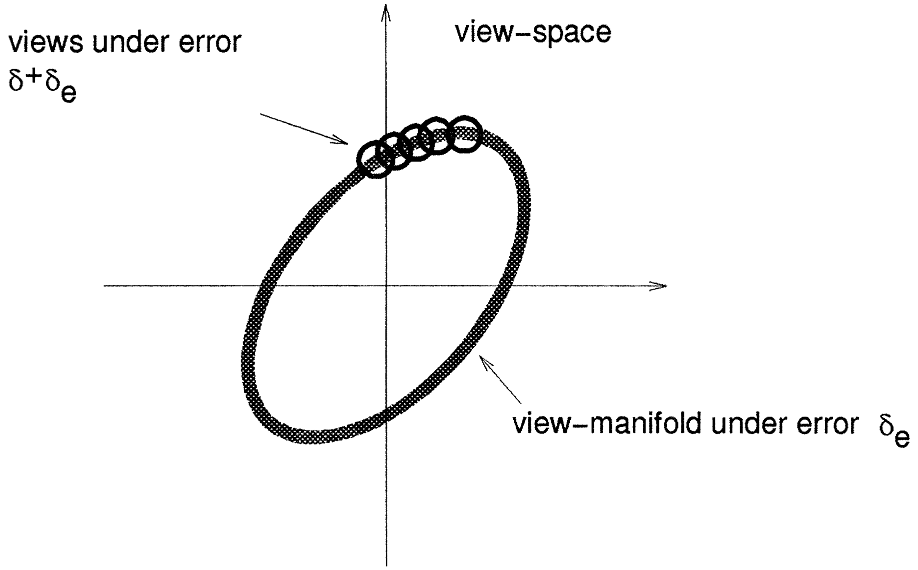

Figure 4-2: Covering the view-manifold with ϵ-balls ($\epsilon = \delta + \delta_e$) centered on individual views.

able recognition system, the probability that a random image is declared to contain a match to a known object must be small.

An important property of bounded error recognition systems is that as the number of dimensions of the view-vector increases, the probability of a random view-vector matching some model under bounded error decreases exponentially. This means that we can use slightly larger view-vectors to compensate for the increased probability of false matches due to the view-based approximation.

To see how much larger we need to make the view-vector (or, equivalently, how many more features we need to use), we first need to compute the probability of finding a random match for the bounded error model and for the view-based approximation to the bounded error model. Under the assumption of "random" views, these questions become questions about the volumes of the dilated view-manifold for the exact bounded error recognition problem, and for the total volume covered by the views in a view-based approximation.

Now, the view-based model, i.e., the collection of ϵ-balls generated by views $v \in M$ that we use to cover the view-manifold, certainly is contained in the dilation of M by ϵ-balls. Thus, the volume of $D(M, \mathcal{B}_\epsilon)$ is a bound on the volume of the collection of ϵ-balls.

For sufficiently small δ and δ_e, the volumes of the δ_e and $\delta + \delta_e$ dilations of M can be approximated in terms of the surface area A_M of M:

$$V(D(M, \mathcal{B}_\epsilon)) \approx A_M \epsilon^{2k-6} \tag{4.9}$$

Here, k is the number of features in the model. The exponent $2k - 6$ occurs because view-space is $2k - 4$ dimensional, and because A_M accounts for the area of the 2D view-surface.

Now, we are asking, given two different values for ϵ, namely $\delta + \delta_e$ and δ_e, for a given k, how do we choose a k' such that:

$$P\left\{v \in D(M(k), \mathcal{B}_{(\delta+\delta_e)})\right\} \leq P\left\{v \in D(M(k'), \mathcal{B}_{\delta_e})\right\} \tag{4.10}$$

Under a suitable assumption of "uniformly distributed" random views, the probability measure in this equation simply reduces to a question about the relative volumes of the dilated view-manifolds for the two different error bounds. Ignoring the dependence of A_M on k, the condition on the probability of error can then be expressed as:

$$(\frac{\delta + \delta_e}{L})^{2k'-6} \leq (\frac{\delta_e}{L})^{2k-6} \tag{4.11}$$

The geometrical factor L makes sure that the dimensions in this inequality are correct (we can only compare powers of dimensionless quantities). It corresponds roughly to the size of the models; hence $\frac{\delta_e}{L}$ corresponds roughly to the error relative to the model size.

After a little manipulation, this reduces to:

$$\frac{2k-6}{2k'-6} \geq 1 + \frac{\log r}{\log \frac{\delta_e}{L}} \tag{4.12}$$

Here, $r = \frac{\delta+\delta_e}{\delta_e}$. For 10% error on location, this means that for error on location $(\frac{\delta_e}{L})$ of 10%, we need to increase $2k - 6$ by about 40%, while for 1% error on location, we need to increase $2k - 6$ by 18% to compensate for the rate of false positive matches due to the view-based approximation. Note that the number of extra features needed decreases as we increase the accuracy of matching.

The tradeoff between using more accurate matching and using larger numbers of features is actually not limited to view-based approximations. In fact, as we will see next, many commonly made assumptions about visual object recognition represent the same kind of approximation that view-based recognition represents. For all these approximations, we can trade off the approximation against using a larger number of features in order to achieve a certain robustness against random positive matches.

Commonly Made Assumptions

Current recognition systems already make a number of approximations and assumptions about the recognition problem that have effects analogous to those of the view-based approximation. Among these are:

- The choice of metric (for example, d_{be} vs. d_{lsq}).
- The assumption of uncorrelated errors.
- The assumption of orthographic projection.
- The use of quantized transformation space (with the Hough transform) or weakness (optional with the RAST algorithm).
- The use of alignments.

Such assumptions are unavoidable: logistic and economic constraints prevent us from modeling the world perfectly.

Now, it is not difficult to see that each of these assumptions has an effect on the recognition problem analogous to that of the view-based approximation: all of them approximate of the true set of view-vectors representing a 3D object by some simpler set.

We have already seen how the use of alignment methods only approximates bounded error recognition. More specificially, in the presence of error on the location of image features of size ϵ, alignment incorrectly generalizes to some bounded error matches under error bounds $c\epsilon$, where c depends on the number of alignment points and the set of transformations under consideration (for 2D matching, $c = 3$, see Section 2.7.1, and for affine matching $c = 5$, Grimson *et al.*, 1990). Note that for alignment, we do not have any control over the amount of classification error introduced by the approximation; on the other hand, for the view-based approximation, we can reduce the amount of recognition error arbitrarily by decreasing the parameter δ.

Another simple case of analyzing this is in comparing our choice of metric. In particular, let us assume that the true metric for the recognition problem is the Euclidean metric (as used with least-square error recognition systems), but we are approximating it with a bounded error metric. Furthermore, assume that our error bound for the Euclidean metric is ϵ, and that models consist of k points (ignore canonicalization for 2D translation, rotation, and scale).

Now, under the Euclidean metric (d_{lsq}) and an error bound of ϵ, if we put all the error into a single coordinate, we can achieve a bounded error of ϵ. If we want to approximate the Euclidean metric with the bounded error metric, and if we do not permit any false negative errors, this dictates that we use an error bound of ϵ for the bounded error matching. However, if we assume a bounded error model with an error bound of ϵ, we have to accept many false positive matches. In particular, the ratio of volumes (equivalent to the ratio in Equation 4.11) is simply the ratio of the (Euclidean) volume of the unit ball of the bounded error metric to the volume of the unit ball of the Euclidean metric. Now, the volume of the unit ball of the Euclidean metric in $2k$ dimensions is given by $\frac{2\pi^k}{2k(k-1)!}\epsilon^{2k}$. The volume of the unit ball of the bounded error metric, on the other hand is simply $\pi^k\epsilon^{2k}$. The ratio of these two volumes is $\pi^k k(k-1)!$ (note that this ratio grows exponentially in $2k$). In general, the same relation holds for dilations of most view-manifolds M with the respective ϵ-balls.

Therefore, we see that the difference between the Euclidean metric and the bounded error metric for evaluating the quality of match is quite similar to the difference between 3D bounded error recognition and the view-based approximation: the volumes of the approximations differ from that of the true dilated view-manifold by a significant ratio, and, furthermore, the magnitude of that ratio grows exponentially in k. Like the recognition error introduced by the alignment method, the effect of our choice of metric on recognition error is fixed and, unlike the effect of the view-based recognition, cannot be compensated for by committing more resources.

Similar considerations apply to the relationship between bounded error matching and Hough transform/transformation space sampling approximations (a related analysis of this can be found in Grimson and Huttenlocher, 1988), and to the relationship between methods that assume uncorrelated error if there is significant correlation between error on the location of features.

The point of the above considerations was to demonstrate that approximations to the recognition problem with consequences analogous to those of the view-based approximation are made routinely in visual object recognition. In fact, the view-based approximation lets us control the quality of the approximation via the δ parameter, while the effects of approximations like choice of metric or the use of alignment are fixed and comparable in magnitude to the case where $\delta = \delta_e$.

The effects of such approximations depends on our application. In the previous section, we saw how such approximations affect the likelihood of randomly occurring matches and how they can be compensated for by using a slightly larger number of features. There may be other effects. For example, if two view-manifolds are very close to one another under a bounded error matching model, then the effect of such approximations can be that the approximations intersect and some views of the two corresponding objects will be confused.

Therefore, there is no reason to believe that the effect of the view-based approximation on the reliability and recognition error of a recognition system is any more significant than that of other, commonly made assumptions. On the contrary, the effect of the view-based approximation can be diminished arbitrarily by using more views, whereas the effects of assumptions like the independence assumption or our choice of metric are fixed. At a given number of views for the view-based approximation, the effects of the view-based approximation on the geometry can also be compensated for by using slightly larger numbers of features for the match, just like the effects of the other, commonly made approximations and assumptions.

Practical experiences support the notion that such subtle differences in the quality of match measure used do not have a significant impact on the reliability of recognition (cf., for example, the small differences in terms of reliability among the different 2D bounded error matching algorithms used in Chapter 5).

4.5 Discussion

In this chapter, we have related the view-based approach to the recognition of 3D objects from 2D images to standard approaches.

View-based recognition represents 3D objects as collections of 2D views and recognizes objects in scenes by finding the best-matching known object view.

The main conclusions from the theoretical analysis are:

- The number of views needed to approximate the 3D structure of a rigid object with attached features (ignoring the structure of the aspect graph) is bounded by $O(\delta^{-2})$, where δ is the approximation parameter.
- The effect of the view-based approximation on the reliability of the recognition of 3D rigid objects with attached features is analogous to the effects of many other commonly used approximations and can be compensated for by using slightly larger numbers of features.

- View-based recognition can deal with any kinds of features for which the relationship between pose and image is a sufficiently smooth (piecewise uniformly continuous) mapping.

In the next two chapters, we will examine simulations and actual applications of view-based recognition that further support the idea that view-based recognition is a practical and useful method for 3D object recognition from 2D images.

Chapter 5

Simulations of View Based Recognition

5.1 A Recognition System

In order to test the theoretical predictions about view-based recognition, a general-purpose, modular simulation system for testing and comparing algorithms for visual object recognition was built. The flow of information in this simulation system is depicted in Figure 5-4.

The system is object-oriented and organized around four basic classes: `Object`, `View`, `Model`, and `Match`.

The `Object` class encapsulates the structure of a 3D object. Views of objects are generated using the `view` and `randomView` methods. Currently, there are two different kinds of objects in the system: `Clip` and `Jumble`.

The `View` class is simply a representation of a single view of an object. It contains a list of features, their locations in the image, and other properties associated with features.

The `Model` class contains implementations of the different recognition algorithms. During the acquisition phase, individual views are added to a model using the `add` method. At the end of the acquisition phase, the `freeze` method is invoked, which allows the `Model` to perform some computation on the stored views before the recognition phase. During the recognition phase, new views are generated and matched by the `Model` class against the stored representation of the object.

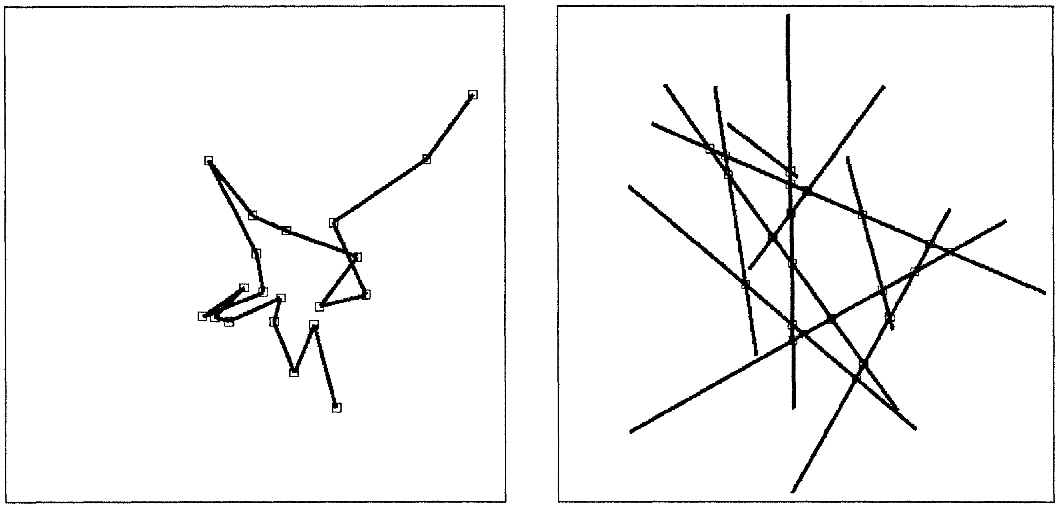

Figure 5-1: Objects and features used in the experiments. Features are marked with a small square.

The output of such a match is an object of class `Match`, which contains the quality of match value and transformation.

The point of describing the simulation system in such detail is that information flow in the system is restricted and passed only through well-defined interfaces, closely paralleling the constraints on information flow in real-world vision systems that learn from examples.

5.2 Large Model-Bases

It is important to compare recognition algorithms on large model bases, since discrimination of only a few objects is a much easier task than recognition from among thousands of candidates. Current computer technology severely limits the size of model bases on which recognition methods can be tested directly. However, for recognition algorithms that return a value indicating the "quality of match" between any given image and model, we can use Monte-Carlo simulations to estimate the performance of the recognition algorithm on a large model base. The basic idea is to compute histograms (empirical distribution functions) of the quality of match values for models against randomly chosen views of the object represented by the model, and against randomly chosen views of randomly chosen different objects; such distribution

`labelerrors`	0	number of mislabeled model points
`noccluded`	0	number of occluded model points
`noise`	20	noise added to the location of each feature in the image
`ntrain`	100	number of training examples
`object`	clip	object used (clip or jumble)
`perfectmodel`	`true`	presence of noise in the training examples
`perspective`	`false`	use perspective projection instead of orthographic
`specificity`	1	number of potential correspondences for each model point
`trainrot`	`false`	sample the viewing sphere deterministically
`trials`	500	number of Monte-Carlo trials

`clip` paperclip-like objects generated by a random walk	`complexity`	20	number of segments plus 1
	`size`	100	size of one segment
`jumble` jumble of line segments in 3D	`complexity`	20	number of line segments
	`size`	300	size of one segment
	`illusory`	`false`	use illusory intersection between segments

Figure 5-2: The table at the top shows parameters affecting the simulator and their default values. The table at the bottom shows parameters affecting the generation of objects and their default values.

`view-bounded-`n VBR using bounded error recognition; quality of match is the number of points that can be brought into correspondence allowing an error of `maxerr`	`minsize`	2	minimum number of matching points
	`maxerr`	n	maximum error allowed
`view-minerr-`n VBR using bounded error recognition; quality of match is the smallest error that under which at least `minsize` points can be brought into correspondence	`minsize`	$\frac{M}{2}$	required size of match
	`maxerr`	100	maximum error allowed
`lcom-random-`v VBR using linear combination of views without view preselection; quality of match is the total least squared error of the points	`ninter`	v	number of views used for the linear combination
`lcom-nearest-`v VBR using linear combination of views and view preselection using bounded error matching; quality of match is the total least-squared error of the points	`ninter`	v	number of views used for the linear combination
	`minsize`	2	minimum size of match for preselection
	`maxerr`	20	maximum error for preselection
`corr-`σ VBR using 2D alignment and gaussians and view preselection; the quality of match measure is described in the text	`sigma`	σ	width of the Gaussian
`alignment-3d` 3D recognition using alignment; quality of match is the total least squared error of the points	`rounds`	10	number of alignment tried
	`points`	3	number of points used for alignment
	`improve`	`false`	find a least-square solution

Figure 5-3: The table shows the different recognition methods used in the experiments, their associated parameters and default values.

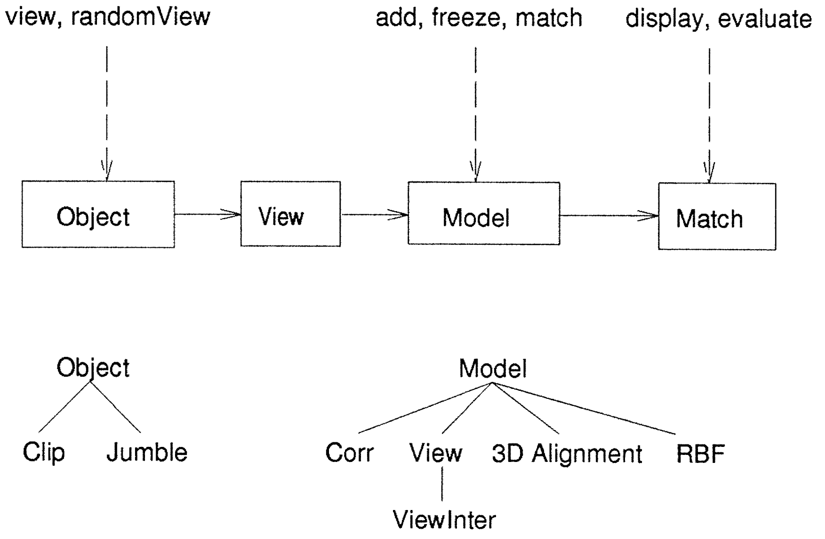

Figure 5-4: The structure of the simulator.

functions are shown in Figure 5-5.

If the true cumulative distribution functions are Φ_1 and Φ_2, then the probability of error for recognition from a model base of size N is:

$$\epsilon = \int \sum_{k=1}^{N} \binom{N}{k} \Phi_2^k(x)(1 - \Phi_2(x))^{N-k}\, d\Phi_1(x)$$

If we replace the true distribution functions with the empirical distribution functions corresponding the the data, we can easily compute an estimate for ϵ (using the exponential bounds for order statistics given in, e.g., Reiss, 1989, we could determine the confidence in this estimate or modify it to obtain weaker bounds with higher confidence). Empirically, this method appears to give good estimates of the frequency of error with only a small fraction of the memory and time required by direct simulations.

5.3 Objects

For the comparison of the recognition algorithms, we used "paper clips" and a 3D "jumble of line segments" as objects (see Figure 5-1).

5.3.1 Paperclips

In the case of the paper clip, the features used were the locations of the endpoints and bends of the clip in the 2D image. These features are *attached*, i.e., they obey the rigid body relationship to the pose parameters usually assumed in the object recognition literature. Paperclips were generated by a random walk (each segment of the clip was 80 units long) and consisted of 19 segments.

Paperclips are essentially simply a collection of 3D points for which correspondences between model and image can be established relatively easily (there is a linear ordering of features, possibly with occasional insertions or deletions in the case of real images). On the other hand, individual features, the locations of bends in the paperclips, carry no non-geometrical information (unlike, say, the vertices of polyhedra, which are labeled by their degree).

Paperclips have been used in psychophysical experiments on object recognition and generalization (e.g., Edelman and Buelthoff, 1990a, Edelman and Buelthoff, 1990b). They have also been used in some other experiments on 3D object recognition from 2D images (e.g., Edelman and Weinshall, 1989, Poggio and Edelman, 1990).

5.3.2 Jumble of Line Segments

In the case of the jumble of segments, the features used were the locations of intersections between pairs of segments in the image. Jumbles were generated by distributing 20 line segments of equal length randomly inside the unit sphere.

This is an interesting case because it is a simple class of objects that has non-attached features (recall that non-attached features are features that do not transform like points on a rigid body under rigid body transformations).

5.3.3 Training

Both kinds of objects were presented with different slants and tilts (no rotation around the optical axis was used). In the experiments reported here, orthographic projection was used, but some experiments with perspective projection at realistic distances of the object from the camera gave qualitatively and quantitatively similar results.

In some experiments, noise was added to the locations of the features; noise vectors with a magnitude less than the noise parameter were chosen from a uniform distribution. In other experiments, the recognition algorithms were not given unique correspondence information; individual features were allowed to match a number of other features given by a specificity parameter.

Collections of 2D views containing feature locations obtained in this way were used as training data for the recognition algorithms. All the error rates shown in the graphs have been estimated for model bases consisting of 1000 objects.

During most of the experiments, the training examples presented to the recognition algorithms were noise-free, and noise on the location of features was only added during the recognition phase (but see Section 5.5.5). Likewise, during most of the experiments, the recognition algorithms received complete correspondence information (but see Section 5.5.7).

5.4 Recognition Methods

We have tested and compared recognition methods that perform 3D matching using alignment, optimal 2D bounded error matching (using the methods described in Section 3), 2D alignment with a gaussian error measure (a method closely related to many neural network algorithms and RBF methods Poggio and Edelman, 1990), and linear combination among two or more views (Ullman and Basri, 1989).

Except for 3D alignment, all methods use a collection of 2D views of any given object to represent the object for recognition. We make a distinction between *strictly view-based methods*, like 2D bounded error matching and 2D alignment, and the interpolation methods, which try to implicitly recover and take advantage of the 3D structure of objects.

As we saw above, each recognition method returns, for each object in the model base, a number indicating the quality of match between image object model.

5.4.1 3D Alignment

In order to obtain a baseline for the performance of the view-based recognition methods, a true 3D recognition algorithm, recognition by alignment (Huttenlocher and Ullman, 1987), was implemented.

The idea behind recognition by alignment is to recover a transformation from the model to the image using a (algebraically) minimum number of correspondences between the image and the model. In the case of recognizing 3D objects from 2D images, we need at least three points for finding a transformation using alignment.

Using this transformation, we align the image with the model. The quality of match is then given as the sum of the squared distances between model points and their corresponding image points.

If there is significant error on the location of one or more of the alignment points, the quality of match found by an alignment algorithm can be significantly worse than that found by an optimal matching algorithm, even though there actually exists a good match between the image and the model. To avoid this problem, alignment algorithms generally try several different collections of alignment points.

The 3D alignment algorithm used the actual 3D object model for recognition. In this, it differs from all the other recognition methods described here, which had to build a 3D object model from examples. Together with a non-linear optimization of the least-square match error between model and image, the performance of 3D alignment then is nearly the optimal performance achievable by any 3D recognition algorithm under ideal circumstances.

Because 3D alignment (as described in Huttenlocher and Ullman, 1987) is only applicable to attached features, 3D alignment was only tested with `clip` objects.

For each match, several randomly chosen subsets (`rounds`) of three points (alignment bases) of image features and corresponding model features were aligned and the model was projected onto the image using the pose determined by the alignment. The quality of match was the minimum of the sum of the squared distances between projected model points and their corresponding image points over all alignment bases that were tried.

Frequently, in applications of 3D alignment, the initial pose estimate is improved using non-linear optimization of the least-square error. This was also tried in the experiments presented here. Doing so does not lead to a significant improvement in recognition accuracy (see Figure 5-7), but considerably increases the computation time.

5.4.2 Optimal 2D Bounded Error Matching

Recognition by optimal 2D bounded error matching compares the image to each view of a 3D model using a 2D matching algorithm. Two kinds of 2D bounded error matching algorithms were used.

2D bounded error matching methods (indicated in the figures as `view-bounded-`n, where n is the error bound) try to determine the rigid 2D transformation that aligns a maximum number of feature points in the image with points in a view from the object model subject to a given error bound. The quality of such a match is given by the number of feature points that can actually be aligned to within the error bounds. The RAST algorithm (see Chapter 3) was used to find optimal bounded error matches under the ∞-norm.

A 2D minimum error matching algorithm (`view-minerr-`n, where n is the number of features required to match) tries to determine a rigid transformation and subsets of given minimum size of feature points in the image and in the view of the model object, in particular, a transformation that minimizes the maximum error. It has recently been argued (Huttenlocher *et al.*, 1991b) that this measure of similarity between a model and an image is preferable to counting the fraction of a model accounted for in an image under fixed error bounds. The overall quality of a match between the image and an object model is the minimal maximum error over all known views of the object.

For both kinds of matching problems, we were using the bounded error recognition algorithm described in Chapter 3.

5.4.3 2D Alignment with Gaussian Error Measure

We have already described above the alignment method for the 3D case. Recognition by alignment in the 2D case uses two points to obtain a 2D translation, rotation, and change of scale.

The usual quality of match measure used with alignment is either a least-square measure or a bounded error measure. In these experiments, we used a slightly different quality measure. Let $\{b_i\}_{i=1,\ldots,n_b}$ be the image points and $\{m'_i\}_{i=1,\ldots,n_m}$ be the corresponding aligned model points. The quality of match is then given by $q = \sum_i e^{\frac{\|b_i - m'_i\|^2}{2\sigma}}$. The effect of this is to remove outliers from consideration and to assign a higher quality to close matches (bounded error recognition, in contrast, makes no distinctions between two matches that satisfy the given error bounds with the same number of matching points but different noise).

In the figures, recognition by alignment is indicated as `corr-`σ, where σ is the standard deviation of the gaussian ("`corr`" because this scheme is also very similar to recognition by "correlation")

Recognition by alignment with a gaussian error measure is related to network models of object recognition such as radial basis functions (Poggio and Edelman, 1990), to

MAP techniques for object recognition (Wells, 1992), and to neural network models in general. It is plausible in terms of the known components of the human visual system: fixation points and textons correspond to alignment points, eye movements and shifter circuits carry out alignments, and the receptive fields of individual neurons for specific features are approximated by gaussian functions.

5.4.4 Linear Combination of Views

Linear combination of three views has been proposed by Ullman and Basri, 1989, as a method for the recognition of 3D objects from a collection of 2D views. This method has also been described as "view based", but it is essentially a simple way to represent and recover a significant part of the 3D structure of an object.

The basic result is that given three views of a 3D object, the x-coordinates of the locations of any view are a linear combination of the x-coordinates of the three given views, and the same result holds for the y-coordinates.

In general, these linear constraints only identify 3D objects up to equivalence classes under 3D affine transformations rather than 3D rigid transformation. Some classes of objects (e.g., pyramids and skew pyramids, or cubes and bricks) cannot be distinguished under affine transformations. It could be argued that these perhaps constitute special cases (analogous to "non-accidental properties" of images), but, as we will see below, the difference between recognition under affine and rigid transformation affects even the performance of recognition of randomly generated paperclips, which have no special symmetries.

In order to distinguish different 3D objects that are similar under affine transformations but not rigid transformations (e.g., pyramids and skew pyramids), additional quadratic constraints need to be imposed.

Tomasi and Kanade, 1991, have described a closely related method for constructing a 3D object model from multiple 2D views of an object by solving first the linear part of the problem followed by a non-linear optimization to approximately satisfy the quadratic constraints under a least-square error criterion.

In the experiments, we used linear view combination of two and three views. In each of these cases, we used view combination between n randomly chosen views (`inter-`n`-random`). The case of `inter-3-random` corresponds to the linear combination of 3 views method described in the literature. Interpolation between 2 views (`inter-2-random`) was simply included as an example of an algorithm that has more limited ability to generalize to novel views.

We also used 2D bounded error matching algorithms to select similar views in a 2D sense before applying the linear combination of views method. The 2D bounded error matching algorithm also provided correspondences and removed feature correspondences from consideration by the linear combination of views method that were outside the error bounds used by the 2D bounded error matching algorithm. Results using this approach are indicated by `inter-`n`-nearest` in the graphs.

5.5 Results

5.5.1 3D Alignment

For choosing parameters for the subsequent tests of view-based recognition, the results from 3D alignment give us some important information.

Figure 5-6 shows the probability of confusing objects (paperclips) of different complexity (i.e., with a different number of segments) under 3D alignment and in the presence of location error (40 units) in the image. We see that even for 3D recognition with a perfect model, in the presence of noise, objects of small complexity cannot be distinguished reliably by any recognition algorithm; at the given noise level, we need objects of approximately 20 segments in order for 3D recognition to even make sense (see Grimson and Huttenlocher, 1989, for a related discussion). This was therefore the object complexity chosen in the experiments.

Figure 5-7 shows the peformance of 3D alignment under different amounts of noise for objects of constant complexity. We see that as the error on location increases (above 40 units), even the 3D model based recognition algorithm cannot recognize objects reliably anymore.

5.5.2 Number of Training Examples in the Absence of Noise

A comparison between the different recognition methods for the problem of recognizing 3D paperclips from 2D images is shown in Figures 5-8 and 5-9.

In the absence of noise, linear combination of 3 views performs perfectly with even very few training examples. This is easy to explain. The linear combination of view method overgeneralizes slightly, since it allows for arbitrary affine transformations of the 3D object, not just rigid transformations. In the absence of noise, this does not affect the reliability of the recognition algorithm because probability that an object matches the image of some other object perfectly is zero. Therefore, with probability one, nonmatching models will not match a given image perfectly, while matching models always give a perfect match. This means that the error rate is almost certainly zero.

When we apply strictly view-based methods to the problem of recognizing 3D objects from 2D images, we find that if we only have a small number of training examples, view-based methods have non-zero classification error. The reason is that it is quite possible that a particular view of a non-matching object model matches an image better in a 2D sense than all known views of the matching object model. However, as we increase the number of views used to represent each object in the model base, the probability of error decreases. As can be seen in Figure 5-9, all view-based methods that were tested behave very similarly in the absence of noise.

5.5.3 Number of Training Examples in the Presence of Noise

Figure 5-10 shows the performance of the recognition algorithms in the presence of a moderate amount of noise (20 units). In the presence of noise, the behavior of the recognition algorithms differs significantly. Linear combination of views techniques that select random views obviously do not benefit from a larger number of training. Bounded error and 2D alignment techniques, on the other hand, improve steadily as more training examples become available.

Hybrid algorithms (`lcom-nearest-`n) that use 2D bounded error matching to select 2D views to be matched by linear combination of views perform best (in fact, the probability of error at 300 training examples was so low that it could not be measured–all objects were classified correctly in the Monte Carlo trials–and in Figure 5-10 the performance is extrapolated).

In a purely linear framework, even in the presence of noise, there would be no obvious reason why choosing nearby 2D views should improve the performance of the method of linear combination of views. However, as we stated above, the 2D bounded error matching algorithm that was used to preselect views also was used to establish correspondences between features. In particular, features that could not be aligned to within given 2D error bounds were removed from consideration by the linear combination of views scheme. This has the effect of outlier removal (Huber, 1981) and is probably responsible for making linear combination of views with 2D bounded error matching more robust than simple linear combination of views.

5.5.4 Robustness to Noise

Figure 5-11 shows how the different recognition algorithms perform when different amounts of noise are added to the locations of features in the 2D image. The number of training examples (100) was chosen such that the 2D methods had a non-negligible error in the case that no noise was added. This graph shows very clearly that using 2D recognition algorithms for 3D recognition is more robust to noise than using linear combination of views.

An interesting observation is that the linear combination of views method without view preselection performs significantly worse than the 3D alignment method in the presence of noise (cf. the graphs in Figure 5-7 and `lcom-random-3` in Figure 5-11). Because the training examples were noise-free, the linear combination of views method had available a perfect linear model of the object, and the loss in classification accuracy must be attributed to the overgeneralization of the linear model.

5.5.5 Noise during Training

The results in the previous section were derived for the case where there was no error on the location of features during training. Whether this is realistic depends on the exact setting in which a recognition system is to be used.

The same experiment was therefore repeated by having noise present both during the acquisition phase and the recognition phase. The results of these experiments are shown in Figure 5-12.

Not surprisingly, the method of linear combination of views performs more poorly in this case than in the case of error-free training examples.

Interestingly, in contrast, the performance of view-based methods improves when training examples have the same kind of noise added that the later test cases have. This is encouraging, because it suggests that the view-based methods are flexible enough that they implicitly model the noise.

5.5.6 Nonattached Features

Figure 5-13 shows the performance of the different recognition algorithms on the "jumble of lines" objects. As we saw above, features for these objects are non-attached: the relationship between feature locations in the image (namely, intersections of line segments) and the transformation applied to the 3D model is very different from that for a 3D rigid body object whose features consist of surface markings. Furthermore, because intersections between the projections of two nonintersecting 3D line segments are visible only from some viewing directions, these objects also have a non-trivial aspect graph structure.

We see that 2D bounded error matching and 2D alignment techniques outperform the linear combination of views method greatly.

There are two reasons for this. The first is that the interpolation model is inapplicable to nonattached features (in general). The other reason is that the nonattached features used in these experiments are visible only from a much smaller portion of the viewing sphere. Recognition algorithms that require correspondences among features in multiple views therefore will usually only have a much smaller number of features available for recognition–the subset of features that is present in the image and in each of the two or three views used for interpolation.

5.5.7 Correspondence Information

The bounded error and alignment techniques described above do not require correspondence information. We have therefore tested what the effect of ambiguous feature labels (multiple possible correspondences) between image features and model features is. The results (shown in Figure 5-14) indicate that these 2D recognition algorithms are still applicable even if each image feature could potentially match many model features. Furthermore, even if we want to use 3D recognition methods, the correspondences established by 2D matching are still very useful as a starting point for 3D verification and matching, since establishing correspondences directly from 3D models is computationally expensive.

5.5.8 Partial Matches

As we can see in Figure 5-15, recognition methods that are able to match objects partially perform significantly better, even if the 3D recognition problem does not involve occlusions. These results were obtained by using the `minerr-`n matching algorithm for different choices of n.

Even though the images used in these experiments had no occlusions, allowing the recognition algorithm to exclude about half of the object before assessing geometric similarity significantly improved recognition rate.

The interpretation of this finding is that many geometric relations among different features in projection remain relatively stable (though not exactly constant) for large parts of the viewing sphere. In different words, in each view, many features will have approximately the same characteristic geometric relations among themselves that they have over relatively large regions of the viewing sphere (this is particularly true for relative angles). Intuitively, allowing for partial matches means that a recognition method can exclude features that are in uncharacteristic relative positions. This was observed in Breuel, 1990, and also derived and studied in more detail by Burns *et al.*, 1990.

As can be seen in Figure 5-15, for the particular parameters used in these experiments, the effect of taking advantage of such approximate invariants on the recognition error was an improvement by factor of 3 in the best case. However, by using different formulations of bounded error recognition (e.g., the log-polar measure used in active vision, Swain and Stricker, 1991, and considered in Breuel, 1990), it may be possible to increase this effect.

Another important effect of allowing partial matches that we did not examine here is that partial matches also allow generalization across similar aspects of an object (aspects in which only a few features have become occluded or visible).

5.5.9 Interpolation and Extrapolation

Another useful way of studying view-based recognition is to look at how the quality of match value changes for a match between two views of a 3D object under differences in relative slant and tilt. Such an analysis is important because it is easy to visualize and can be related to the results of psychophysical experiments on view-based recognition (Edelman and Buelthoff, 1990a, Edelman and Buelthoff, 1990b).

We can visualize how the quality of match changes for different viewing directions by showing the quality of match for each point on the viewing sphere. Such an image is called the *error sphere.* Error spheres (flattened out) are shown in Figures 5-16 and 5-17.

The images of the error sphere are parameterized as follows. The amount of the relative rotation of the two views is given by the distance from the center of the plot, and its angle by the angle that a point in the plot makes with the x-axis. In each case, a horizontal cross-section is shown on the right.

Error spheres let us visualize to which set of rotations a given model consisting of a single view will generalize: low error (high quality of match, dark regions on the error sphere) indicates that the particular view of the object is likely to match better than any view of a non-matching object, while high error indicates the opposite.

We notice immediately that the error sphere for view-based recognition with a bounded error measure shows an anisotropic pattern, whereas the error sphere for bounded error recognition does not. This is a simple consequence of the fact that the bounded error recognition algorithm used in the experiments was based on the ∞-norm rather than the 2-norm.

However, more interesting is the fact that the cross sections for the two methods are very similar. This suggests that discriminating between different methods used by a view-based recognition algorithm for 2D matching may be difficult. In particular, while the psychophysical experiments provide strong evidence that the human visual system is view-based, they do not necessarily allow us to distinguish between many kinds of 2D matching methods that the human visual system might use.

However, there is an important observation about the view-based methods described here and the psychophysical findings. This difference relates to how two nearby (on the viewing sphere) views of a view-based object model interact.

Figure 5-18 shows a plot of the quality of match for two views (i.e., a view-based model consisting of two views or two training examples). On the right hand side, two cross sections (horizontal and vertical) through the center of this plot are shown. If we examine these plots carefully, we find that the quality of match measure returned by the view-based recognition method for a novel view is the same for a given distance from one of the known views, independent of the way in which the novel view is related to the two known views. The psychophysical findings, on the other hand, show that generalization to novel views is significantly dependent on the relationship of the novel view to the two training views.

The reason why the experiments presented here do not reproduce the psychophysical findings exactly is that views are treated as completely independent: the overall quality of match of an image to an object model is simply the best match to one of its views.

The interaction between views that is observed psychophysically has been interpreted in Edelman and Buelthoff, 1990a (see also Poggio and Edelman, 1990), as the result of interpolation between different views. In an interpolation model, the purpose of such interactions would be to allow the visual system to allow more reliable generalization to novel views. However, while an interpolation theory (in particular, the RBF method) is consistent with the psychophysically observed results, it is not the only possible explanation.

One alternative explanation for the psychophysically observed interaction among object views is that the fact that two views match the given view could be considered as "partially independent evidence" (in a Bayesian framework) for the presence of the object in the image (see also Breuel, 1990). Such a Bayesian interaction would fit well with the non-geometric, statistical aspects of recognition that a visual system has to address (see Chapter 7).

Which of the two hypotheses about the interaction of different views of the same object, interpolation or Bayesian interaction, applies to the human visual system is an open question.

The question could be resolved experimentally as follows. Subjects are trained on two different, separated views of an object. Assume that the object consists of two parts. Some test cases would consist of intermediate views like those tested in Edelman and Buelthoff, 1990a. Other test cases would consist of intermediate views in which the two parts of the object have been moved relative to one another such that each part is closer to one of the two training views.

A view interpolation theory would predict that the deformed test object constitutes a poor match, since its geometry has been altered relative to the training cases. For certain choices of parameters and models, a Bayesian theory, on the other hand, would predict that certain deformed objects are recognized better than intermediate views, because each partial match provides strong, independent evidence for the presence of the object. Therefore, a finding that certain kinds deformed objects are recognized better than some intermediate views would provide evidence against an interpolation theory of visual object recognition and could be explained in terms of certain theories of Bayesian combination of evidence; a negative result would leave the question undecided.

The RBF theory of visual object recognition has been formulated and described as an interpolation theory (Poggio and Edelman, 1990). However, above, we saw that there is an important distinction between a Bayesian and an interpolation theory of object recognition from training views. It is therefore interesting to note that the RBF method could also be interpreted as a commonly used Bayesian heuristic: linear combination of evidence (Berger, 1985). In such a view, each of the "centers" or "units" of an RBF network provides evidence for the presence of an object in the image. The evidence from each unit is weighted and combined linearly.

While this interpretation is intuitively appealing, it is important to note that the training methods for linear combination of evidence and for an RBF network are different (giving rise to different weights). Furthermore, no sound theoretical foundation for the use of linear combination of evidence methods in Bayesian statistics appears to be known (but the method has been found to work well on many practical problems).

5.6 Discussion

This chapter has described the implementation of a general purpose simulation system for 3D recognition. Using this system, we have conducted a number of experiments comparing the performance of 3D model based and strictly view-based recognition system on large model bases and in the presence of error.

Based on these simulations, we make the following observations:

- Strictly view-based recognition systems are easy to implement.
- Strictly view-based recognition systems are robust to deviations from the rigid body/attached feature model commonly used in computer vision. Not surprisingly, methods that rely on the 3D structure of objects (linear combination of views) fail under such conditions. 3D model-based recognition schemes (3D alignment) are simply inapplicable.
- Even "optimal" 3D model-based recognition systems (least-square error recognition) with perfect models cannot distinguish arbitrarily large sets of objects in the presence of error. This fact is not surprising in principle, but it is interesting that it can be observed in practice.
- The need to establish correspondences between multiple example views and the image is a significant problem for applying the linear combination of views method.
- Allowing for partial matches between model views and an image allows features in uncharacteristic positions to be excluded from a match and reduce the number of views needed to represent an object.
- Combining a strictly view-based approach with a 3D model-based recognition approach like linear combination of views may be advantageous for domains that require high-precision 3D matching and do not require robustness to non-attached features. In such an approach, strictly view-based recognition is used to establish feature correspondences efficiently, and the 3D model-based step verifies the precise geometry of the object.

The simulations also have significant implications for the interpretation of psychophysical experiments. Most importantly, they show that we do not need to assume any interactions between views (as postulated by view interpolation schemes) to achieve significant generalization over patches of the viewing sphere. Furthermore, even very simple bounded error matching schemes generate smooth error spheres that are qualitatively very similar to the error spheres for more "neural network-like" models of matching.

Psychophysically observed effects that we made no attempt to duplicate in these experiments are those of anisotropic generalization, as well as interactions between training views. In particular the question of the nature of the interaction between training views, whether they interact via interpolation or Bayesian combination of evidence, is an important one and should be addressed psychophysically and theoretically.

In order to carry out these simulations, specific choices of parameters needed to be made. The most important choices were the number of objects in the database and the amount of noise on the location of features in the image. These two choices dictated the complexity of the objects and the choices of parameters for individual recognition algorithms. The qualitative results of the simulations are not very dependent on these choices of parameters. It is important to realize, however, that the quantitative results (number of views needed to represent an object, degree of generalization to novel viewpoints) are dependent on such choices of parameters. The same observation applies to other simulations and even psychophysical experiments reported in the literature.

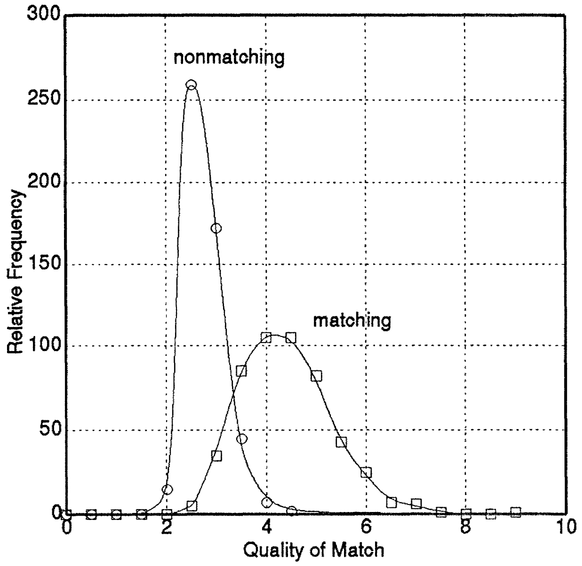

Figure 5-5: An example of the distribution functions of the quality-of-match measure.

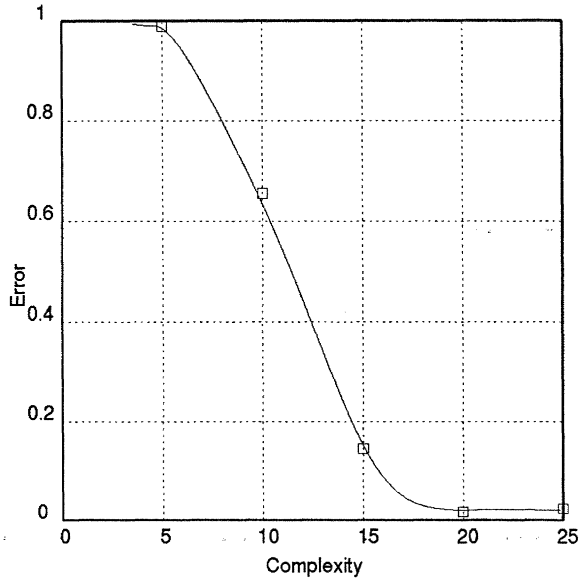

Figure 5-6: Probability of confusing objects of different complexity under 3D model based recognition (alignment; noise: 40 units).

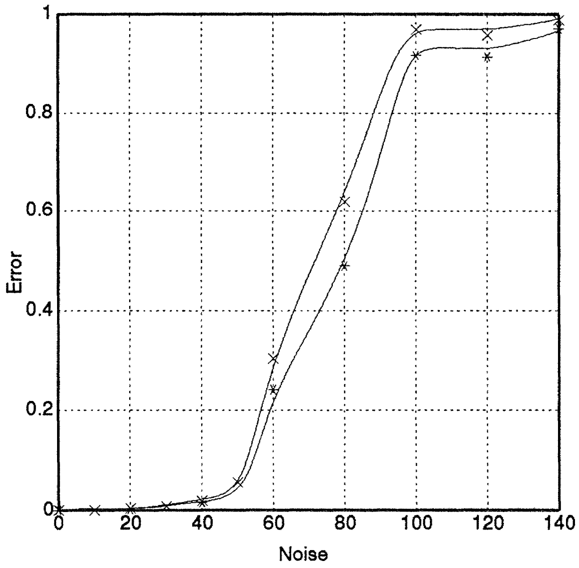

Figure 5-7: Robustness of 3D model based recognition to noise. The top graph (×) shows the error rate for 3D alignment; the bottom graph (*) shows the error rate for optimal least-square matching.

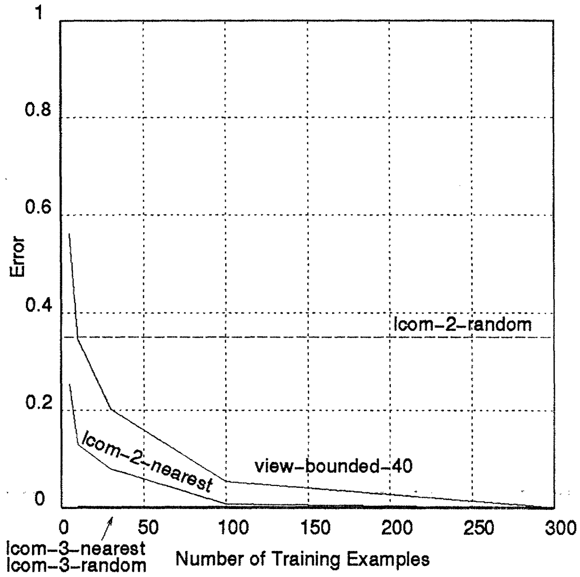

Figure 5-8: Number of training examples vs. error, in the absence of noise.

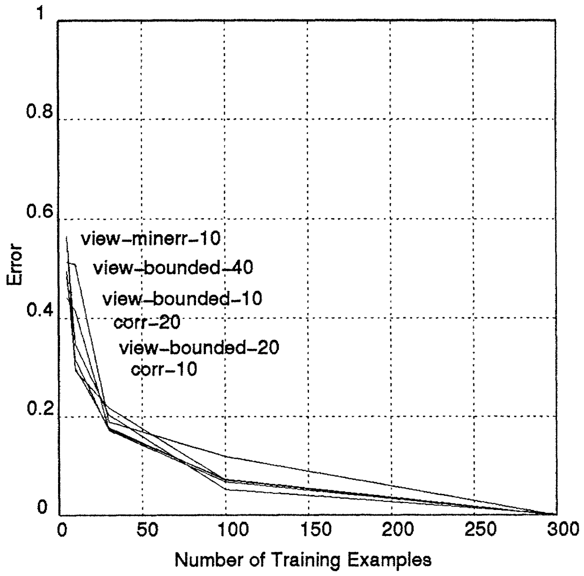

Figure 5-9: Number of training examples vs. error, in the absence of noise; a comparison of bounded error and alignment techniques.

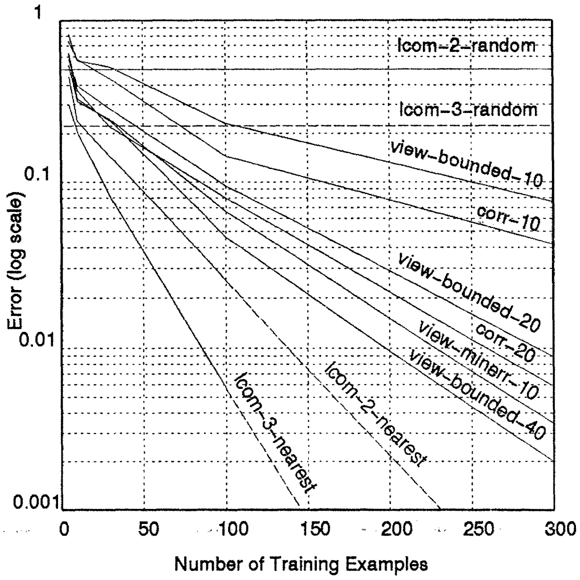

Figure 5-10: Number of training examples vs. error, in the presence of noise (noise: 20 units).

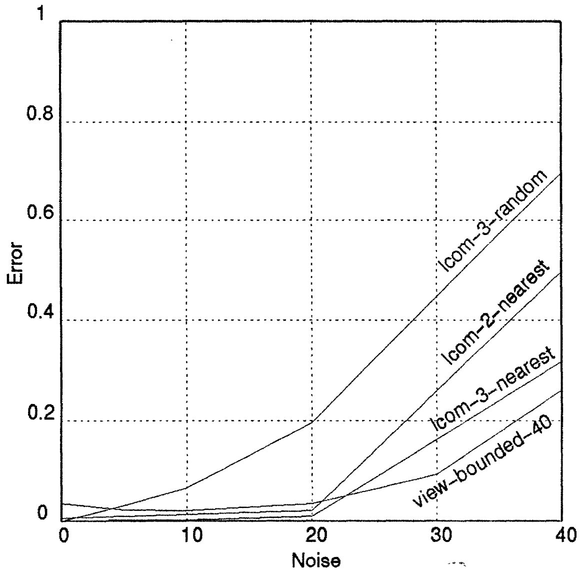

Figure 5-11: Robustness to noise (100 training examples).

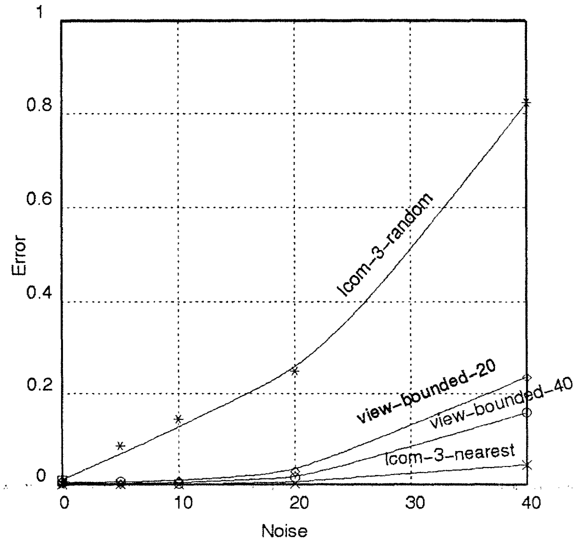

Figure 5-12: Robustness to noise, imperfect models (100 training examples).

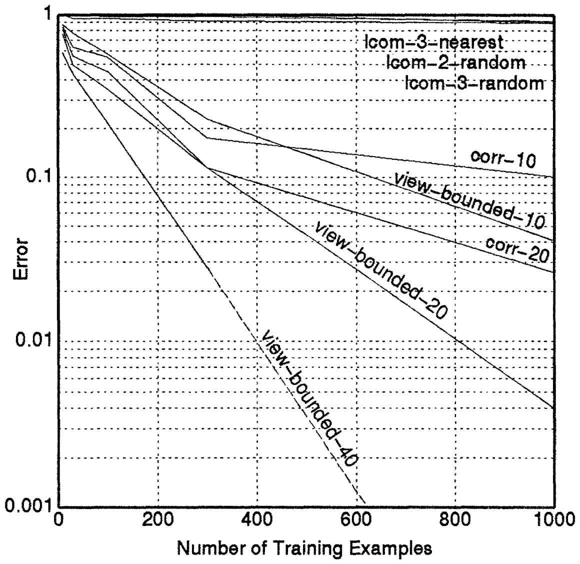

Figure 5-13: Number of training examples vs. error for a class of objects with non-attached features (no location error).

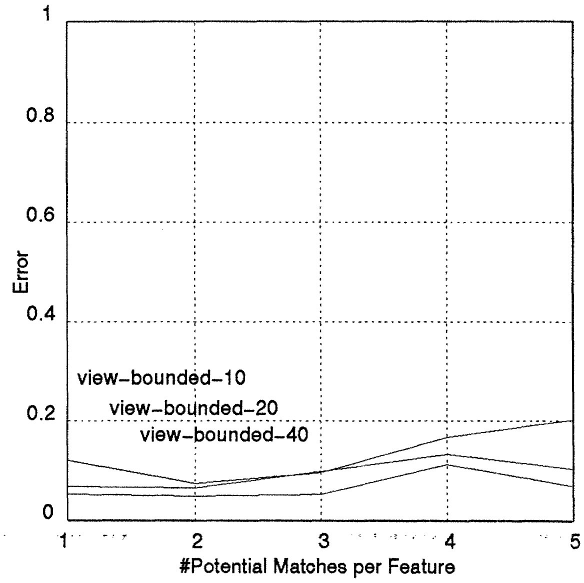

Figure 5-14: Robustness to ambiguous correspondence information. (no noise, 100 training examples)

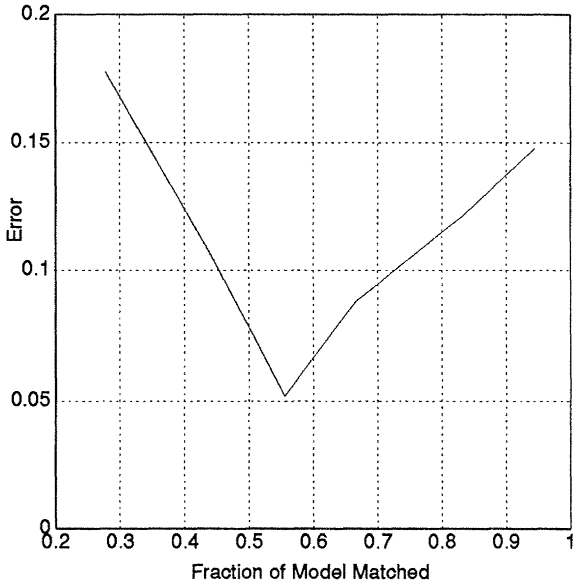

Figure 5-15: Allowing partial matches improves performance significantly. (no noise, 100 training examples)

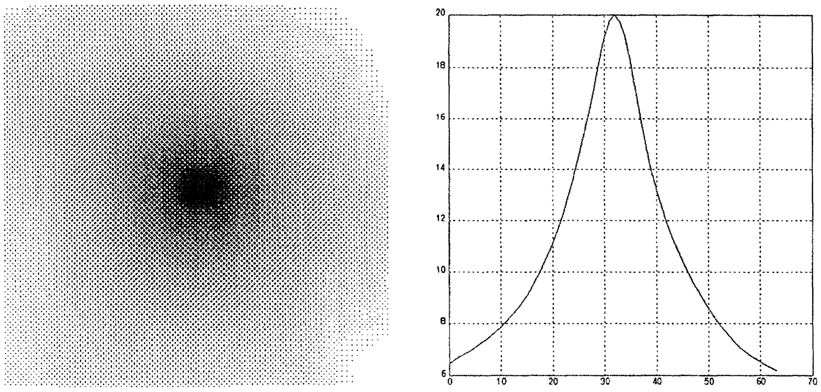

Figure 5-16: Extrapolation for a single view under 2D alignment with a Gaussian error measure ($\sigma = 20$).

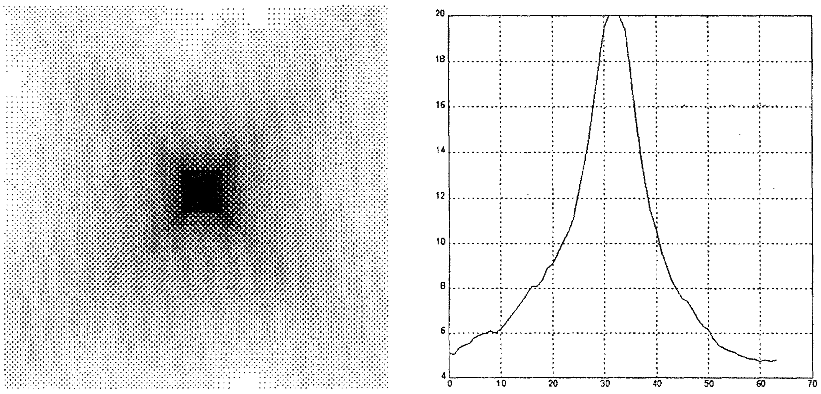

Figure 5-17: Extrapolation for a single view under bounded error recognition ($\delta = 10$).

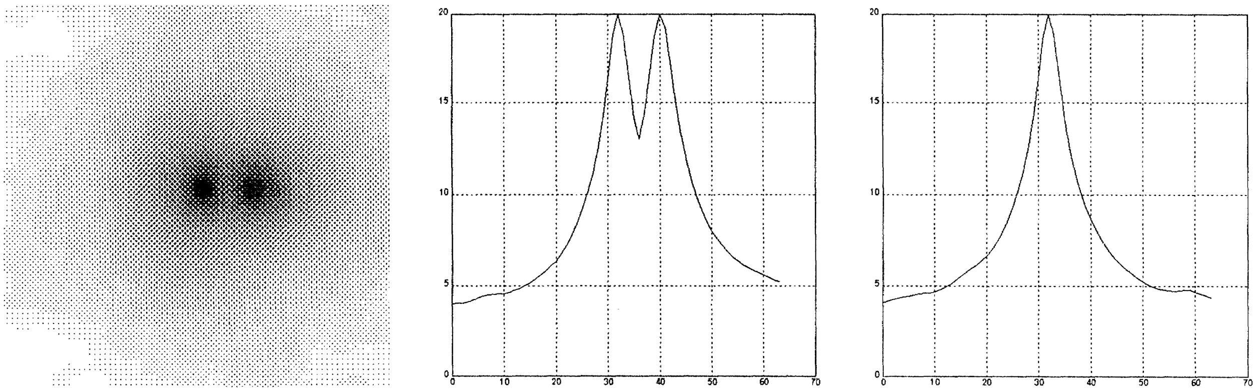

Figure 5-18: Extrapolation and interpolation for two views under 2D alignment with a Gaussian error measure ($\sigma = 10$, views separated by 23 deg).

Chapter 6

Examples of View-Based Recognition

6.1 Introduction

In previous chapters, we have studied view-based recognition from a theoretical perspective and described the results of simulations.

This chapter gives some examples of recognition by a view-based recognition system on scenes of actual 3D objects. The domain is recognition of toy airplanes (a model Korean airliner and a model Mig-21) in scenes consisting of toy airplanes, cars, wooden blocks, and plastic animals from grey-level images.

6.2 Methods

Preprocessing consisted of edge extraction with a Canny-Deriche edge detector. A single set of parameters for the edge detector was used for all models and images. Edges from the edge detector were approximated by polygonal chains with segments of fixed length (10 pixels). The location and orientation of each segment was used as features for the recognition algorithm (for further details on the feature extraction, see Appendix A).

View-based recognition, as described in Chapter 5, uses a 2D matching algorithm that is invariant under translation, rotation, and scale. However, for these experiments, we took an even simpler approach. Because the edge detector and the features used are not invariant under scale, for each model view, features were generated at three scales of the objects (0.7, 1.0, 1.4) by subsampling the original grey-level image and carrying out the feature extraction process. Because we have a very efficient (fixed point) implementation of RAST for the case of 2D translation, invariant matching under rotation was achieved by representing each model at 41 different orientations (i.e., differing by 9° each).

The quality of match between a 2D image and a 2D model was taken to be the fraction of the model features accounted for in the image, maximized over all translations, the 3 scales and 41 rotations, and the 32 views. An error of 10 pixels was allowed in the alignment between model and image features.

All the parameters for feature extraction and recognition were "first guesses": there was no attempt to vary parameters to optimize recognition performance.

6.3 Images and Models

Altogether, the view-based model of the Korean airliner consisted of 32 different views (a similar experiment with similar results used 21 different views of the Mig-21). These views are shown in Figures 6-24 and 6-25. The model views were not processed or "cleaned up" by hand in any way.

The view-based models were matched against 22 scenes consisting of several objects. Some scenes contained occlusions. Other scenes did not contain any matching object.

6.4 Results

In 21 out of 22 images, the Korean airliner was identified correctly as the first choice by the recognition algorithm, even in the presence of occlusions. For the image in Figure 6-22, which is the most heavily occluded case, the second highest match corresponds to the correct answer.

Because of the coarse quantization of scale (only three scales were used for the matching), the match is less than perfect in some images even though a better match should be possible (e.g., Figures 6-7 and 6-13).

In some cases, correspondences between some model parts and some image parts are incorrect, but the match is still in approximately the correct position (e.g., Figure 6-13). This may be due again to the coarse quantization of scale and rotation, or it may point out a shortcoming of the simplistic 2D quality-of-match measure used; we will investigate alternative quality-of-match measures in Chapter 7.

Except for those cases in which the Korean airliner was heavily occluded, matches against images containing the Korean airliner scored higher than images not containing the Korean airliner. Furthermore, in all cases, when the best match of the model of the Korean airliner was to a Mig-21 (Figure 6-18 and 6-20), a match of the Mig-21 model to the same image scored significantly higher than the Korean airliner model (data not shown). This suggests a rudimentary ability to generalize to other shapes: a Mig-21 is a kind of airplane, but it matches a model of a Mig-21 better than a model of a Korean airliner.

In fact, in a separate set of experiments, the view-based recognition system was tested on the same set of images using view-based models of both the Mig-21 and the Korean airliner, and we found that it always correctly distinguished between the Korean airliner and the Mig-21.

By examining the images closely, we find that the view-based recognition system also matched the shadow of the airplane (e.g., Figure 6-3). For an object like the Korean airliner, the shadow is actually a useful feature for aiding in recognition from many viewpoints. Likewise, in some images, highlights along the fuselage contributed to the correct match. It is interesting to see that the view-based recognition algorithm can easily and automatically take advantage of such regularities that would otherwise be very difficult to discover and model analytically.

6.5 Discussion

These results demonstrate that even very simple and coarse view-based matching strategies can give good results for 3D object recognition from 2D images.

What appears to make the system robust is the fact that it does not rely on the extraction of a small number accurate point or line features. Instead, it is based on a large number of very simple features: approximations to edges by polygonal chains with fixed segment length. These features can be extracted reliably and stably (within the uncertainty of the segment length). Using such features with previous matching algorithms would have been prohibitively expensive. But the RAST algorithm has no difficulty with matching images consisting of thousands of unlabeled features against models consisting of hundreds of unlabeled features.

However, the evaluation of the quality-of-match between models and views clearly leaves many things to be desired. Some of the important open questions that these experiments raise are:

- We need to find methods for comparing the quality of match of models at different scales: the fraction of the matching boundary of a small model is more likely to be large by chance than that of a large model (this is the cause of the incorrect match in Figure 6-22).
- We need to find methods for comparing the quality of match of different models: some models may be much more likely to match well by chance than others.
- Different parts of a model may have different importance for evaluating the quality of match. We should be able to acquire these weights automatically (that is, we should estimate the α's and β's from Section 7.4 automatically; some work on this problem has been done in the context of industrial parts recognition, Wilson, 1990).
- Some objects need to be decomposed into parts in order to evaluate the quality of match measure meaningfully. For example, an airplane without wings is not a good airplane, no matter how well the fuselage matches. On the other hand, wings without a fuselage do not make a good airplane either. The interaction is non-linear.

In Chapter 7, we examine some of these issues in more detail.

It is important to realize that the question of what we mean by a "good 2D match" is fundamental both to view-based recognition methods and to 3D model based recognition methods.

For example, recognition by parts based on 3D models has been studied extensively in the literature (see Brooks, 1983, Biederman, 1985, Connell, 1985, Ettinger, 1987, Grimson, 1989c). However, acquiring a library of such 3D parts automatically is apparently a difficult task, and most researchers have been satisfied with picking a small set of 3D primitives (generalized cylinders, geons, etc.) on which to base parts-based recognition schemes. In a view-based approach, however, parts can be 2-dimensional; a learning algorithm that attempts to identify and use object parts is not constrained to represent parts as consistent 3D objects. This simplifies the learning problem significantly.

Altogether, the results from these experiments support the idea that view-based recognition is a practical approach to 3D recognition. But they also point out that further research is needed to improve the quality of match measure between 2D views of objects. Such improvements are needed regardless of whether we take a view-based approach to 3D object recognition or an approach based on 3D object shape; however, they are particularly easy to implement and test in the context of view-based recognition.

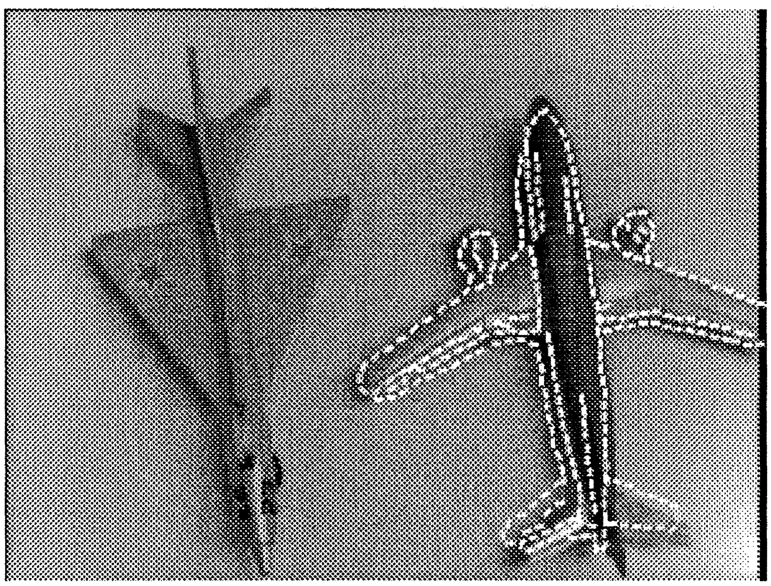

Figure 6-1: model: 15, image: 20, scale: 1.0, N: 231, fraction: 0.825

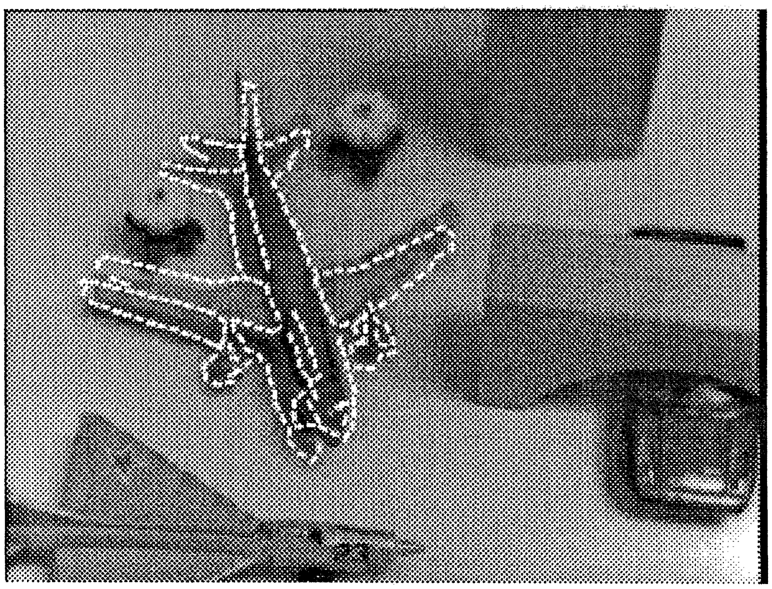

Figure 6-2: model: 01, image: 17, scale: 1.0, N: 177, fraction: 0.815668

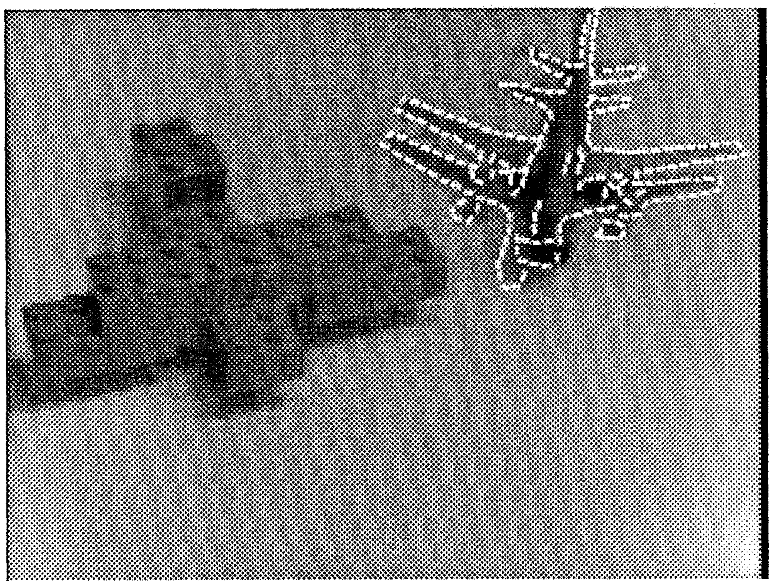

Figure 6-3: model: 32, image: 13, scale: 0.7, N: 160, fraction: 0.80402

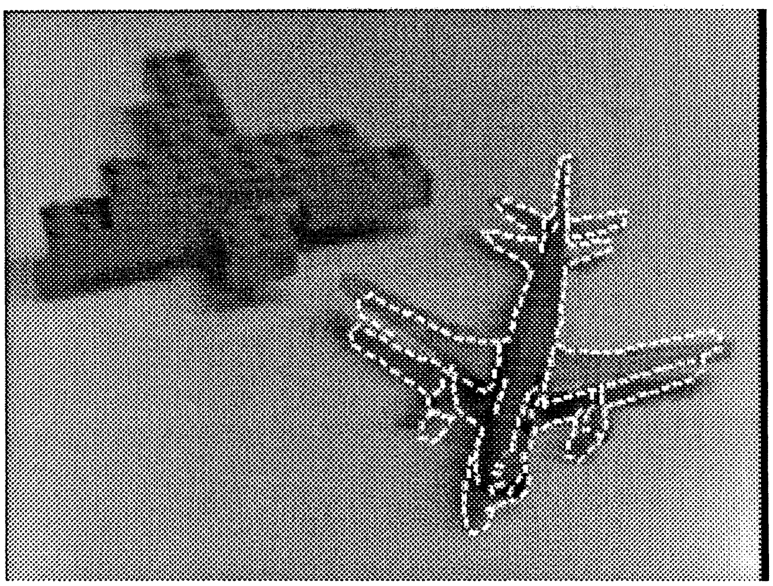

Figure 6-4: model: 02, image: 14, scale: 1.0, N: 180, fraction: 0.803571

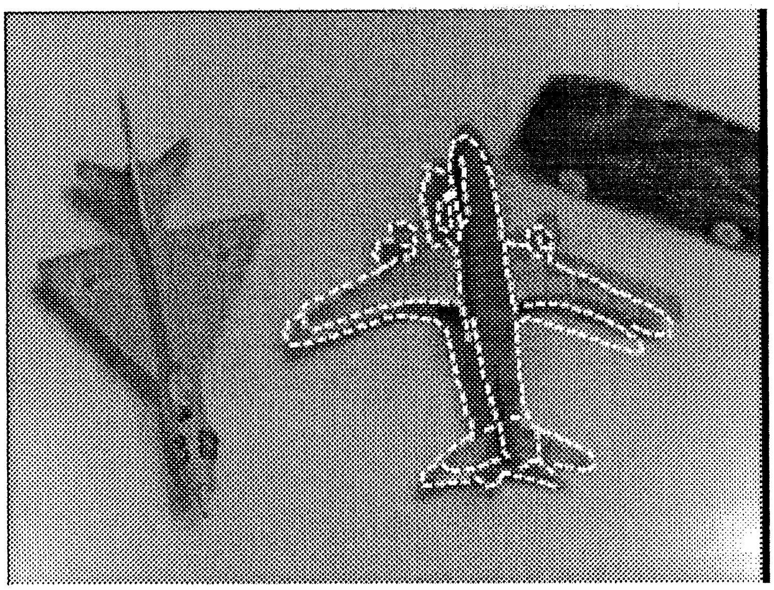

Figure 6-5: model: 06, image: 19, scale: 1.0, N: 173, fraction: 0.797235

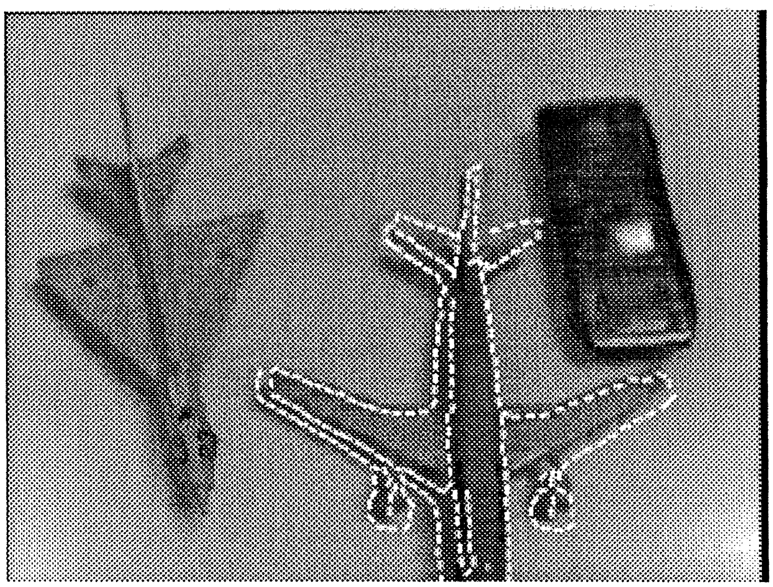

Figure 6-6: model: 19, image: 18, scale: 1.0, N: 186, fraction: 0.788136

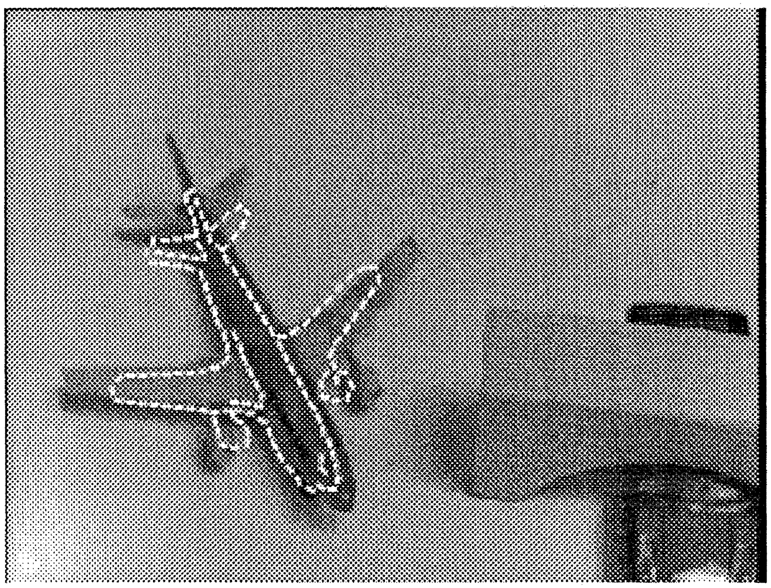

Figure 6-7: model: 19, image: 16, scale: 0.7, N: 105, fraction: 0.783582

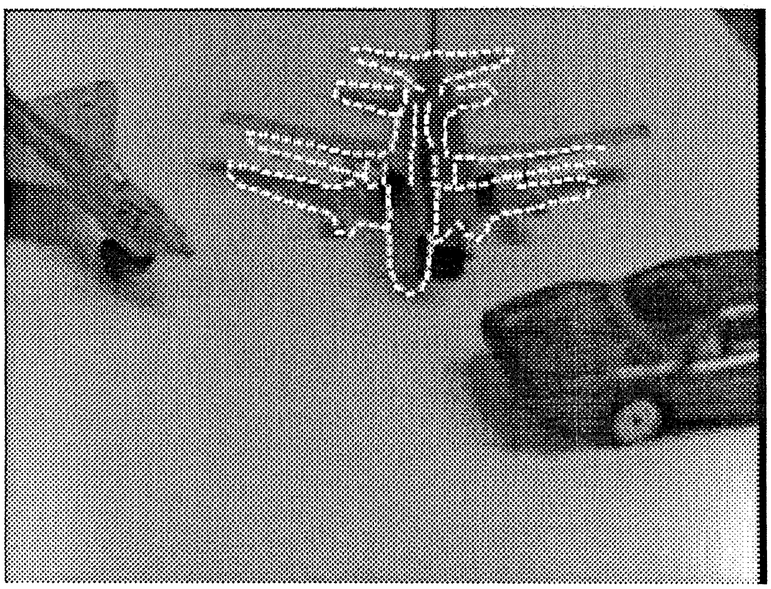

Figure 6-8: model: 26, image: 01, scale: 0.7, N: 150, fraction: 0.777202

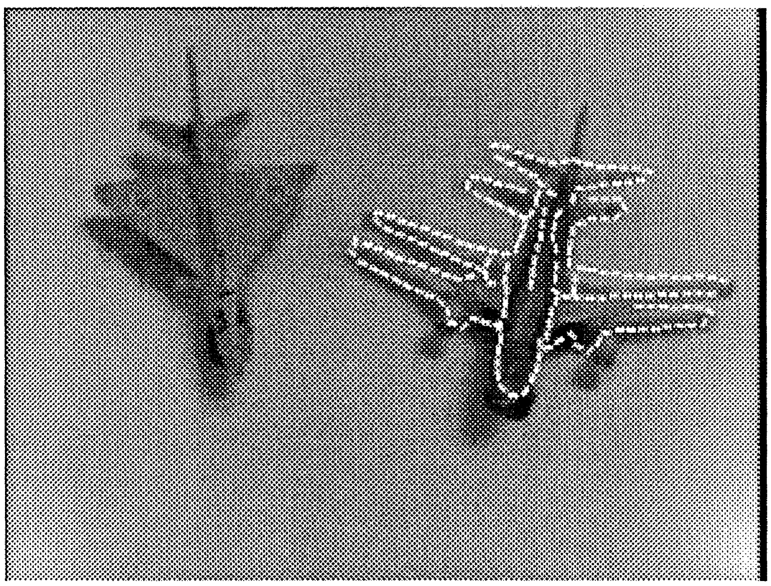

Figure 6-9: model: 26, image: 12, scale: 0.7, N: 150, fraction: 0.777202

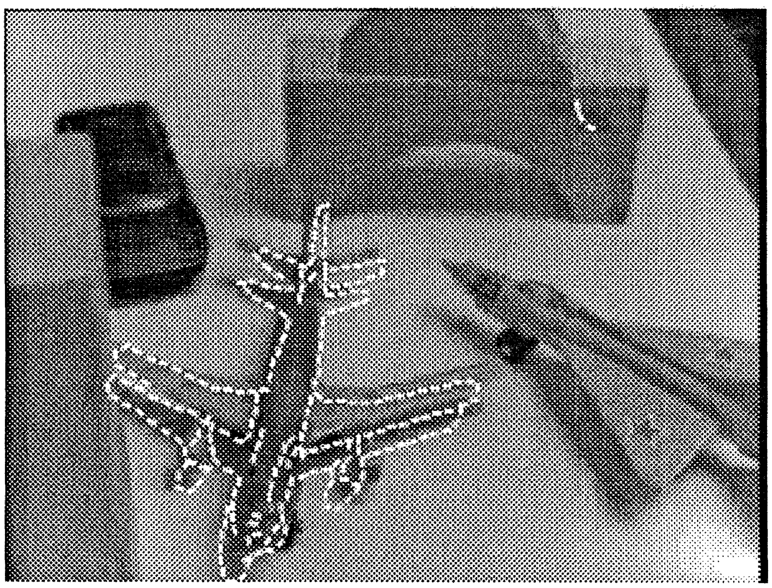

Figure 6-10: model: 02, image: 05, scale: 1.0, N: 173, fraction: 0.772321

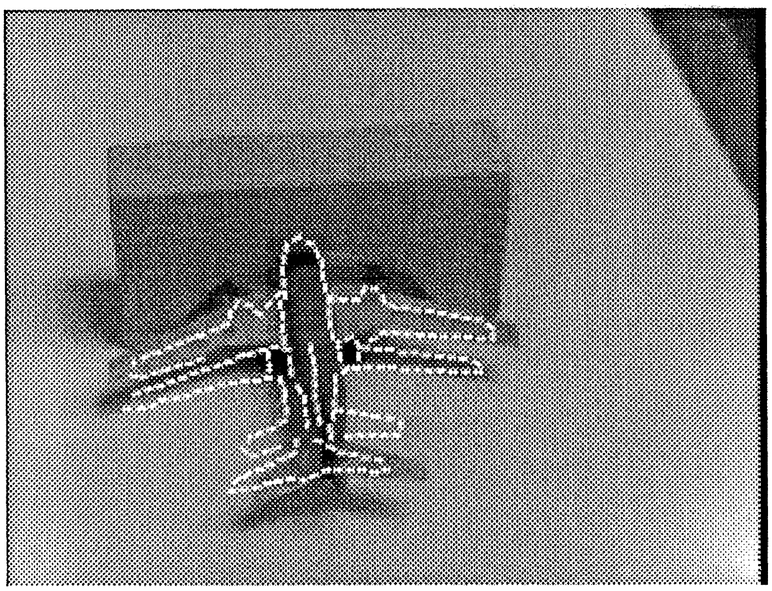

Figure 6-11: model: 26, image: 08, scale: 0.7, N: 146, fraction: 0.756477

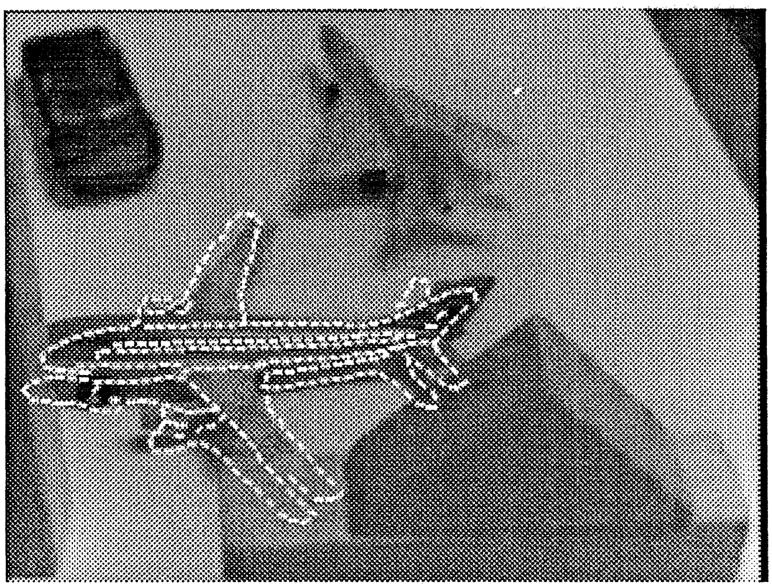

Figure 6-12: model: 04, image: 06, scale: 1.0, N: 215, fraction: 0.751748

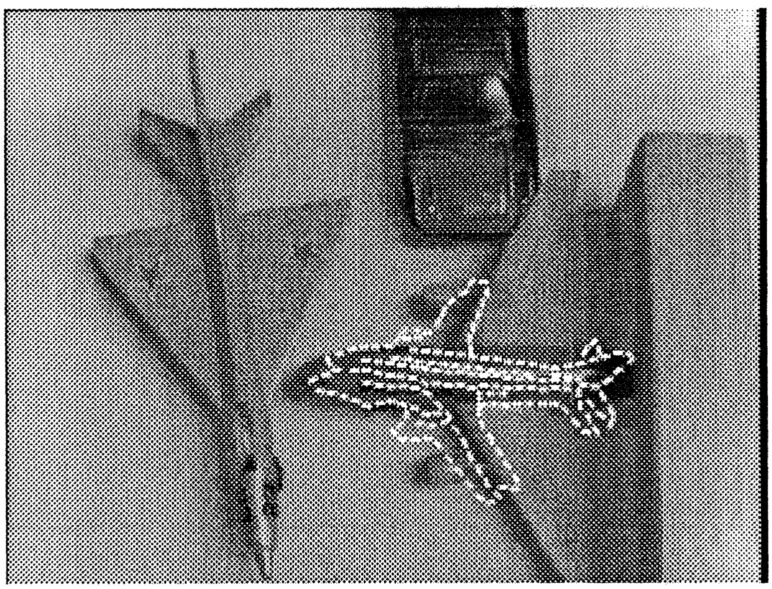

Figure 6-13: model: 04, image: 21, scale: 0.7, N: 131, fraction: 0.744318

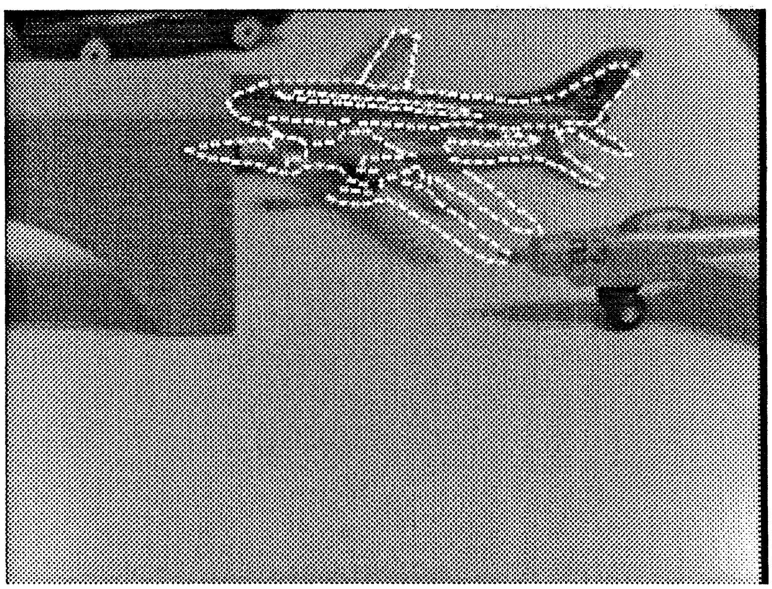

Figure 6-14: model: 23, image: 03, scale: 0.7, N: 182, fraction: 0.7251

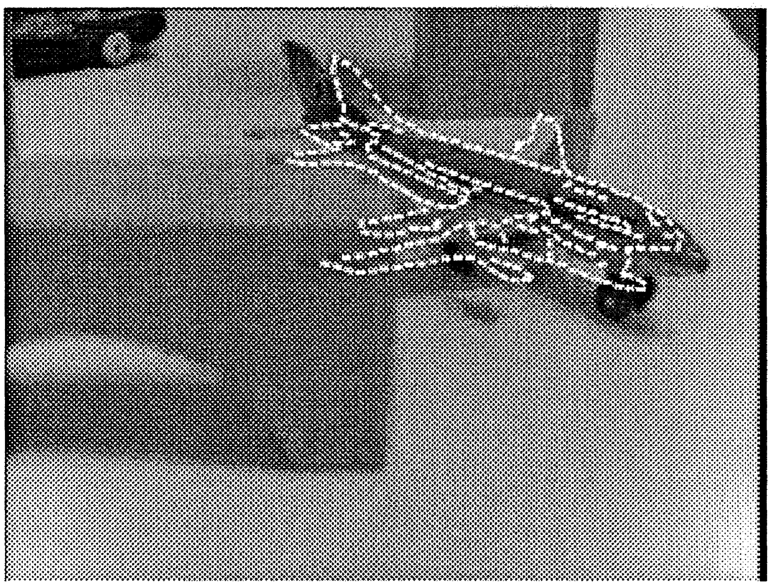

Figure 6-15: model: 30, image: 04, scale: 0.7, N: 165, fraction: 0.708155

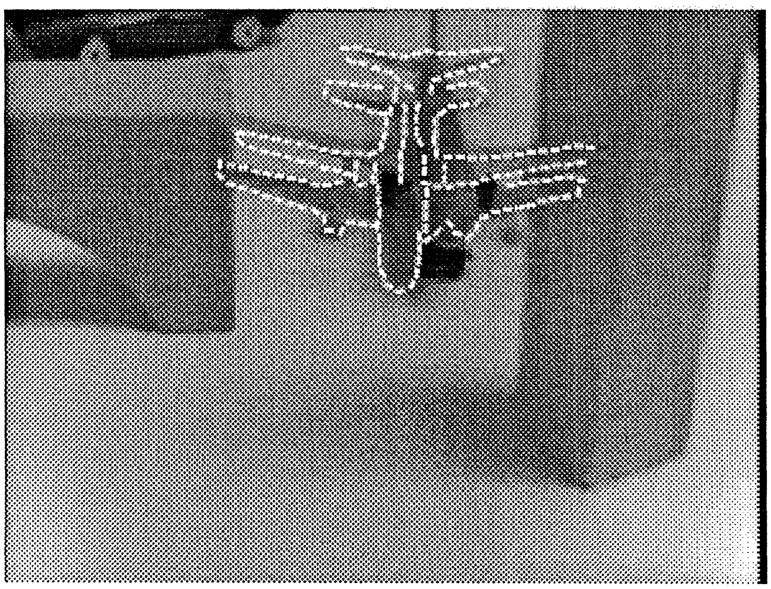

Figure 6-16: model: 26, image: 02, scale: 0.7, N: 132, fraction: 0.683938

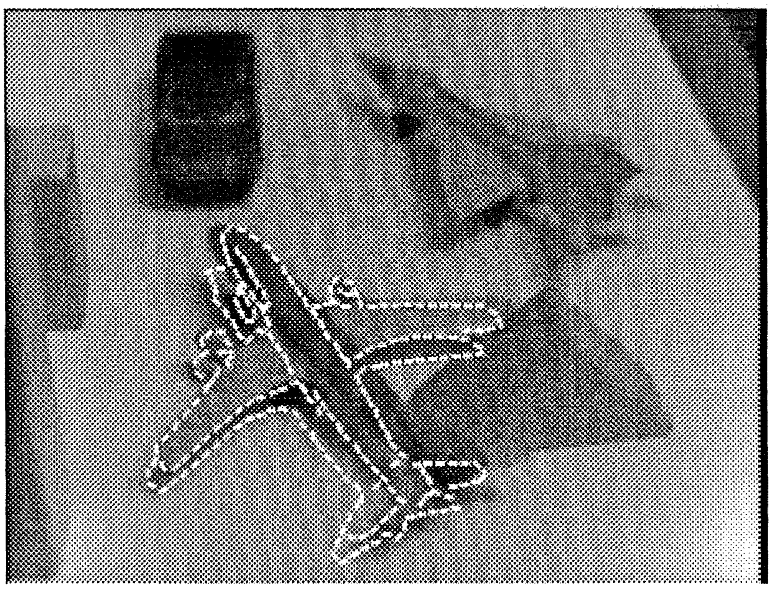

Figure 6-17: model: 06, image: 07, scale: 1.0, N: 148, fraction: 0.682028

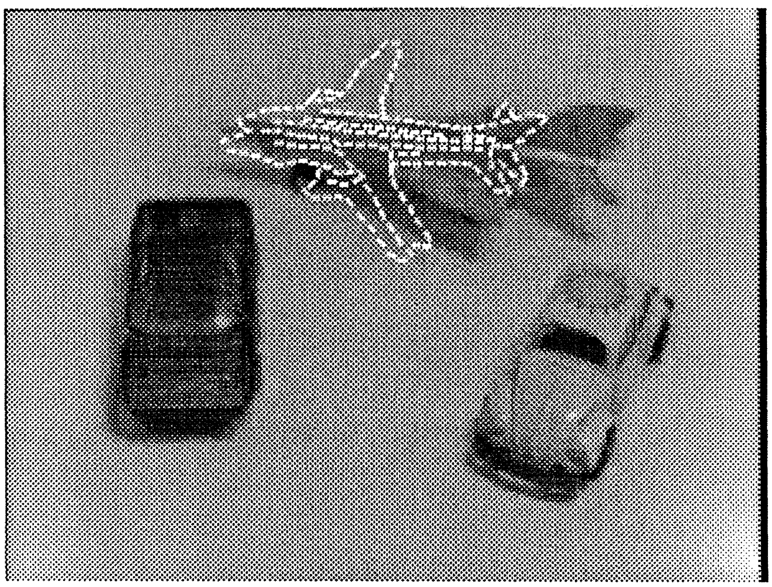

Figure 6-18: model: 04, image: 11, scale: 0.7, N: 115, fraction: 0.653409

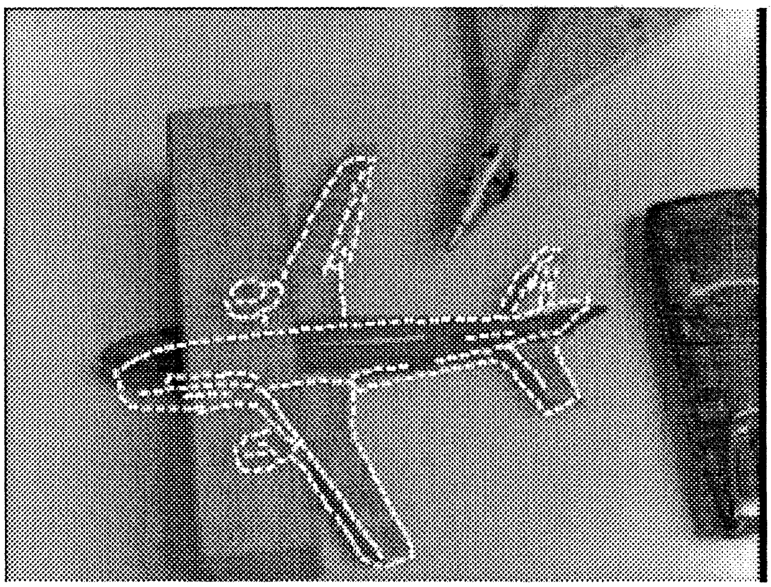

Figure 6-19: model: 14, image: 22, scale: 1.0, N: 170, fraction: 0.62963

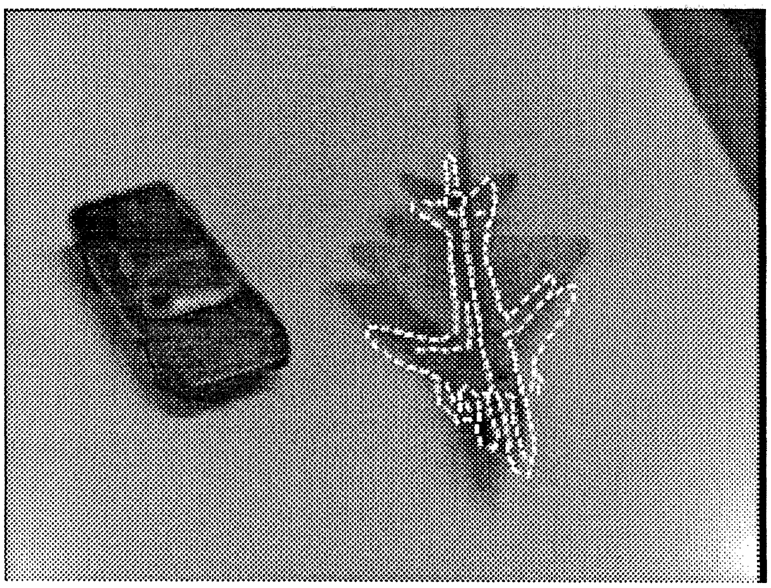

Figure 6-20: model: 08, image: 10, scale: 0.7, N: 92, fraction: 0.61745

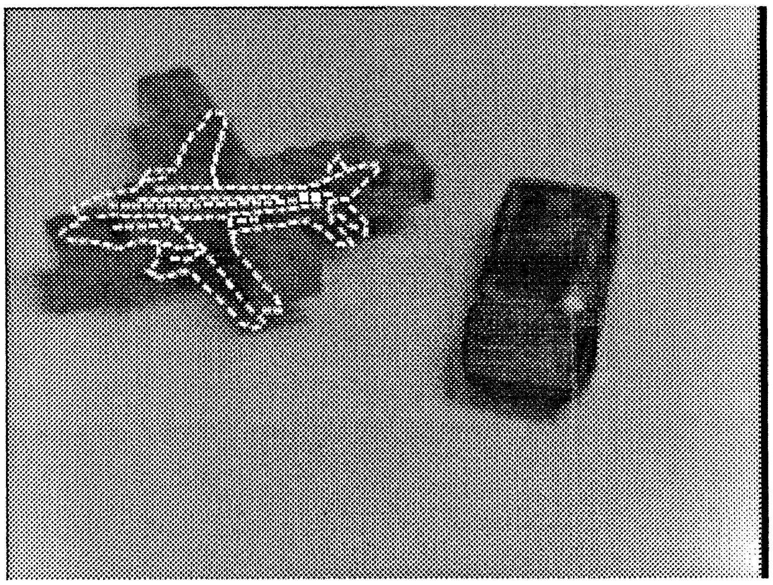

Figure 6-21: model: 04, image: 15, scale: 0.7, N: 88, fraction: 0.5

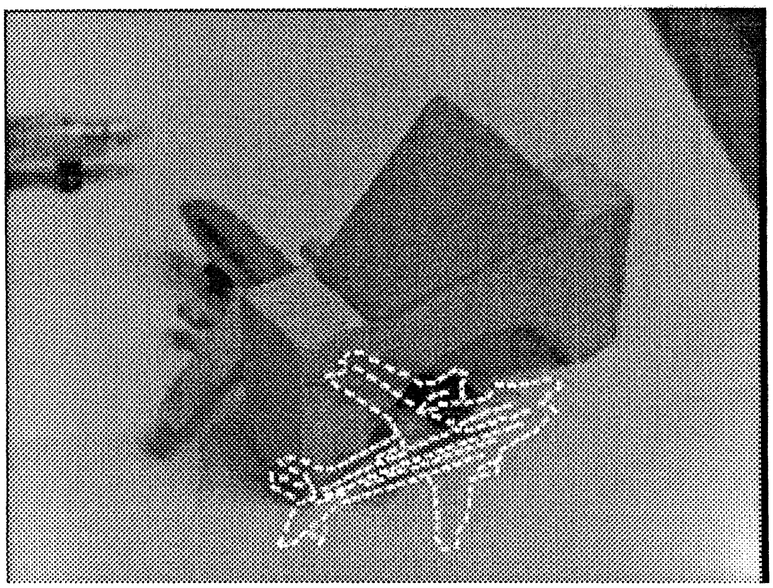

Figure 6-22: model: 04, image: 09, scale: 0.7, N: 87, fraction: 0.494318

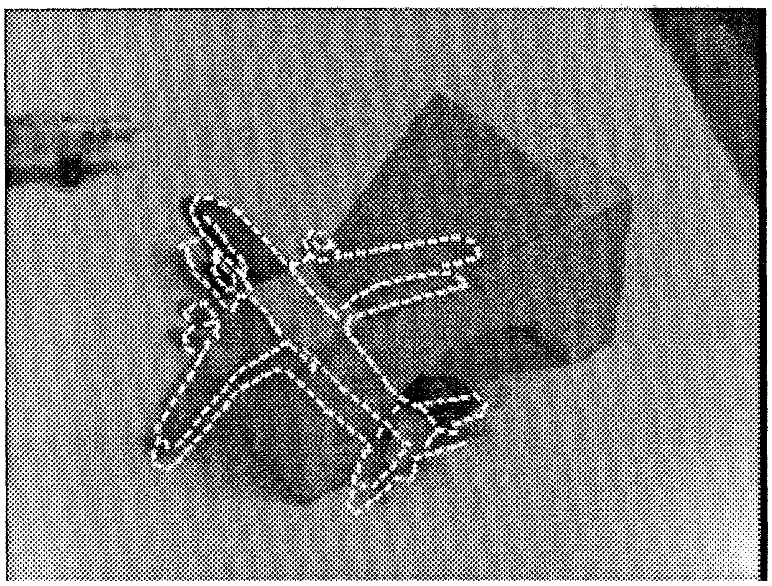

Figure 6-23: model: 06, image: 09, scale: 1.0, N: 107, fraction: 0.493088

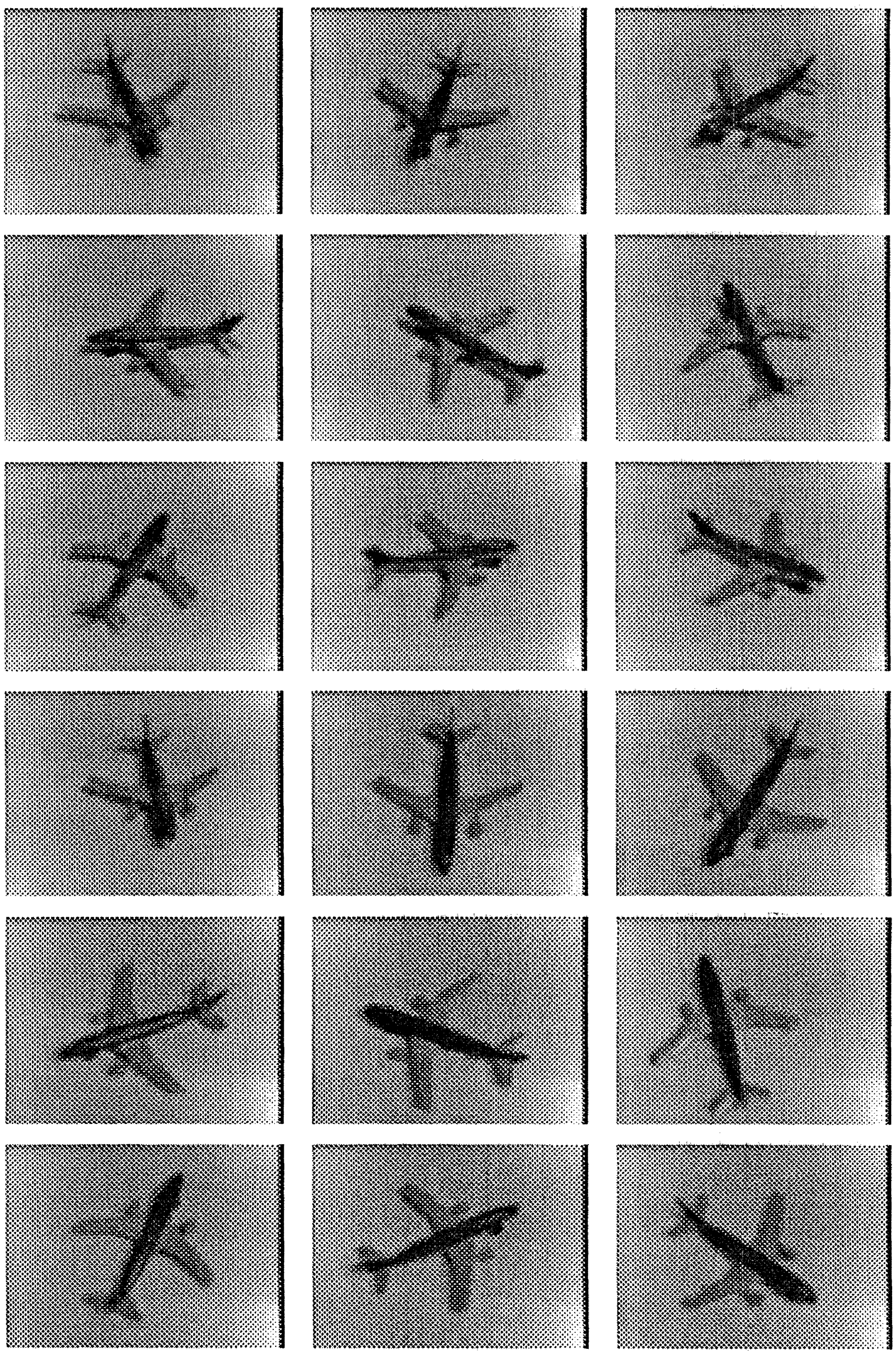

Figure 6-24: Views used for the recognition experiment.

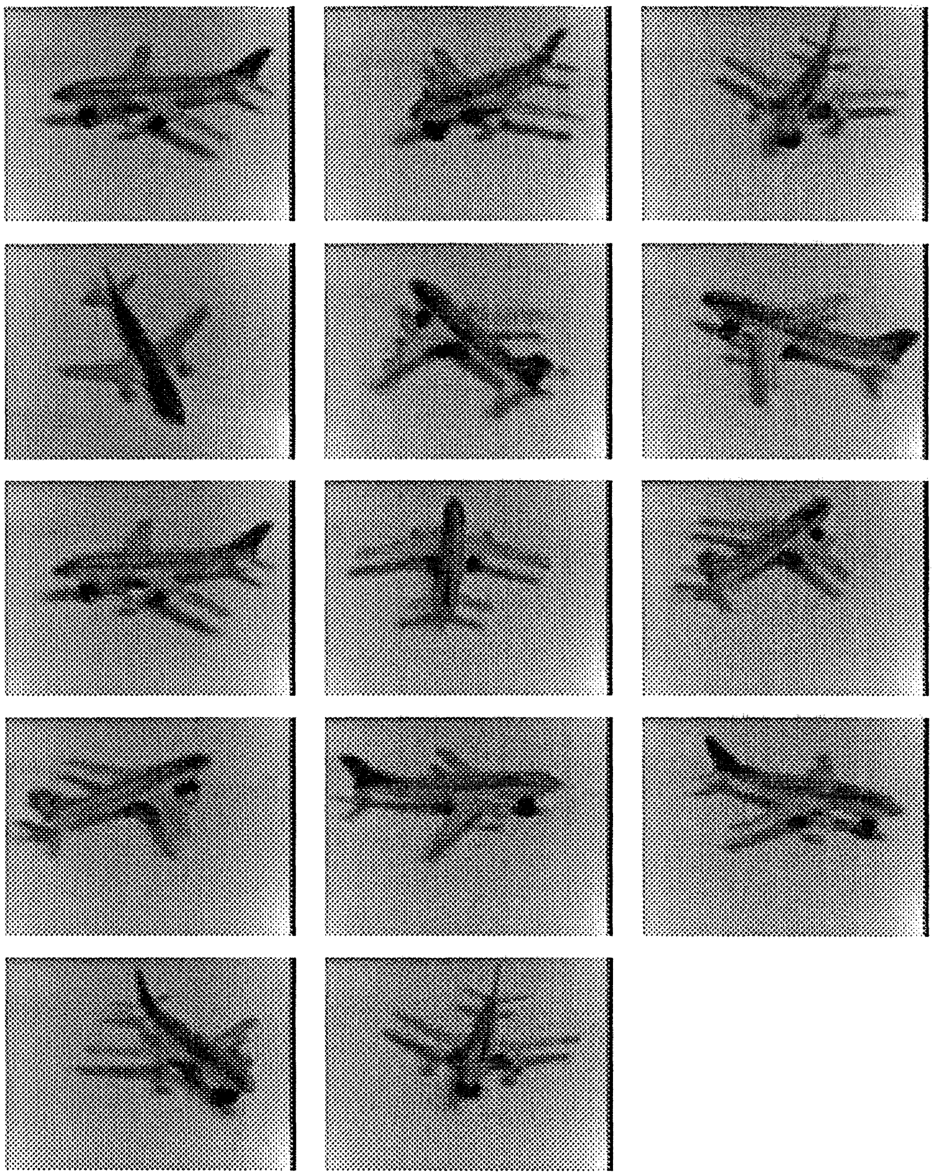

Figure 6-25: More views used for the recognition experiment.

Chapter 7

Statistical Principles

7.1 Introduction

In previous chapters we have analyzed the problem of 2D and 3D object recognition under a bounded error model. This means that the quality of match between a model and an image was given by the number of features or the fraction of the boundary of the image that could be accounted for in the image (cf. Section 2.5).

This approach works well for objects like metal widgets or scissors whose geometry can be modeled well (Figure 7-1). For the recognition of less well-defined objects in a cluttered scene, however, it fails (Figure 7-2). We have encountered this phenomenon already in Section 5.5.1: even if we model the geometry of an object perfectly, in the presence of significant error, the number of objects that can be distinguished reliably by geometry alone is limited.

Our goal in this chapter is to describe an approach towards finding quality of match measures that let us recognize objects in images more reliably.

In order to be able to distinguish objects more reliably even in the presence of significant error on the location of features, we have several means at our disposal:

- Use a *better model of the transformations* that an object can undergo. It is plausible that in many cases different instances of an object are related by a small class of deformations rather than just rigid body transformations.

- Take advantage of *topological* information; e.g., objects can be divided into broad classes based on the topology by which their parts are interconnected.

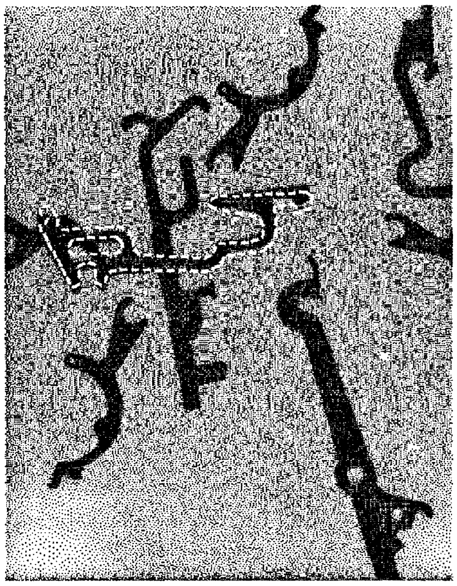

Figure 7-1: Bounded error recognition works well on rigid objects with well-defined geometrical shape, even in the presence of occlusion and clutter.

- Use more *complex features.* For example, many polyhedral objects can be distinguished simply by the types of vertices that they contain (this is similar to a hierarchical approach to matching).
- Use *non-geometric* information like color or texture.
- Use *world knowledge* about the structure of scenes.

In this chapter, we will analyze and demonstrate the use of world knowledge and its expression in the statistical properties of matches for improving the reliability with which a recognition system identifies the correct model that corresponds to an object in the image.

7.2 Statistical Principles

7.2.1 Why does bounded error recognition fail?

Recognition algorithms usually take as a measure of the quality of match between an image and a model the number of features that can be matched between the image and the model subject to given error bounds (other geometric error models are similar).

Empirically, bounded error recognition is not very discriminating for real objects: when error bounds become large enough to account for common errors on location, a model will match well in inappropriate places (see Figure 7-2 for an example). We find that such inappropriate matches occur when one object model matches small parts of a large number of other objects in the image. This is particularly common in very cluttered images or images that contain textured regions.

An example of this is shown in Figure 7-2: a bounded error recognition algorithm applied to the model of Snoopy's head located the model in the brushes rather than at its correct location. The reason is that the brushes contain a large number of features. Under large error bounds, many of these features might match features from Snoopy's head well. Making the error bounds small, on the other hand, means that only very few model features can be brought into correspondence with the instance of the object in the image, because the object "Snoopy" is geometrically not very well defined.

An incorrect match of Snoopy as shown in Figure 7-2, is characterized by a large number of small, disconnected, matching boundary segments (Figure 7-3). Intuitively, in order to justify such a match, a recognition algorithm would have to postulate a very complicated occluding object that would make a large number of small, disconnected pieces of the matching object visible. Furthermore, the different image parts that constitute such inappropriate matches do not usually have the non-accidental properties (Lowe, 1986) that we would expect between different parts of a single occluded object.

We will examine the statistical consequences of this idea in the next sections. Furthermore, we will see that such inappropriate matches can often be identified and excluded without a detailed interpretation of the complete image by considering the higher-order statistics of the matching features.

Conversely, we may be willing to accept a certain match of a model to an image if we can find a coherent and consistent interpretation of the match in terms of a model

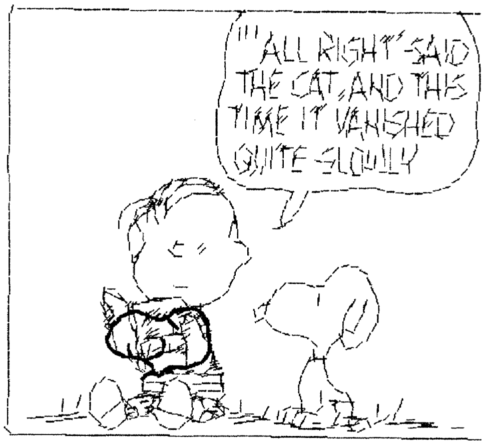
"ALL RIGHT" SAID THE CAT, AND THIS TIME IT VANISHED QUITE SLOWLY"

Figure 7-2: Bounded error recognition failing.

and an occluding object.

7.2.2 Minimum Description Length

One way to derive better statistical models of scenes is to use a "Minimum Description Length" (MDL) approach (Rissanen, 1978). In general, an MDL principle states that the optimal solution to a recognition problem can be found by describing or explaining the input data as concisely as possible in terms of a given (formal) language. The choice of language encodes the prior knowledge about the domain.

When applied to the problem of interpreting scenes, this means that we try to describe an unknown image as concisely as possible in terms of known objects and transformations on them.

MDL approaches are closely related to statistical methods, and, in particular, Bayesian methods. An MDL principle corresponds to a choice of prior probability distribution. The advantage of using an MDL principle is that the MDL principle is a much more convenient means for reasoning about higher-order statistical constraints than marginal probability distributions.

However, I do not want to advocate using the MDL principle directly for building recognition systems. This is because an actual implementation of an MDL approach

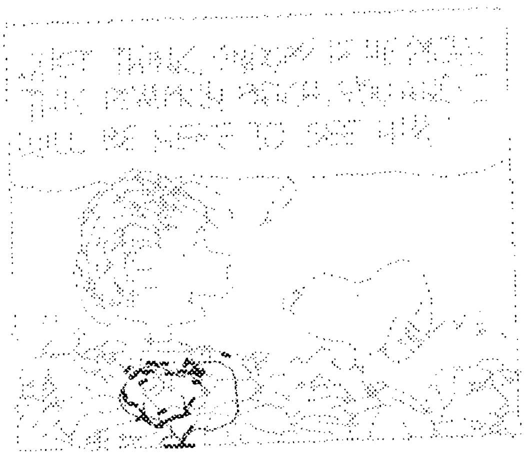

Figure 7-3: The matching image segments in a semantically incorrect but geometrically optimal bounded error match.

to recognition would present significant practical problems. First, finding an optimal or near-optimal description of an image in terms of some language can be a computationally hard problem. Furthermore, a complete interpretation of the image may be impossible because it might contain unknown objects. Finally, some occluding objects (e.g., trees) may not be describable concisely in terms of a deterministic language (however, extending the MDL principle to include non-deterministic grammars might be possible).

Another reason for not directly implementing a recognition system based on the MDL principle is that an actual recognition system is likely to use adaptive (learning) methods. But estimating probability distributions automatically is generally more convenient than grammatical inference.

Therefore, in what follows, we will be using the MDL principle primarily as a guide to understanding what kinds of statistical constraints might help improve the reliability of a recognition system. But actual implementations of recognition systems will be directly in terms of those simple statistical constraints.

7.2.3 Description Language

Let us examine informally what kind of description language we might want to choose for applying MDL principles to the problem of visual object recognition.

Different Description Languages for Objects and Occlusions

Our first simplification is therefore the following. Instead of describing the scene completely in terms of known objects and transformations on them, we attempt to describe only the part of the image that actually corresponds to a hypothesized match.

We then use the object model itself to try to explain as much as possible of the part of the image under consideration, and use a simpler, more general set of primitives to try to describe the occlusions that have given rise to the actual image of the object.

The Description of Occlusions

For the purpose of the argument and examples below, we wil consider a world in which occluding objects are mostly convex and opaque; by "mostly convex" I mean that they are convex or composed of a few convex objects–many real world objects satisfy these properties. A concrete description of occlusions might then be in terms of unions convex polygons.

Many other kinds of occlusions could also be expressed concisely in a MDL framework; for example, regular structures like streaks of rain or lines of a grid or network, can be described concisely in terms of simple primitive and repeated elements. For more irregular occluding structures like trees, even though the description of the detailed structure of the occluding object itself may be more costly, what is important is that the occluding structure can be adequately explained.

Transformations and Deformations

Above, we have pointed out an important relationship between grouping/segmentation constraints (expressed in terms of an MDL principle) on the one hand, and evaluation functions based on higher-order statistical properties, on the other hand.

An analogous relationship seems to exist between non-accidental properties (Lowe, 1986) and non-rigid transformations (deformations). In an MDL framework, we can handle non-rigid transformations by assigning short descriptions to common deformations (e.g., using an approach like the one described in Horowitz and Pentland, 1991).

However, intuitively, common deformations of objects tend to preserve non-accidental properties like "curvilinear collinearity", and a simple and practical approach towards matching under non-rigid transformations may be to test for the existence of such non-accidental properties among the features constituting a match and penalize matches in which non-accidental properties are not preserved. In what follows, we will not examine this particular aspect of the application of MDL principles to recognition.

7.2.4 Statistical Implications

Let us examine what some statistical implications of the MDL principle are for our example world in which occluding objects are mostly convex and opaque. For a correct match of a model to an image, the following properties should hold:

1. If some model feature matches, a nearby model feature is likely to match as well.
2. Edges (curved or straight) are unlikely to be broken into a large number of small edges by an occlusion. Hence, the number of edges in a hypothesized match should not be significantly larger than the number of edges in the model.
3. Within the convex closure of matching model features, spurious image features are unlikely (because occluding objects are usually opaque).

It is important to realize that these statistical principles that we have formulated as a derivative from our MDL considerations above do not involve global interpretations of the scene anymore, but simply apply to individual hypothesized matches.

In what follows, we will give several examples of how these principles can be applied in detail and how they can help improve the reliability of a recognition algorithm on some sample images. The images in the examples are cartoon line drawings (of the Peanuts). The cartoons have been converted into line drawings using morphological operations and thinning. The features that were used consisted of uniformly spaced samples from the boundaries with their associated orientations; that is, each cartoon image or model consists of a collection of points and associated orientations (usually of the order of 1000-2000).

7.3 Simplified Model

To illustrate the statistical principles suggested in this chapter, let us consider a simplified statistical model of recognition.

A recognition algorithm matches models against images. Usually, a model and an image is a collection of geometrical features associated with their geometric and visual properties (location, orientation, size, configuration, color, texture, etc.).

In our simplified view, models and images will simply consist of binary feature vectors. Each element of the feature vector indicates the presence or absence of an image feature at a particular location in the image.

This is actually not an unrealistic statistical model. For any particular pose (hypothesized transformation between model and image), the image and model can pretty much be described as binary feature vectors. The difference between our simplified view and the general problem of deciding whether a model is present in an image, nearby poses have statistically correlated feature vectors.

To express this model mathematically, let the features be given by boolean (random) variables $\{\phi_j\}_{j=1,\ldots,m} = \vec{\phi}$. Also, let there be n different kinds of objects, $\omega_1, \ldots, \omega_n$.

If we choose a Bayesian criterion for recognition, classification is carried out according to:

$$\omega = \arg \max_{\omega=\omega_1,\ldots,\omega_n} P(\omega|\vec{\phi}) \tag{7.1}$$

This basic framework provides a nice interpretation of the standard quality-of-match measures used in computer vision. Commonly used quality-of-match measures in computer vision are based on the number of model features that have a match in the image, or, similarly, the fraction of the boundaries or edges of a model that are accounted for in an image. We can easily see that this corresponds to assuming that the ϕ_j are mutually independent; assuming independence, we can write:

$$P(\omega_i|\vec{\phi}) = P(\omega_i|\phi_1, \ldots, \phi_m) = \prod_j P(\omega_i|\phi_j) \tag{7.2}$$

If we take logarithms on both sides, we get:

$$\log P(\omega_i|\vec{\phi}) = \sum_j \log P(\omega_i|\phi_j) \tag{7.3}$$

If we let $P(\omega_i|\phi_j) = \text{const}$, as is often the case in object recognition systems, this means that the quality of match measure is proportional to the number of matching features:

$$\text{quality of match} \sim \# \text{ matching features} \tag{7.4}$$

In the more general case, if we weigh image features differently, the weights we should choose are simply given by the logarithms of the conditional probabilities $P(\omega_i|\phi_j)$. These are marginal distributions of the true class conditional distribution $P_0(\omega_i|\vec{\phi})$. The class conditional distribution in Equation 7.3 is then the maximum entropy distribution consistent with these marginals.

To improve the performance or recognition systems, we need to extract more information from the matches between the model and the image. Formally, this means that we need to decrease the entropy of the distribution $P(\omega_i|\vec{\phi})$. Since the distribution in Equation 7.3 is the maximum entropy distribution consistent with the 1st order marginals, we can only reduce its entropy further by imposing higher-order marginal constraints.

7.4 2nd-Order Statistics

As we saw above, in a Bayesian framework, the usual quality-of-match measure corresponds to an independence assumption: the probabilities that each of two features is missing from an image are independent.

As we argued above, this view is likely to be a poor approximation for many classes of real images. In particular, under the assumption of mostly convex, opaque occluding objects, if some model feature is present in the image, it is more likely that a nearby model feature is also present in the image, because the object that occluded the first model feature will probably also occlude a nearby feature. This dependence can be captured using second order statistics.

The effect of this approach is that "locally coherent" matches are preferred over matches where model features are present and absent haphazardly over the whole model.

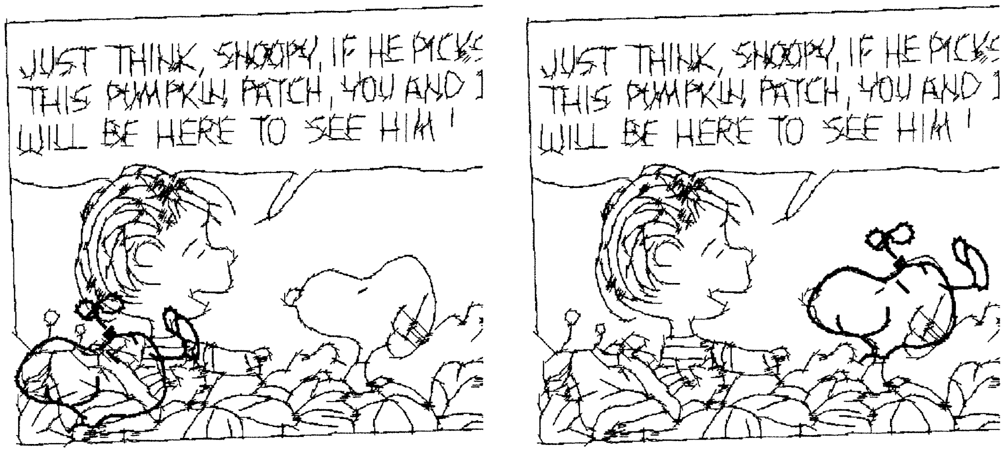
JUST THINK, SNOOPY, IF HE PICKS THIS PUMPKIN PATCH, YOU AND I WILL BE HERE TO SEE HIM!
JUST THINK, SNOOPY, IF HE PICKS THIS PUMPKIN PATCH, YOU AND I WILL BE HERE TO SEE HIM!

Figure 7-4: Improving recognition using second-order statistics.

We find that this approach will often lead to correct recognition where first order statistics fails. An example is shown in Figure 7-4. In this case, the parameters describing the second order statistics were simply guessed.

A plausible representation of statistics with specific first and second order moments is the log-linear representation. This is a good representation to choose because the maximum entropy distribution for given first and second order moments can be represented this way (see Goldman, 1987, for a discussion of these representations).

In the log-linear representation, we express the conditional probabilities as follows:

$$\log P(\omega_i|\vec{\phi}) = \sum_j \alpha_j^i(\phi_j) + \sum_{j,k} \beta_{jk}^i(\phi_j, \phi_k) + \text{const} \tag{7.5}$$

Recall that the ϕ_j take on values in $\{0, 1\}$. Each α function can therefore be characterized by two numbers, and each β function by four numbers.

In the experiments, a log-linear representation with estimated parameters was used. The α's were set to a constant, and the β's were chosen as

$$\beta_{jk}^i = \begin{cases} \text{const} & \text{if distance}(\phi_j, \phi_k) < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{7.6}$$

In this equation, distance(ϕ_j, ϕ_k) refers to the distance of the locations of the features ϕ_j and ϕ_k on the model.

For the experiments, α^i_j was chosen to be 0, and ϵ was set at 20 pixels.

7.5 Grouping/Segmentation Information

The method described in the previous section did not use any segmentation information. However, lower level vision modules often give us some segmentation information (albeit, usually incomplete and unreliable). For example, an edge detector will usually output long traces of curves in the image, and it is very likely that such traces come from a single object (or, rarely, from the merging of edges of two different objects).

Consider an unoccluded view of the object to be recognized. In an image of this view, the edge detector will find a number G of edge traces in the image. If the object is now partially occluded by another simple object, some edge traces will be occluded and others will be partially occluded. Usually, only a few curves will be split in two. "Good" matches of the object will therefore tend to involve no more curves in the image than are present in the model.

In order to use this idea, we use as a quality-of-match measure the number of model features that can be brought into correspondence with image features, subject to the constraint that the image features can be taken from at most G distinct curves.

An actual implementation is straightforward. We can use our usual algorithms to compute matches between the model and the image. Each match is then represented by a list of model features and what groups in the image contain image features that can match the model feature. Using a greedy algorithm, we can then select a subset of groups of size G that contain as many model points as possible.

An example of this approach is shown in Figure 7-5.

7.6 Occlusions

As we observed above, for opaque objects, there is another important constraint: portions of the object that are hypothesized to be unoccluded should not contain image features that are not explained by the model. Furthermore, for the kinds of occlusions that we are assuming, such regions will tend to have relatively simply boundaries (for more details on the shape of unoccluded regions by random collections of convex

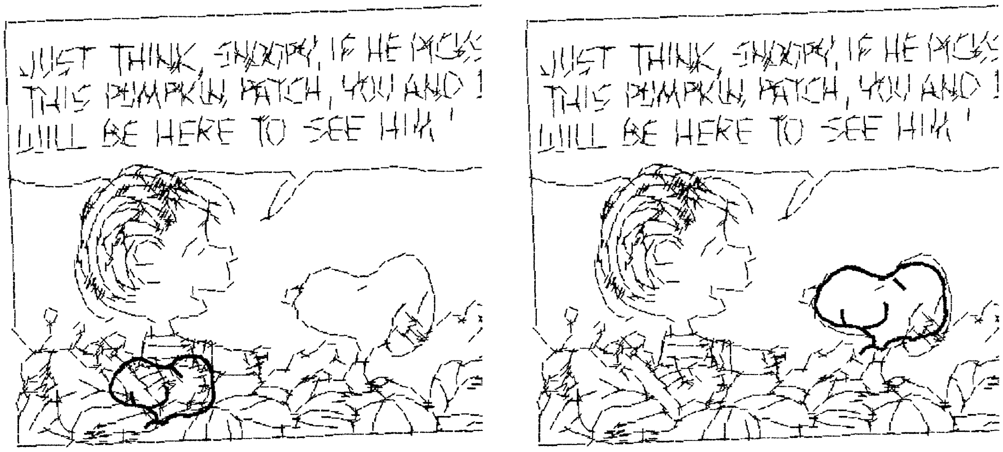
JUST THINK, SNOOPY, IF HE PICKS THIS PUMPKIN PATCH, YOU AND I WILL BE HERE TO SEE HIM!
JUST THINK, SNOOPY, IF HE PICKS THIS PUMPKIN PATCH, YOU AND I WILL BE HERE TO SEE HIM!

Figure 7-5: The use of grouping information to improve recognition.

objects, see Hall, 1988).

An example of using this constraint is shown in Figure 7-6.

7.7 Related Work

The methods we propose in this paper are related to grouping methods. The kind of constraints used for grouping are based on the higher-order statistics of features.

For example, Lowe, 1986, considers the probabilities that features are parallel or collinear in scenes of random line segments. Line segments in real images that show parallelism or collinearity that is unlikely to occur at random (i.e., that is non-accidental) are then argued to be like to have originated from the same object. Jacobs, 1988, uses similar constraints on small groups of line segments to argue that they bound a common region.

Traditionally, indexing algorithms (i.e., algorithms for speeding up recognition from large model bases) have required as input collections of features that come from the same object. Therefore, the main motivation for the development of grouping algorithms has been to provide such collections of features for input to an indexing algorithm. In different words, grouping has been used in a bottom up fashion and

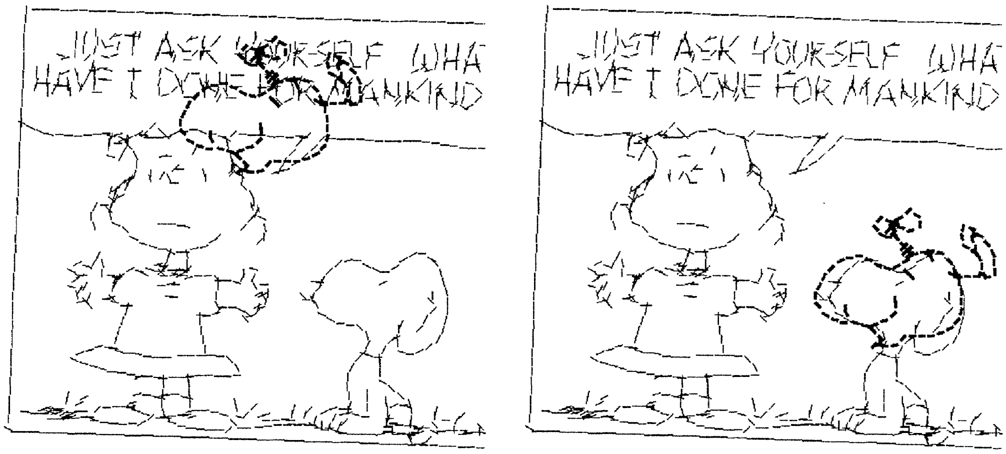

Figure 7-6: The use of exclusions to improve recognition.

primarily for reasons of efficiency.

But grouping also has a profound effect on recognition accuracy. If we exclude implausible groupings from ever being considered by the matcher, this will implicitly prevent the matcher from assigning a good quality-of-match measure to an implausible grouping. Jacobs, 1988, makes this point as well.

In fact, the significance of this observation goes much deeper: it shows us that we can use the information used by a grouping module to improve all aspects of recognition, not just to select small parts of an image for indexing. This chapter discusses how we go about doing this.

A number of other recognition systems have also been formulated in a Bayesian framework. However, the emphasis in most cases has been on error models for the location of individual features (Wells, 1992) or the likelihood that individual object parts or objects are present in particular views (Burns and Kitchen, 1988, Dickinson *et al.*, 1990, Mann and Binford, 1992).

Despite its obvious utility for reasoning about visual object recognition, the Minimum Description Length principle (Rissanen, 1978) has so far only been applied on low- and intermediate-level vision, primarily segmentation (Pentland, 1989, Pentland, 1990, Darrell *et al.*, 1990, Dengler, 1991, Keeler, 1991).

There has also been some work in trying to develop better measures of similarity among 2D shape. Such work has mostly concentrated on identifying "salient" features (e.g., Shashua and Ullman, 1988, Subirana and Richards, 1991). However, saliency is primarily a first-order constraint: different parts of an image or model simply are more or less important individually for the overall quality of match. Huttenlocher *et al.*, 1991a, has approached the question of meaningful 2D similarity measures from a geometrical point of view.

There is also a large body of relevant literature in pattern recognition and neural networks. In particular, a significant amount of research has been directed at devising methods that can learn or build "higher-order feature detectors" automatically; examples of this approach to vision are the TRAFFIC system (Zemel *et al.*, 1988), the Pandemonium model (Selfridge, 1959). While, ultimately, adaptive systems like neural networks are almost certainly needed, most neural network models are designed as "black boxes", without an understanding of the combinatorial and statistical structure and constraints of the vision problem. Therefore, the approach begun in this chapter is complementary to such learning approaches. We can hope that by understanding the higher-order structure of the visual recognition problem better in well-defined cases like the "mostly convex, opaque" world used above, we are guided towards better, more efficient, and more practical learning algorithms, rather than having to rely on luck and random search.

7.8 Discussion

In this chapter, we have formulated the problem of evaluating the quality of match in a statistical framework. We have seen some examples of how such statistical constraints can help improve the reliability of recognition algorithms.

Ad-hoc methods such as grouping, recognition by parts, and extraction of complex features like arcs have in the past been viewed primarily as means for reducing the amount of data that needed to be handled by a recognition algorithm, and they have been used primarily in a bottom-up manner. The statistical view of quality of match measures presented here promises to be very important for understanding and explaining the effects of such methods on the reliability of recognition algorithms.

A particularly important idea is that for reliable recognition, non-accidentalness and grouping information should probably be used in a top-down way; empirically, using such information bottom-up is a dangerous early commitment.

In addition, for the limited domain of opaque, mostly convex objects, we have seen three potentially useful enhancements to standard quality-of-match measures.

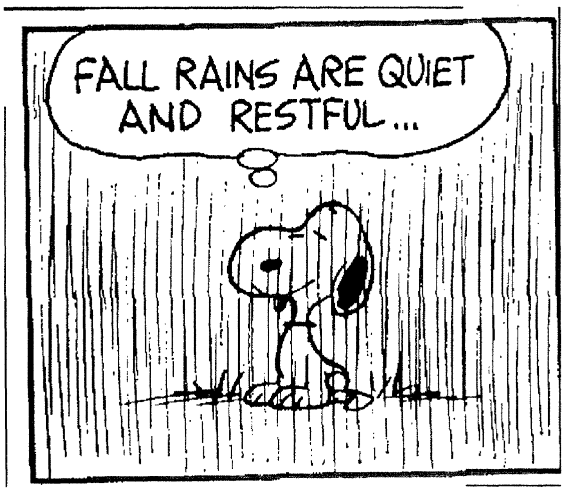

Figure 7-7: An example of an image where additional primitives are needed to describe the occlusion in the image.

The work presented in this chapter is, of course, preliminary. The motivation for presenting it here was to show where I believe the main emphasis of research in visual object recognition should be put. Even for the limited view of the problem of visual object recognition as the problem of determining geometric similarity, the current bounded error similarity measures are insufficient.

Chapter 8

Discussion

8.1 Reliability

Recognition systems will sometimes make mistakes and falsely indicate the presence of an object (false positive errors) or fail to find an object that is actually present in an image (false negative errors). The probability of false negative or positive errors determines the reliability of a recognition algorithm.

There are two fundamentally different causes for unreliability: either, the recognition algorithm solves the recognition problem exactly but is based on an incorrect world model (*modeling errors*), or the algorithm itself returns incorrect solutions occasionally (*internal errors*).

As a concrete example, consider a bounded error recognition algorithm. As we saw in Chapter 7, a bounded error recognition algorithm will often fail to find even simple instances of Snoopy in an image. However, the algorithm itself does find geometrically correct solutions to the recognition problem. Hence, the fact that the bounded error recognition algorithm is unreliable is due to poor modeling, rather than any kind of mistake made by the algorithm.

On the other hand, approximations and heuristics like the view-based approximation, quantization of transformation space, or early search termination (Grimson, 1990), cause a recognition algorithm to make internal errors occasionally.

In order to build reliable and economical recognition systems, we need to make careful tradeoffs between these two factors.

Probably one of the main approaches towards trying to improve the reliability of recognition algorithms has been to try to develop better geometric models. But I do not believe that better geometric models lead to more reliable recognition under most circumstances.

Empirically, passive sensing in natural environments seems to give only very approximate geometric measurements (for example, under changes of lighting, standard edge detectors like the Canny edge detector, can produce edges that move by several pixels in a 512×512 image). Furthermore, as soon as we attempt to recognize classes of objects (rather than specific instances), there is usually considerable variation in the shape of individual objects. Such effects imply an upper bound on the reliability of a recognition algorithm due to geometry (see Section 5 and Grimson and Huttenlocher, 1989), and there is little to be gained by reducing internal errors due to approximate geometric modeling beyond the bound due to fundamental limitations on the input to the recognition algorithm and due to limited prior knowledge about objects.

Note, incidentally, that for such practical reasons, the problem of the analysis of rigid motion and the problem of 3D rigid object recognition are very different, despite their superficial mathematical similarity.

Attempts to model the geometry of objects better probably has had an overall negative effect on the reliability of object recognition systems, for the following reason. In order to achieve better geometric modeling, recognition systems have had to rely on complicated 3D models and algorithms. In order to compensate for the high computational complexity of 3D matching algorithms, such systems have had to perform extensive bottom-up processing on the image. Concrete examples of such preprocessing are the extraction of features and the use of grouping. But such bottom-up processing implies an *early commitment* to particularly compact representations and features. Any error (usually, omission of an important feature or group) introduced early during processing can have serious consequences at later processing stages.

In this thesis, we have seen that a view-based approximation to the problem of 3D object recognition is no different from many other approximations that are made routinely and that do not seem to affect the reliability of recognition algorithms. Therefore, I believe that the 3D geometry of most kinds of visual object recognition is modeled sufficiently well by a view-based approach.

As we have seen in Chapter 6, using such simple, approximate approaches to 3D object recognition lets us use simpler, more robust features as the basis for recognition, and it lets us more easily use other, non-geometric means for improving reliability (2D similarity, learning, parts decomposition, nonrigidity).

8.2 Efficiency

As we noted in Section 4.2, one of the major concerns in the application of multi-view representations has been about the efficiency of such representations. The most common feeling was that multi-view representations require too much memory and too much time for matching. 3D models or interpolation methods are widely thought to be the best way to reduce the storage required for representing objects for object recognition.

However, it is important to realize that 3D models are only one way of compressing the information required for representing the different views of an object to sufficient accuracy. Furthermore, as we have argued above, 3D models are actually not particularly convenient representations for matching, since most of the 2D information required for matching is not made explicit in a 3D model.

If we take a strictly view-based approach to recognition, the efficiency problem then becomes one of representing a large collection of 2D images compactly and in a form suitable for matching. What makes this task simpler is the fact that the only use of object views in a strictly view-based approach to recognition is for matching; no explicit geometric information at all is required from any view. This means that we can compress the information in each view greatly and use a signature-based approach to recognition (e.g., Schwartz and Sharir, 1985). Furthermore, using suitable representations of signatures, we can achieve (average-case) sublinear access to large collections of signatures for 2D views (see Lamdan and Wolfson, 1988, Burns and Kitchen, 1988, Breuel, 1990, for different approaches to the problem of rapid access to large collections of views).

View-based recognition schemes are also ideally suited to implementation in massively parallel hardware, or even associative memory. In such an implementation, each processor can represent an individual object view and is responsible for identifying an object from a particular viewpoint. We can eliminate all computation (other than nearest-neighbor lookup) from each processor by carrying out transformations of the image and broadcasting transformed feature locations to each of the processors.

Therefore, the use of 3D models is only one of a number of possible approaches to trying to achieve compact representations of the collection of 2D views of a 3D object. Which method we choose should depend primarily on tradeoffs between memory requirements, efficiency, and robustness.

It is interesting to note that the representation by separate views of the Korean airliner and the Mig-21 used for the experiments reported in Chatper 6 required 89 kbytes and 58 kbytes, respectively. The representation of the outer hull of an X29 fighter jet by a CAD model required 74 kbytes. Therefore, it is not obvious that view-based recognition schemes necessarily require significantly more space than CAD models.

8.3 Pose Determination

View-based recognition can be used for pose determination. For example, using 32 views in a strictly view-based system, rotations out of the viewing plane can only be determined to within 20° (the 2D components of the viewing transformation can usually be determined with higher precision). Using 1000 views, the accuracy becomes 3.5°.

However, in certain industrial settings, when very high-quality visual input is available, and when we have reliable, accurate feature extraction methods available, we may want to determine pose more accurately.

In such a case, a hybrid approach is likely to be the best strategy: we can use a view-based recognition system to locate an object in the image and establish correspondences, and then use a conventional or interpolating pose estimation algorithm to optimize the pose estimate (such hybrid methods have been proposed in the literature; see Wallace and Wintz, 1980, and Poggio and Edelman, 1990, for example). Using the correspondences established by the strictly view-based preprocessing stage, an approximate local 3D model could even be built using the methods of Ullman and Basri, 1989, and Tomasi and Kanade, 1991, and used as input for the pose estimation (Wallace and Wintz, 1980, is essentially such a hybrid system, combining view-based recognition with interpolation, but for isolated objects and based on Fourier descriptors).

8.4 Features and Grouping

In actual recognition systems, in order to reduce the amount of information that needs to be processed by a subsequent recognition stage, edges in an image are commonly grouped in a bottom up way and then used to define more complex primitives like straight lines, circular arcs, regions, and vertices. As we have already observed above, such an approach implies an early commitment and tends to make recognition algorithms less robust.

In contrast, the experiments reported in this thesis were based on very simple sampled representations of the edges in an image. Such representations are not affected significantly by occasional failures of an edge detector.

However, apart from its (dubious) utility as a means of speeding up recognition algorithms, the identification of complex features and the use of grouping do have a very profound effect on the reliability of recognition algorithms, because they capture some very important statistical properties of the matching problem.

The approach to feature extraction and grouping suggested by the work in this thesis is then a *top-down* approach: grouping information is used directly in the comparison of individual models with an image. In such an approach to grouping, the grouping phenomena that are observed psychophysically would then not represent the result of a dedicated bottom-up processing stage, but the aggregate behavior of the matching of a large number of models.

8.5 Recognition by "Neural Networks"

The approaches to visual object recognition presented in this thesis are particularly well suited to implementation in terms of neural network hardware. In fact, most of the more formal analysis in this thesis was motivated by a simple neural network recognition system (Breuel, 1987, Breuel, 1989, Breuel, 1990) for 3D objects; we will return to a brief discussion of that system later.

Much of the extensive literature on "neural network models" of biological information processing tasks makes rather detailed assumptions about the computational abilities and deficiencies of real neurons and their interconnections. While I believe that such detailed assumptions are unwarranted, there are some general principles that are commonly accepted, and that express themselves in constraints on the kinds of circuits that can plausibly implemented in neural hardware. In particular, circuits implemented in neural hardware likely

- cannot have a great depth
- must settle within constant time
- must be robust to unit failure
- can only pass low-precision analog magnitude information (and possibly some phase information)

Furthermore, biological plausibility limits the kinds of neural circuits that can be constructed by a biological system. In particular, it is unlikely that large networks of neurons can rely on precise point-to-point wiring or on complete interconnections between individual units. Instead, local connectivity is likely to be randomized, while the more global structure of the network is probably determined by constraints on the interconnection of differently labeled cells and by the geometry (relative distance, ontogenical relation) of the modules that contain them.

Such hypotheses about the capabilities and limitations of biological neural networks make 3D model based approaches to recognition unappealing. This is because all known approaches to the 3D model based recognition problem require comparatively precise and non-robust computations.

In contrast, the methods for 3D object recognition presented in this thesis can quite easily be implemented in terms of hardware with properties like those listed above. This may not be immediately apparent and therefore deserves some discussion.

A major obstacle faced by implementing any recognition algorithm is that the appearance of an object on the image sensor is affected in a number of ways by rigid body transformations. Now, mathematically, it is not difficult to identify and undo those transformations for individual objects, for example, using a method like the alignment algorithm or the RAST algorithm. And, such algorithms are easily implemented in terms of digital computer hardware. However, expressing this mathematical concept in terms of noisy analog hardware is a much more difficult problem.

In order to alleviate this problem, we can try and take the following approach. First, we would like to be able to separate the transformational aspects of visual object recognition from the recognition aspects. That is, we would like to pre-process the visual input in such a way that subsequent recognition stages are only concerned with a simple (geometrically non-invariant) classification problem like that described in Chapter 7.

It is not difficult to see that for 3D object recognition, such a division of the recognition problem into a model-independent pre-processing step and a classification step is impossible. Recently, equivalent results have been stated as the non-existence of invariants (see Burns *et al.*, 1990, Moses and Ullman, 1991, Clemens and Jacobs, 1991). It must be emphasized that for the problem of indexing or object recognition with digital computers, this result is actually not of great significance, since (as we have seen in Chapter 4) using algorithms from computational geometry, we can still perform model base indexing in optimal time $\Theta(\log N)$ (where N is the number of models in the data base). The main significance of the observation really lies in the architectural implications for recognition systems built from "neuronal hardware" (i.e.,

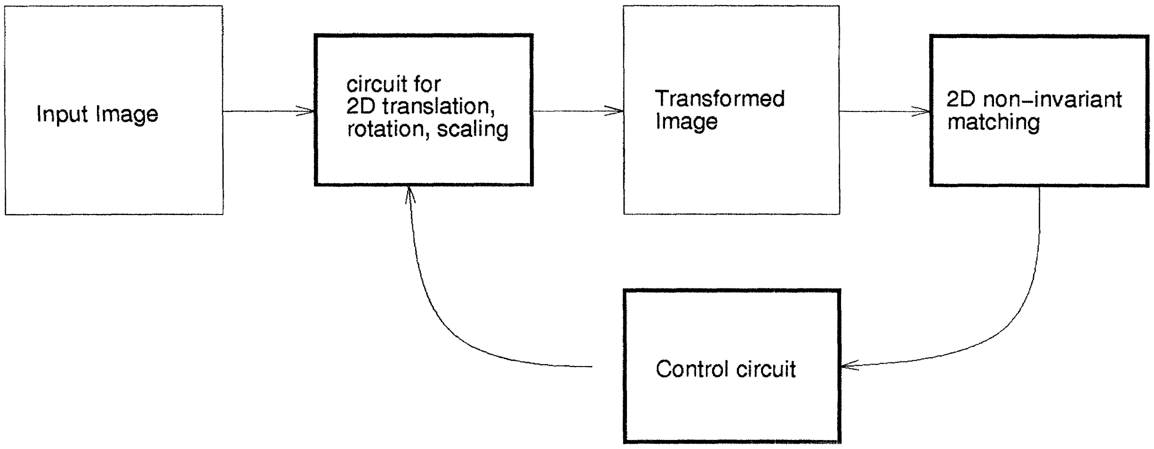

Figure 8-1: Schematic illustration of view-based recognition in neural hardware. The input image is transformed by a transformational module under 2D translation, rotation, and scale. The transformed image is using a recognition module that is not invariant under 2D transformations. An object is only recognized if it is presented to the recognition module under approximately the same 2D transformation under which it was previously stored. Recognition requires that many transformations be tried. It is likely that the output of the recognition module at each stage affects which transformations are likely to be tried next.

hardware with the properties described above) or simple associative memory chips.

Now, because we can canonicalize under 2D transformations, these observations naturally suggest that a recognition system built from "neural hardware" consist of a single, model-independent transformational module that canonicalizes under 2D transformations (2D translation, rotation, and scale), and a model-dependent recognition module that recognizes 2D views of 3D objects. This is illustrated in Figure 8-1.

2D canonicalization in the visual system could be implemented in terms of eye movements (fixation) and shifter circuits, both of which are well-suited to implementation in biological hardware (Andersen and van Essen, 1987). Such circuits apply 2D transformations to the input image and present the output to the recognition module. The 2D transformations that are applied to the input image before matching can be chosen using a variety of methods, analogous to methods for bounded error recognition systems: correspondence and alignment methods, and transformation space methods.

A correspondence-based approach uses pairs of point features to obtain a model-independent 2D canonicalizing transformation; this approach was used, for example, by Lamdan and Wolfson, 1988. But, as we have seen in Chapter 3, correspondence-based approaches tend to unnecessarily re-explore regions of transformation space. A more efficient hybrid approach might use well-separated key features (c.f. Julesz, 1981, Treisman, 1982) for canonicalization under 2D translation. Then, it might account for a limited amount of rotation and scale by simply trying transformations covering a small range of rotations and scales for each translation. This is very similar to the approach used in Chapter 6, and is a plausible model of certain aspects of human recognition.

Except for Chapter 7, the work described in this thesis has primarily been concerned with complexity issues related to the division of 3D recognition into 2D canonicalization followed by a recognition module. As we have seen in previous chapters, such an approach has recently been taken by a number of other recognition systems. The complexity results about the view-based approximation presented in earlier chapters apply to all these methods.

The way in which these systems differ is in the evaluation of individual matches. Lamdan and Wolfson, 1988, and use the standard bounded error evaluation function. Edelman and Weinshall, 1989, use a simple neural network for classification that minimizes overall classification error. Poggio and Edelman, 1990, use a correlational measure for recognition that takes into account some of the additional local geometric structure of images.

Despite their superficial differences, these systems all have in common that they define an optimal match as a match in which as much of an object can be brought into close correspondence with part of an image as possible. That is, they are essentially first-order evaluation functions, as defined in Chapter 7. Edelman and Weinshall, 1989, and Poggio and Edelman, 1990, argue that such evaluation functions can plausibly be implemented in neural hardware.

However, as the results in Chapter 7 suggest, higher-order evaluation functions are likely to be important for recognition. Proposals for neural implementations of higher-order evaluation functions can be found in Breuel, 1990. This method matches small object parts and combines the evidence in a variety of ways (Bayesian, linear combination, maximum probability). A similar approach has been taken by Brunelli and Poggio, 1991.

However, such approaches are probably too simplistic for real-world recognition systems. Realistic uses of higher-order statistics for object recognition will likely require a better understanding of the statistical properties of images, perhaps along the lines of the analysis presented in Chapter 7.

8.6 Human Vision

There exists now good evidence that certain aspects of human visual object recognition of 3D objects are based on comparisons of 2D views, and that human subjects have surprising difficulties to generalize from a training view, even if it contains complete depth information, to novel views of an object (Rock and DiVita, 1987, Rock *et al.*, 1989, Edelman and Buelthoff, 1990a). Such results provide strong support for the idea that human visual object recognition utilizes a viewer-centered representation of objects.

The results presented in this thesis significantly strengthen such theories of human visual object recognition: the thesis gives bounds on the number of views of a 3D object, establishes that the effects of the view-based approximation to 3D object recognition are comparable to those of the effects of other common approximations and unavoidable modeling errors, and demonstrates in an actual system that curved 3D objects can be recognized in 3D scenes using very simple view-based matching schemes.

The specific nature of viewer-centered representations used by the visual system remains an open question, however. Three competing theories are: strictly view-based representations (as described here), linear (convex) combinations (as suggested by Ullman and Basri, 1989), and non-linear interpolation (as suggested by Poggio and Edelman, 1990).

The performance of human subjects on the generalization from specific views of objects (even in the presence of depth information) to novel viewpoints is comparatively poor, and the experimental results presented here suggest that even strictly view-based recognition schemes might be sufficient to account for this level of performance. On the other hand, psychophysical results do indicate certain interactions between stored views that are consistent with an interpolation theory of human vision. But such results can also be interpreted in a strictly view-based framework if we assume a Bayesian interaction between views. This question should be resolved by future psychophysical experiments.

8.7 Conclusions

The first major result presented in this thesis, the RAST algorithm, is a very useful research tool. It represents a very efficient (probably in some sense "optimal") approach to the solution of many kinds of recognition problems. However, the RAST algorithm applies mainly to the case of object recognition from small model bases. With extensive pre-processing and/or for large model bases, there may be recognition algorithms available. An important general lesson from the RAST algorithm is that we should take explicit measures to avoid re-exploring similar transformations during the recognition process; failure to take such measures is responsible for the comparatively poor performance of alignment methods when compared with the RAST algorithm.

A major part of this thesis has been concerned with the view-based approximation to visual object recognition. We have seen that we do not need to use explicit 3D models in order to achieve geometrically sufficiently accurate 3D object recognition. This is a very general result and provides justification for a number of existing approaches to 3D object recognition. It also strongly suggests that future work on visual object recognition should mainly concentrate on questions other than geometry.

One such question, that I believe is of major importance, is the question of statistics. By this I mean the question of what kinds of properties scenes tend to have, and how such properties can be used to improve the reliability of recognition systems. In the third part of this thesis, I have outlined the beginnings of a statistical theory of scenes and shown in some simple examples, how the application of such statistical principles helps improve recognition.

Appendix A

Feature Extraction

A.1 Extracting Features from Grey-Level Images

The basic method for extracting features from grey-level images used in this thesis is:

- *Median filtering* (e.g., Lim, 1990). Median filtering removes "shot noise". It also tends to suppress irrelevant image detail without affecting edge localization too much.
- *Canny-Deriche edge extraction* (Canny, 1986, Deriche, 1987). The Canny-Deriche edge detector consists of several steps:
  - "optimal" IIR filtering
  - non-maximum suppression (a simple form of ridge detection on gradient-magnitude images)
  - noise estimation for adaptive setting of thresholds
  - hysteresis thresholding
  - thinning

Edges are then processed in a number of ways, depending on the particular application. The most commonly used processing steps for this thesis are:

- edge tracing (Ballard and Brown, 1982)
- estimation of local orientation

The last two steps deserve some more comments. The output of the tracing step is a list of points. For each point in this list, the point and its neighbors are considered (the neighbors are considered in a circular sense if the endpoints of the traced edge are sufficiently close to one another), and the principal direction for this point cloud is computed.

For each point, this gives a smoothed estimate of the local orientation of the line. This sequence of local orientation estimates is again smoothed by convolution with a 1D Gaussian; this second smoothing step is necessary because the quantization of pixel locations makes the estimates of local orientations also quantized. The local orientations are given a consistent direction using the direction of the gradient in the grey-level image.

We could use the set of all such edge pixels with local orientation estimates as image features. However, neighboring pixels will be highly correlated and provide very similar information about the image. To reduce the information further, we can proceed in several ways:

1. polygonal approximation
2. approximation by polygons and arcs
3. fixed scale polygonal approximation/sampling

All three have been implemented and used for the recognition experiments in this thesis.

Variable-scale polygonal approximation attempts to approximate edges in the image using extended straight lines. Commonly used algorithms are splitting algorithms and split-and-merge algorithms. Such polygonal approximations have the disadvantage that they are not stable on circular arcs. This problem can be helped by allowing both straight lines and circular arcs as primitives for approximating edges (Grimson, 1989b).

But increasing the inventory of matching primitives complicates later processing significantly. For example, a single straight edge in a model, when partially occluded, can give rise to two or more straight edges in an image. Furthermore, the geometrical constraints (see below) arising from pairing arcs and/or line segments are significantly more complicated than those arising from point features.

A simpler approach is to approximate edges using polygonal chains consisting of segments of fixed size. In practice, this means that we simply decimate (sample) the

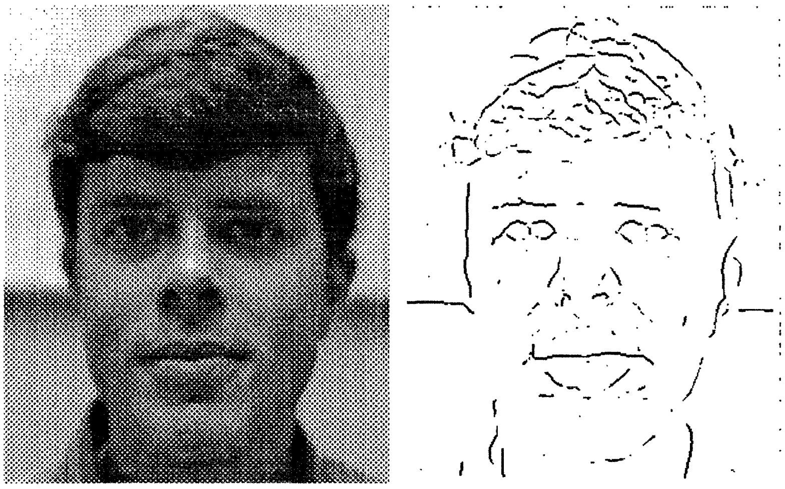

Figure A-1: Ridges are useful for visual object recognition.

sequence of points that make up the trace of an edge. When choosing the rate at which the original sequence of points is sampled, the spatial scale at which edge detection took place should be taken into account.

This results in a list of (usually) nearly equidistant points with associated orientations. For each such point, we can also retain a record of what chain the point belonged to; such simple grouping information is used by some of the algorithms discussed in Chapter 7.

This representation has turned out to be a good compromise. It is explicit enough to be reasonably stable and robust to occlusions, but it also reduces significantly the amount of input to later processing steps. Its main disadvantage is that it is not scale-invariant.

For some experiments and images, we have also used *ridge detection* (Fischler and Wolf, 1983, Horn, 1985, Riley, 1987).

A.2 Extracting Features from Binary Images

For many of the experiments, binary (bi-level) images, usually hand-drawn cartoons, were used as input. The later stages of processing are identical to those for edge detection. The process of arriving at the traces of edges, however, is different.

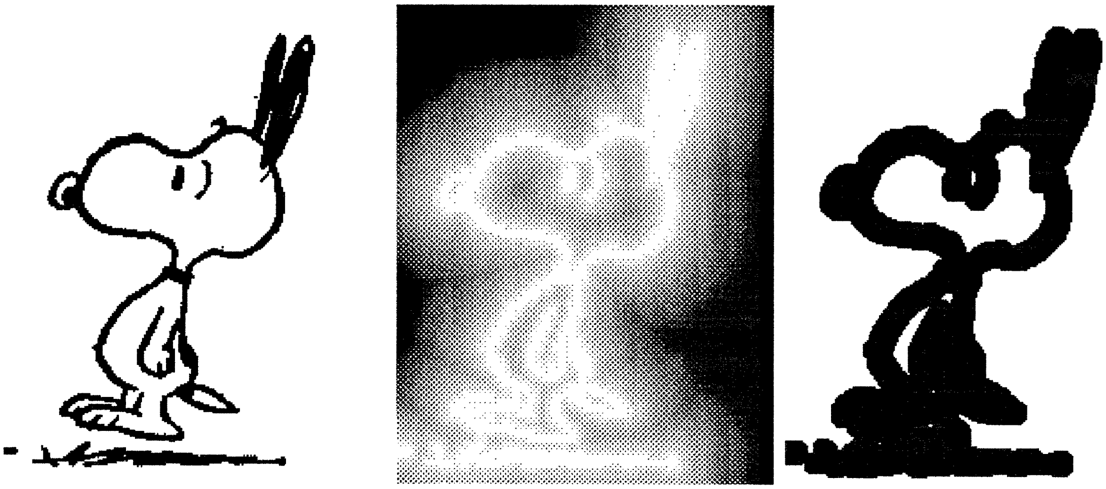

Figure A-2: An application of the Euclidean distance transform in the computation of dilations. The left image is the input, the middle image shows the distance transform, and the image on the right is the distance transform thresholded at 5, which is the same as a dilation with a disk of radius 5. The computation of the distance transform for this 212 by 266 pixel image takes about 2.5 seconds on a SparcStation 2 and requires 57 lines of C code.

Broadly speaking, binary images were segmented and features are extracted using *morphological operations* (Serra, 1982). The basic morphological operations used are thinning, dilation, erosion, opening, and closing.

The only structuring element used in this thesis is a circle, and it is easy to see that morphological operations with the circle as a structuring element can be implemented efficiently as thresholding operations on the *Euclidean distance transformations.*

A Euclidean distance transform of a binary image assigns to each pixel the distance of that pixel from the nearest pixel that is "ON".

For the efficient computation of the *Euclidean distance transform*, a dynamic programming approach is used. The same method works for any convex structuring element.

The basic technique is similar to a brushfire algorithm. Pixels on the front of the brushfire are maintained in a queue to make the method more efficient. However, unlike the brushfire algorithm in the natural norm on the square gird, the 1-norm, we maintain a record of the source pixel for each pixel in the distance transform, and we

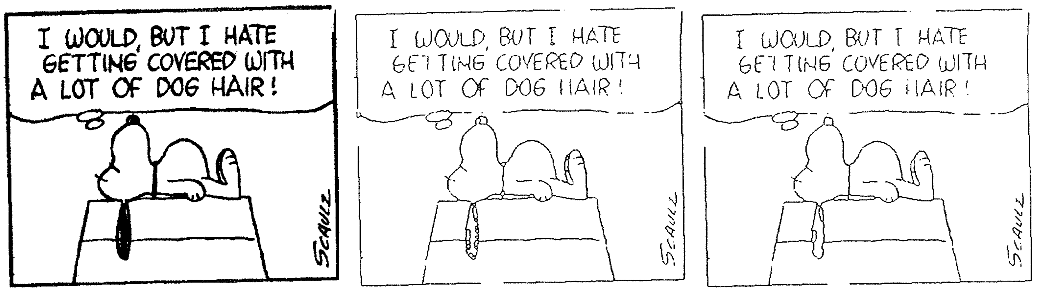
I WOULD, BUT I HATE GETTING COVERED WITH A LOT OF DOG HAIR!
SCHULZ
I WOULD, BUT I HATE GETTING COVERED WITH A LOT OF DOG HAIR!
SCHULZ
I WOULD, BUT I HATE GETTING COVERED WITH A LOT OF DOG HAIR!
SCHULZ

Figure A-3: Elimination of small "bubbles" in the thinned image using the method described in the text. On the left, the original binary image. In the middle, the image after morphological edge extraction, but with bubbles. On the right, the image after removal of small bubbles using the method described in the text.

continue to propagate pixels to their neighbors as long as the distance in the 2-norm from the propagated pixel is shorter than the original distance in the 2-norm for the target pixel.

This approach actually computes only an approximation to the distance transform. For our purposes, this fact is unimportant. In practice, the distance transform algorithm described above is very easy to implement, and quite efficient.

The three main computations for obtaining line drawings from cartoons are:

1. segmentation of a page of cartoons into individual boxes
2. conversion of dark regions into edges ("morphological edge extraction")
3. thinning of lines

For the segmentation, the scanned cartoon pages (about 3300 by 2500 pixels at 300dpi resolution) are subjected to a dilation operation to eliminate small gaps in the boxes surrounding the individual cartoon panels. Then, the background is filled. The dilated image is subtracted from the filled image. This gives masks for each cartoon panels. A bounding box for each mask is computed, and the resulting coordinates are used for cutting out the individual cartoon panels from the original scanned image.

For the morphological edge detection, a slightly eroded version of the image is subtracted from the original image. In the eroded version, lines in the original cartoon disappear and therefore preserved in the result. Larger, dark areas are transformed into their borders.

This method sometimes leads to the formation of spurious "bubbles" on edges (Figure A-3). The reason is that after the subtraction operation, there may be regions left in the image that have nearly parallel and touching boundaries. These artifacts can largely be eliminated by filling in all connected background regions with area smaller than a chosen threshold.

For thinning, a standard iterative algorithm was used (see Pavlidis, 1982).

References

Andersen C. H., van Essen D. C., 1987, Shifter circuits: a computational strategy for dynamic aspects of visual processing., *Proc. Natl. Acad. of Sci. USA*, 84:6297–6301.

Ayache N., Faugeras O. D., 1986, HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objecs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):44–54.

Baird H. S., 1985, *Model-Based Image Matching Using Location*, MIT Press, Cambridge, MA.

Ballard D. H., Brown C. M., 1982, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ.

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

Berger J. O., 1985, *Statistical decision theory and Bayesian analysis*, Sprinter Verlag, New York.

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

Bichsel M., 1991, *Strategies of Robust Object Recognition for the Automatic Identificaiton of Human Faces.*, PhD thesis, Eidgenössische Technische Hochschule Zürich.

Biederman I., 1985, Human image understanding: Recent research and a theory, *Comp. Vis. Graph. Im. Proc.*, 32:29–73.

Breuel T. M., 1987, 3D Recognition from 2D Views, Unpublished manuscript.

Breuel T. M., 1989, Adaptive Model Base Indexing, In *Proceedings: Image Understanding Workshop*, pages 805–814, Los Angeles, CA, Morgan Kaufmann, San Mateo, CA.

Breuel T. M., 1990, Recognition is Polynomial Time Reducible to Verification, A.I. Memo 1268, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Breuel T. M., 1991, An Efficient Correspondence Based Algorithm for 2D and 3D Model Based Recognition, In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition.*

Brooks R. A., 1983, Model-Based Three-Dimensional Interpretations of Two-Dimensional Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):140–149.

Brunelli R., Poggio T., 1991, Hyberbf networks for real object recognition, In *Proceedings IJCAI*, Sydney, Australia.

Burns J., Kitchen L., 1988, Rapid recognition out of a large model base using prediction hierarchies and machine parallelism , In *Intelligent Robots and Computer Vision. Sixth in a Series*, volume vol.848, pages 225–33.

Burns J., Weiss R., Riseman E., 1990, View variation of point set and line segment features, In *Proceedings Image Understanding Workshop*, pages 650–659.

Canny J. F., 1986, A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

Cass T. A., 1988a, Parallel Computation in Model-Based Recognition, Master's thesis, Massachusetts Institute of Technology.

Cass T. A., 1988b, Robust Parallel Computation of 2D Model-Based Recognition, In *Proceedings Image Understanding Workshop*, Boston, MA, Morgan and Kaufman, San Mateo, CA.

Cass T. A., 1990, Feature Matching for Object Localization in the Presence of Uncertainty, In *Proceedings of the International Conference on Computer Vision*, Osaka, Japan, IEEE, Washington, DC.

Chazelle B., 1985, Fast searching in real algebraic manifold with applications to geometric complexity, In *Proc. Coll. on Trees in Algebra and Progr. 1985, Lecture Notes in Comput. Sci. 185*, pages 145–156, Springer Verlag, Berlin.

Chin R. T., Dyer C. R., 1986, Model-based Recognition in Robot Vision, *ACM Computing Surveys*, 18(1):67–108.

Clarkson K. L., 1987, New applications of random sampling in computational geometry, *Discrete COmput. Geom.*, 2:195–222.

Clemens D. T., Jacobs D. W., 1991, Space and Time Bounds on Indexing 3-D Models from 2-D Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10).

Clemens D. T., 1991, Region based feature interpretation for recognizing 3d models in 2d images, Technical Report 1307, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Connell J. H., 1985, Learning shape descriptions: generating and generalizing models of visual objects, A.I. TR No. 853, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Darrell T., Pentland A., Sclaroff S., 1990, Segmentation by minimal-length encoding, In *Proceedings of the International Conference on Computer Vision.*

Davis P., 1975, *Interpolation and approximation*, Dover Publications, New York.

Dengler J., 1991, Estimation of Discontinuous Displacement Vector Fields with the Minimum Description Length Criterion, In *Proceedings: Computer Vision and Pattern Recognition*, IEEE Computer Society Press.

Deriche R., 1987, Using canny's criteria to derive a recursively implemented optimal edge detector, *International Journal of Computer Vision*, pages 167–187.

Dickinson S. J., Pentland A. P., Rosenfeld A., 1990, Qualitative 3d shape reconstruction using distributed aspect graph matching, In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition.*

Edelman S., Buelthoff H. H., 1990a, Viewpoint-specific representations in three-dimensional object recognition, Technical Report 1239, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Edelman S., Buelthoff H. H., 1990b, Viewpoint specific repreesentations in 3d object recognition, *Proceedings of the National Academy of Science.*

Edelman S., Weinshall D., 1989, A self-organizing multiple-view representation of 3D objects, A.I. Memo No. 1146, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

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

Ettinger G. J., 1987, Hierarchical object recognition using libraries of parameterized model sub-parts, Technical Report 963, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Falconer K., 1990, *Fractal Geometry*, John Wiley & Sons, England.

Fischler M., Bolles R., 1981, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, 24(6):381–395.

Fischler M. A., Wolf H. C., 1983, Linear delineation, In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 351–356.

Goldman S. A., 1987, Efficient methods for calculating maximum entropy distributions, Technical Report MIT/LCS/TR-391, MIT Laboratory for Computer Science, Cambridge, MA, USA.

Goldstein H., 1980, *Classical Mechanics*, Addison Wesley.

Grimson W. E. L., Huttenlocher D. P., 1988, On the Sensitivity of the Hough Transform for Object Recognition, A.I. Memo 738 1044, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Grimson W. E. L., Huttenlocher D. P., 1989, On the Verification of Hypothesized Matches in Model-Based Recognition, A.I. Memo 1110, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Grimson W. E. L., Lozano-Perez T., 1983, Model-Based Recognition and Localization From Sparse Range or Tactile Data, Technical Report A.I. Memo 738, MIT.

Grimson W. E. L., Huttenlocher D. P., Jacobs D. W., 1990, On the sensitivity of geometric hashing, Technical Report A.I. Memo 1250, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Grimson W. E. L., 1988, The Combinatorics of Object Recognition in Cluttered Environments using Constrained Search, Technical Report A.I. Memo 1019, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Grimson W. E. L., 1989a, The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments., A.I. Memo No. 1111, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Grimson W., 1989b, On the Recognition of Curved Objects in Two Dimensions, *IEEE Pattern Analysis and Machine Intelligence.*

Grimson W., 1989c, On the Recognition of Parameterized 2D Objects, *International Journal of Computer Vision.*

Grimson E., 1990, *Object Recognition by Computer*, MIT Press, Cambridge, MA.

Grötschel M., Lovász L., Schrijver A., 1988, *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, Heidelberg.

Hall P., 1988, *Introduction to the Theory of Coverage Processes*, John Wiley & Sons, New York, NY.

Horn B. K. P., 1985, *Robot Vision*, MIT Press, Cambridge, Mass.

Horowitz B., Pentland A., 1991, Reocvery of non-rigid motion and structure, In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition.*

Huber P., 1981, *Robust Statistics*, John Wiley and Sons, New York.

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

Huttenlocher D. P., Ullman S., 1987, Object Recognition Using Alignment, In *Proceedings of the International Conference on Computer Vision*, pages 102–111, London, England, IEEE, Washington, DC.

Huttenlocher D. P., Kedem K., Sharir M., 1991a, The Upper Envelope of Voronoi Surfaces and its Applications, In *Proceedings of the Seventh ACM Symposium on Computational Geometry.*

Huttenlocher D. P., Klanderman G. A., Rucklidge H. J., 1991b, Comparing Images Using the Hausdorff Distance Under Translation, Technical report, Cornell Computer Science Dept.

Huttenlocher D. P., 1989, *Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image*, PhD thesis, Massachusetts Institute of Technology.

Ikeuchi K., Kanade T., 1988, Applying sensor models to automatic generation of object recognition programs, In *Proceedings of the International Conference on Computer Vision*, pages 228–237, Tarpon Springs, FL.

Ikeuchi K. I., 1981, Recognition of 3-D Objects Using the Extended Gaussian Image, In *Proceedings IJCAI*, pages 595–600.

Jacobs D. W., 1988, The Use of Grouping in Visual Object Recognition, A.I. Technical Report No. 1023, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Julesz B., 1981, Textons: the Elements of Texture Perception, and their Interactions, *Nature*, 290:91–97.

Keeler K., 1991, Map representations and coding-based priors for segmentation, In *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (91CH2983-5)*, pages 420–5.

Koenderink J. J., van Doorn A. J., 1979, The internal representation of solid shape with respect to vision, *Biological Cybernetics*, 32:211–217.

Korn M. R., Dyer C. R., 1987, 3d multiview object representations for model-based object recognition, *Pattern Recognition*, 20(1):91–103.

Lamdan Y., Wolfson H.-J., 1988, Geometric Hashing: A General and Efficient Model-Based Recognition Scheme, Technical Report Technical Report No. 368, Robotics Report No. 152, New York University, Robotics Research Laboratory, Department of Computer Science, New York, NY.

Li H., Lavin M. A., LeMaster R. J., 1986, Fast hough transform–a hierarchical approach, *Computer Vision, Graphics, and Image Processing*, 36(2–3):139–161.

Lim J. S., 1990, *Two-dimensional Signal Processing*, Prentice Hall.

Longuet-Higgins H. C., 1981, A computer algorithm for reconstructing a scene from two projections, *Nature*, 293:133–135.

Lowe D. G., 1986, *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, Boston.

Mann W. B., Binford T. O., 1992, An example of 3d interpretation of images using bayesian networks, In *Proceedings Image Understanding Workshop.*

Marr D., Nishihara H., 1978, Representation and Recognition of the Spatial Organization of Three Dimensional Structure, *Proceedings of the Royal Society of London B*, (200):269–294.

Marr D., 1982, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H. Freeman and Company, San Francisco.

Megiddo N., 1984, Linear Programming in Linear Time when the Dimension is Fixed, *J. Assoc. Comput. Mach. (USA)*, 31(1).

Moses Y., Ullman S., 1991, Limitations of non model-based recognition schemes, Technical Report 1301, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Mundy J. L., Heller A. J., 1990, The evolution and testing of a model-based object recognition system, In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition.*

Norton J. P., 1986, *An Introduction to Identification*, Wiley.

Pavlidis T., 1982, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD.

Pentland A., 1989, Part segmentation for object recognition, *Neural Computation*, 1(1).

Pentland A., 1990, Automatic extraction of deformable part models, *International Journal of Computer Vision*, 4:107–126.

Poggio T., Edelman S., 1990, A network that learns to recognize 3D objects., *Nature*, 343(6255):263–266.

Ponce J., Kriegman D. J., 1989, On recognizing and positioning curved 3d objects from image contours, In *Proceedings Image Understanding Workshop.*

Ponce J., Kriegman D., 1990, Computing exact aspect graphs of curved objects: parametric surfaces, In *AAAI-90 Proceedings. Eighth National Conference on Artificial Intelligence*, pages 1074–9.

Ponce J., Hoogs A., Kriegman D., 1991, On using CAD models to compute the pose of curved 3D objects, In *Workshop on Directions in Automated CAD-Based Vision. (Cat. No.91TH0377-2)*, pages 136–45.

Reiss R.-D., 1989, *Approximate Distributions of Order Statistics*, Springer-Verlag.

Riley M. D., 1987, Time Frequency Representations of Speech Signals, Technical Report AI-TR-974, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Rissanen J., 1978, Modeling by Shortest Data Description, *Automatica*, 14:465–471.

Rock I., DiVita J., 1987, A case of viewer-centered object perception, *Cognitive Psychology*, 19:280–293.

Rock I., Wheeler D., Tudor L., 1989, Can we imagine how objects look from other viewpoints?, *Cognitive Psychology*, 21:185–210.

Russell B., 1921, *Analysis of Mind*, Allen and Unwin, London.

Schulz C. M., 1967, *You're Something Else, Charlie Brown*, Holt, Rinehart, and Winston, Inc.

Schwartz J. T., Sharir M., 1985, Identification of partially obscured objects in two and three dimensions by matching of noisy characteristic curves, Technical Report 46, New York University.

Selfridge O. G., 1959, Pandemonium: A paradigm for learning., In *The Mechanisation of Thought Processes*, London: H.M. Stationary Office.

Serra J., 1982, *Image analysis and mathematical morphology*, Academic Press, New York.

Shashua A., Ullman S., 1988, Structural saliency: the detection of globally salient structures using a locally connected network, In *Proceedings of the International Conference on Computer Vision*, pages 321–327, Tarpon Springs, FL, IEEE, Washington, DC.

Stein F., Medioni G., 1991, Structural hasing: Efficient three dimensional object recognition, In *Proceedings Image Understanding Workshop.*

Stockman G., Kopstein S., Bennet S., 1982, Matching Images to Models for Registration and Object Detection via Clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(3).

Stockman G., 1987, Object recognition and localization via pose clustering, *Computer Vision, Graphics, and Image Processing*, vol.40, no.3:361–87.

Subirana B., Richards W., 1991, Perceptual Organization, Figure-Ground, Attention and Saliency, A.I. Memo No. 1218, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Swain M. J., Stricker M., 1991, Promising directions in active vision, Technical Report CS 91-27, University of Chicago.

Tomasi C., Kanade T., 1991, The factorization method for the recovery of shape and motion from image streams, In *Proceedings Image Understanding Workshop.*

Treisman A., 1982, Perceptual Grouping and Attention in Visual Search for Features and for Objects, *Journal of Experimental Psychology: Human Perception and Performance*, 8:194–214.

Ullman S., Basri R., 1989, Recognition by Linear Combinations of Models, A.I. Memo No. 1152, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Ullman S., 1979, *The Interpretation of Visual Motion*, MIT Press, Cambridge and London.

Villalba M. J., 1990, *Fast Visual Recognition of Large Object Sets*, PhD thesis, Massachusetts Institute of Technology.

Wallace T. P., Wintz P. A., 1980, An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors, *Computer Graphics and Image Processing*, 13:99–126.

Weinshall D., 1991, Model based invariants for linear model acquisition and recognition., Technical Report RC17705 (#77262), IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY, USA.

Wells W. M., 1992, Posterior marginal pose estimation, In *Proceedings Image Understanding Workshop*, Morgan Kaufmann, San Mateo, CA.

Wilson S. S., 1990, Teaching network connections for real time object recognition, Technical Report 34, Applied Intelligent Systems, Inc.

Zemel R. S., Mozer M. C., Hinton G. E., 1988, Traffic: A model of object recognition based on transformation of feature instances, Technical Report CRG–TR–88–7, Department of Computer Science, U. of Toronto.