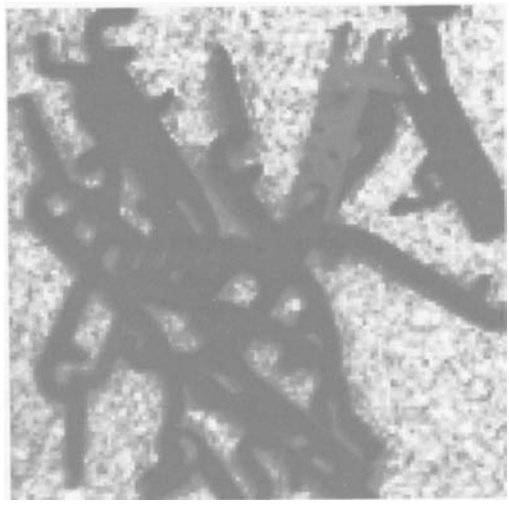
1 Introduction

If it is to interact intelligently and effectively with its environment, a robot must recognize and locate objects in that environment. Stated more informally, a robot often must use sensor data to determine *what objects* are in its environment, and *where* they are in that environment [Marr, 1982]. This holds true in a variety of tasks, including:

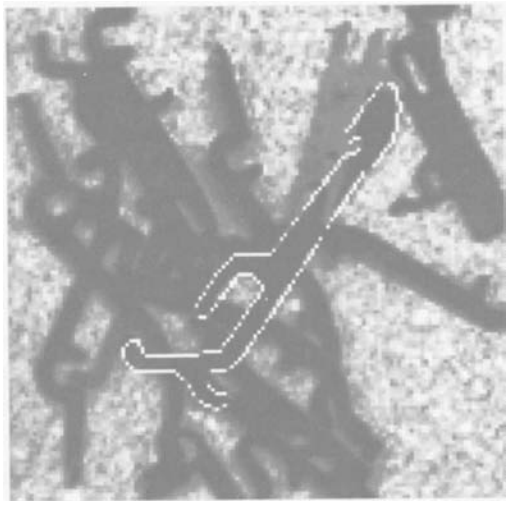
- identification of an object's pose in a cluttered environment, in order to pick it up and manipulate it, (for example, see Figure 1.1);
- inspection or gaging of an object, either to ensure that its components are present and correctly sited, or to measure and compare parts of an object against specifications, (for example, see Figure 1.2);
- vehicle navigation and localization, in order for a mobile robot to determine its position relative to a map of its world, (for example, see Figure 1.3).

All of these tasks involve either the problem of recognition – deciding which objects are present in the scene, or the problem of localization – determining the position of each object with respect to the sensor, or both. To determine *what objects are where*, that is to recognize and locate objects, one must have information about the environment, but simply acquiring sensory input is not sufficient in itself to solve this problem. Sensory data usually only provides measurements about properties of objects in an environment, for example, distances from the sensor to points in the world, or the location of edges of objects relative to the sensor. This alone is not sufficient to tell a robot what it is seeing. For that, the robot must also *interpret* those sensory measurements, that is, the robot must relate those measurements to knowledge it has about objects in its domain of experience, in order both to identify instances of such objects in the world, and to determine their location and orientation relative to the robot. This monograph describes an investigation into the problem of sensory interpretation, and more specifically, into the problem of object recognition and localization.

There are many aspects to the subject of object recognition, including the following:



a.



b.

Figure 1.1
An example of a localization task. Given an image of a jumble of parts, as in part (a), a robot must determine if an instance of an object model exists in the data and if so the position of the object, as shown in part (b). It must do so in the presence of noisy data, occlusion of the object, and clutter (or spurious data) in the scene. Figure courtesy of Todd Cass.

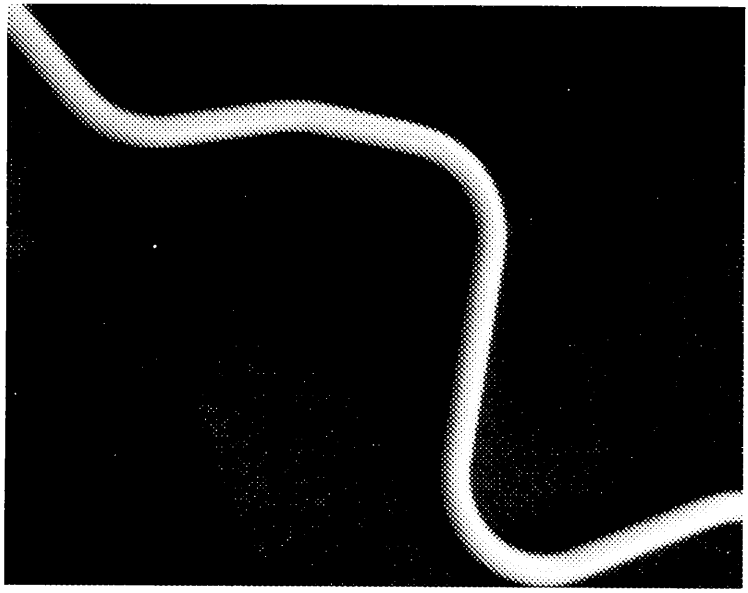
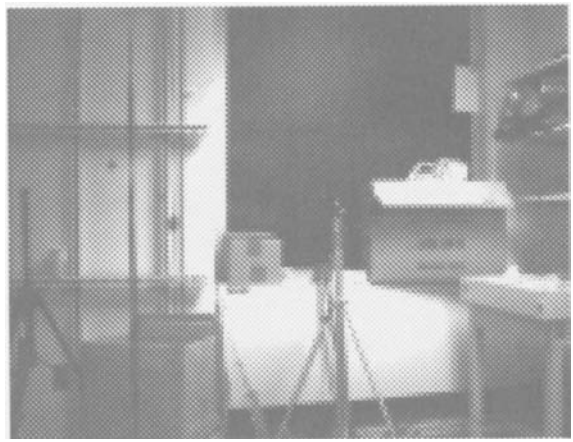
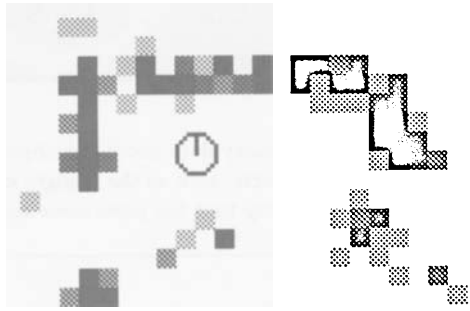


Figure 1.2
An example of a gaging task. Given sensory data about the object, a vision system must identify and measure individual parts, such as the straight cylindrical sections and toroidal bends of the tube, and verify that the parts meet some desired specifications.

- what sensory cues are used to recognize candidate objects?
- how should one represent information about individual objects?
- how are large libraries of objects efficiently stored?
- what indexing methods are used for extracting candidate objects from such libraries?
- what methods are used to establish a correspondence between sensory cues and features of an object?
- how do we deduce the position and orientation of an object from sensory cues?
- how are new objects learned and added to the library?
- what role does attention play in selecting portions of the sensory data on which to concentrate?
- what role does context or expectation play in aiding recognition?
- what role does an object's function or use play in aiding recognition?



a.



b.

Figure 1.3
An example of a navigation task. Given sensory data about a room, such as a sequence of images of the form shown in part (a) taken as the robot spins in place, a robot must interpret that data relative to a map of the world, to determine its global position. Part (b) shows an example in which a sequence of stereo images have been processed to obtain 3D information about the room, which has then been projected into a 2D ground plane. This representation of the scene has been matched with a model of the world that the robot has learned over time, in order to deduce the robot's location, as indicated in the figure. Figure courtesy of David Braunegg.

Our goal in this book is not to attempt a definitive answer to all of these aspects of recognition, although we will touch on most of them to some extent. Rather, our goal is to consider components of the recognition problem while describing a detailed exploration of one aspect of object recognition. Specifically, we want to investigate the following question:

- What is the role of geometric measurements and constraints in object recognition and localization?

What do we mean by geometric measurements and constraints? We are interested in understanding how the shapes of objects can be used to determine which objects from a library of possible objects are actually present in a scene, to determine the correspondence between data features and object features, and to determine the pose of the object in the scene. The geometric measurements are intended to capture aspects of an object's shape as perceived by a sensor, and any changes in that perceived shape as the object is transformed in the scene. We will see some instances of geometric measurements of shape in the simple example of Section 1.2. We will be particularly interested in using measurements of shape that are invariant under the set of allowed transformations, as these measurements will sharply constrain the solutions to all three subproblems: the set of possible object models in the scene, the set of possible correspondences between scene features and model features, and the set of possible poses of an object in the scene.

Thus, this monograph describes an extended series of experiments into the role of geometric measurements in object recognition. This description will include providing precise definitions of the recognition and localization problems, as well as descriptions of the methods used to address them. We will also examine the performance of these methods, both on controlled synthetic data and real data, and we will provide a formal analysis of our solutions to the problems. This analysis will enable us to address implications of such methods.

Although we focus on the role of geometry in object recognition, this is not to imply that the methods discussed here are so restricted in scope as to have no practical import. Indeed, the problems solved by the described methods are of fundamental importance in many real applications, and versions of the techniques described here are already in use in industrial settings. Thus, while many questions remain to be solved before a completely general solution to the recognition problem is available, by exploring in detail the role of object shape in recognition and localization we provide a framework for understanding both the strengths and limitations of using object shape to guide recognition. This serves both to provide a basis on which to build more complete recognition systems, and a means of identifying which parts of the recognition problem remain as the main stumbling blocks to such complete systems.

1.1 A definition of the recognition and localization problem

To set the stage for the discussion to follow, we first provide a definition of the problem to be considered, and then provide a simple example to illustrate the particular subproblem of using geometric constraints in recognition and pose localization.

The goal of the recognition systems to be described is to identify which objects are present in a scene, and to determine the pose of each object relative to a sensor. By *pose*, we mean the transformation needed to map an object model from its own inherent coordinate system into agreement with the sensory data. Because this is a very broad problem, we will restrict our attention to a narrower, and more tractable, version of the problem, as outlined below.

1.1.1 Rigid objects

To begin with, we will consider only rigid objects. By rigid, we mean that the distance between any two points on the object remains the same, as the object undergoes any legitimate transformation in the scene. Other types of objects exist, for example, articulated objects, like a pair of scissors, or flexible objects, like a snake, or deformable objects, like modeling clay or jelly. We will focus most of our attention on the simpler problem of recognizing and locating rigid objects, though in Chapter 16 we will consider extensions of the methods we develop to deal with some types of articulated and deformable objects.

One consequence of concentrating on rigid objects is that the pose of an object is constrained to a small number of parameters, e.g. the position and orientation of a distinguished point on an object, as measured in a coordinate frame centered about the sensor.

For example, suppose we are considering flat rigid objects constrained to lie on a known support plane. For such objects, the pose is determined by two translational parameters, one rotational parameter, and possibly a scale factor, if the object is allowed to change overall size (see Figure 1.4). If we are considering more general rigid objects, then three translational and three rotational parameters, as well as possibly a scale factor, are needed to specify the pose of an object (see Figure 1.5).

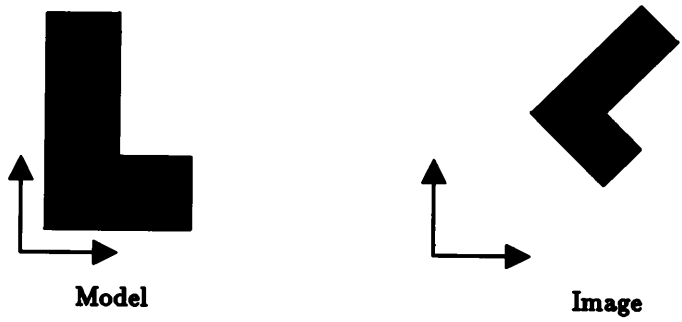


Figure 1.4
The object model on the left has an instance in the simple image on the right, and is specified by a two dimensional translation, a rotation in the plane and a scaling.

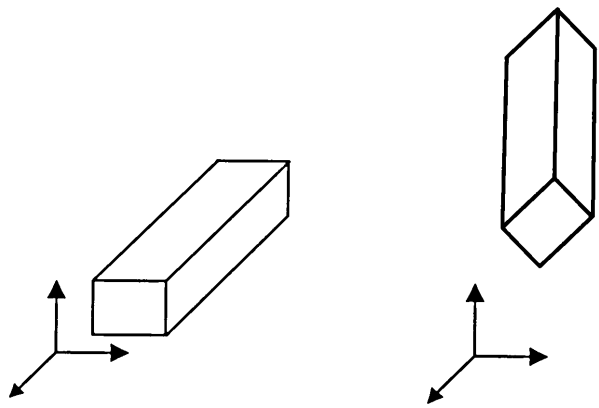


Figure 1.5
The object model on the left is specified in the scene on the right by a transformation consisting of translation along three orthogonal axes, a three degree of freedom rotation, and a scaling.

A second consequence of concentrating on rigid objects is that measurements of an object's shape that are invariant under the class of legal transformations (in this case rotations and translations) are considerable simpler to compute and apply, as we will see shortly.

1.1.2 Model-based recognition

Even if we restrict ourselves to rigid objects, we still must consider what information we will use to characterize an object to be recognized. Since

we are focusing on geometric measurements, we will assume that we have shape information about particular objects available for comparison with data from the scene. Thus, we will focus on the problem of *model-based* recognition. This requirement restricts our problem domain to some extent, since it requires a specific model for each object of interest. By comparison, suppose we want our system to recognize chairs. Humans are quite adept at identifying instances of an object with very different appearances but similar function. As observers, we can easily identify a chair as such, whether it be Chippendale or Louis XIV. To some extent, this identification is based on generic shape properties and their relationship to the function of the chair. By the problem definition given above, in our investigation we must have a different model for each different shaped chair. Hence, recognizing a new kind of chair on the basis of similarity of structure and function is beyond the scope of our system.

1.1.3 Problem definition

With these restrictions in mind, we can more formally characterize the problem to be investigated. In our study, we will assume the following information is available:

- A model (or library of models) of the object(s) of interest. Each model must describe the shape of an object, although, as we shall see, this description need not be complete, nor need it be an exact representation of the shape.
- A set of sensory measurements about the environment being examined. These measurements are assumed to capture geometric information about the position and orientation of pieces of surfaces in the world.

Given such input, the recognition system is expected to produce the following output:

- The set of "feasible interpretations" of the data with respect to the object(s). By an *interpretation*, we mean both an identification of the correspondence between data elements and parts of the object model, and an estimate of the coordinate frame transformation needed to transform the model from its own inherent coordinate frame into the sensor coordinate frame. By *feasible*, we mean that applying the transformation to the model would cause the elements of the model to appear in the sensor coordinate frame in positions

commensurate with the data elements identified with them in the correspondence. Note that we ask for the **set** of feasible interpretations, as there may be more than one feasible interpretation, either because more than one object from a library is consistent with the data, or because more than one pose of a single object is consistent with the data.

1.2 A simple example

Our goal is to understand how geometry (i.e. the shapes of objects) can be used to constrain the set of solutions to the recognition and localization problems. To illustrate the role of geometric constraints in finding poses of an object consistent with the available data, we provide the following simple example. Suppose we consider the simple two dimensional object shown in Figure 1.6. Further, suppose we are given a set of data features, such as the bounding edges shown in Figure 1.6. To determine feasible poses of the object consistent with the data, we must determine correspondences between the data edges and the model edges, i.e. pairings of data edges with model edges.

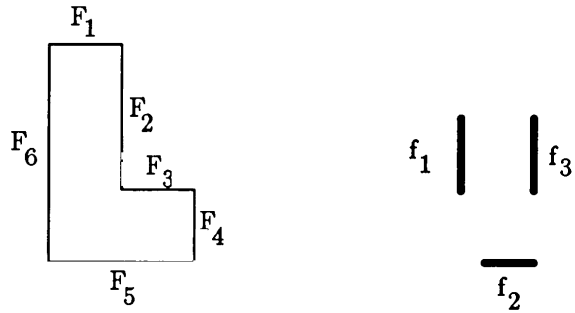


Figure 1.6
On the left, a simple two dimensional object, described by its bounding edges. On the right, a set of data edges, extract from some scene of an object. The goal is to deduce the pairing between the data edges and the model edges.

In principle, we could simply consider all possible such correspondences, and determine if there is a coordinate frame transformation that maps the model into the data in agreement with the correspondence. Even for such a simple method, there are a number of possible variations. The

most naive approach might be to correlate the model with the image. By this we mean:

- Take all possible rotations of a model, sampled at some resolution such as in increments of a degree;
- Then take all possible translations of those rotated models, sampled at some resolution such as every pixel in the horizontal and vertical directions,
- Then overlay each such rotated, translated model on the image, and count the number of edge points in the model that overlap an edge point in the image
- Keep the rotated, translated model that has the highest overlap with the image.

While this method is straightforward to visualize, for any realistic sized problem, however, it is prohibitively expensive. For example, at the suggested sampling rates, one would need

$$500 \times 500 \times 360 = 90,000,000$$

different overlays of a single model with an image, and for each of these, one must still count the number of overlapping edge points. Thus, we need more efficient ways of restricting our search for consistent correspondences.

One way to do this is to focus on the features in the image. For example, we could consider all possible ways of pairing data features f_i with model features F_j, and then testing the degree to which those correspondences are consistent with the geometry of the objects. By pairing, we mean asserting that a data edge is a visible instance of a part of a model edge, and by a correspondence we mean a collection of pairings, one for each data edge. While this will avoid some of the wasted computation of the simple correlation method (by avoiding situations with no overlap between model and image features), it can still be expensive. If we don't know which side of a data edge is the inside of the object, then each pairing f_i to F_j has two orientations, and hence there are $(2m)^d$ possible correspondences to consider, where d is the number of data features, and m is the number of model features. In the simple case shown in Figure 1.6, there are 1728 different correspondences to consider. If we do, in fact, know inside from outside (e.g. by measuring the contrast across an edge), then there are only m^d cases to consider, but even in the simple example of Figure 1.6 this still leaves 216 cases. Clearly, many of these correspondences are not likely candidate solutions, since they

don't make geometric sense (i.e. there is no rigid transformation of the model that would satisfy all of the pairings in the correspondence). We would like to find a way to quickly remove them from consideration.

To do this, we will use geometric constraints on the matching process. In particular, the relative shapes of subsets of the data clearly preclude many possible correspondences. For example, if we match data edge f_1 to model edge F_1 (in the case in which we know which side of the edge is the inside of the object), then data edge f_2 cannot possibly match model edge F_5, because F_1 and F_5 are parallel, and f_1 and f_2 are not. Thus, we should not bother considering any interpretations that include both the pairing $f_1 : F_1$ and the pairing $f_2 : F_5$, unless the data is very noisy. In other words, the relative shape of data edges f_1 and f_2 is sufficiently different from the relative shape of model edges F_1 and F_5 that the pairing $f_1 : F_1$ and the pairing $f_2 : F_5$ cannot both be part of a consistent interpretation.

To illustrate how we can take advantage of these constraints on relative shape, we sketch a simple set of geometric constraints, and show their use in finding correspondences. The details of such methods will be expanded upon in detail in the succeeding chapters.

	F_1	F_2	F_3	F_4	F_5	F_6
F_1	0	$\frac{3\pi}{2}$	0	$\frac{3\pi}{2}$	π	$\frac{\pi}{2}$
F_2	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	0	$\frac{3\pi}{2}$	π
F_3	0	$\frac{3\pi}{2}$	0	$\frac{3\pi}{2}$	π	$\frac{\pi}{2}$
F_4	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	0	$\frac{3\pi}{2}$	π
F_5	π	$\frac{\pi}{2}$	π	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$
F_6	$\frac{3\pi}{2}$	π	$\frac{3\pi}{2}$	π	$\frac{\pi}{2}$	0

Table 1.1.
Table of relative angles for the object in Figure 1.6. For each entry, the angle listed is that required to rotate the edge identified by the row index into the edge identified by the column index.

First, we can capture part of the relative shape of two edges by considering the relative angle between them, which is an extension of the idea used above. For example, we can build a table of relative angles for all pairs of edges in the model. Such a table is shown in Table 1.1, where

the angle listed for each pair of edges is that angle needed to rotate the edge labeled by the row index into the edge labeled by the column index.

We can build a similar table for the relative angles between the data edges, shown in Table 1.2.

	f_1	f_2	f_3
f_1	0	$\frac{\pi}{2}$	π
f_2	$\frac{3\pi}{2}$	0	$\frac{\pi}{2}$
f_3	π	$\frac{3\pi}{2}$	0

Table 1.2.
Table of relative angles for the data edges shown in Figure 1.6.

We need to find possible correspondences, using the information contained in these tables to restrict our search to viable possibilities. Suppose we begin by considering the first data feature f_1, and we consider the possibility that this data feature actually corresponds to each of the model features, $F_i, i = 1, \ldots 6$, in turn. For each such correspondence, we take the second data feature f_2, and consider the possibility that it corresponds to each of the model features in turn, $F_i, i = 1, \ldots 6$. This is shown by the trees in Figure 1.7.

Now given these pairings, we can use our information about relative shapes to constrain the search for interpretations. In particular, for each node at the second level of the tree, we can use the indices for the two model features to look up the relative angle between them in Table 1.1. For the two data features, we can use their indices to look up their relative angle in Table 1.2. If these two angles do not agree, then the shapes are inconsistent, and we can drop this partial interpretation from further consideration. By inconsistent, we mean that there is no rigid transformation that will map the data into the model, since the relative angle is an invariant under rigid transformations and must therefore be maintained in both the data and the model edges. In other words, the relative shape of the features constrains the possible matches. This is shown in Figure 1.8, in which nodes inconsistent with this simple test are crossed off. This reduction in the number of possible interpretations has been achieved by using the geometric constraint that relative angles between edges must be preserved in an object's pose, i.e. the relative angle between two edges is pose invariant for rigid objects.

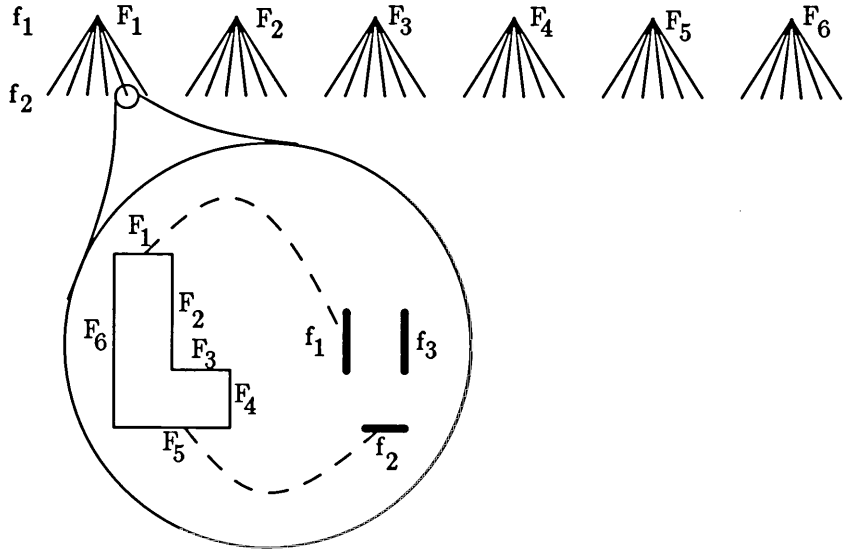


Figure 1.7
For the first data feature, f_1, we consider all possible pairings of it to each model feature, $F_i, i = 1, \ldots 6$. For each such pairing, we consider the assignment of the second data feature, f_2, to all possible model features, $F_i, i = 1, \ldots 2$. This is shown in the diagrammed tree structure, where each node at the second level defines a matching for the first two data features. The node identified by the circle corresponds to the pairing of data and model features shown in the expansion.

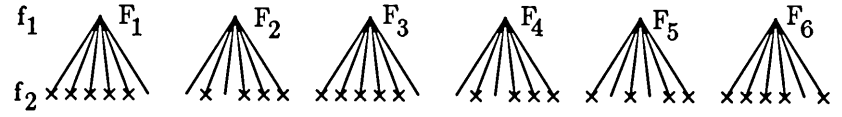


Figure 1.8
Any nodes at the second level of the tree that are inconsistent with the partial geometric shape constraints are removed from further consideration.

For the remaining interpretations, we can now consider the third data edge, and its possible assignment to each of the model edges. As before, we can subject the resulting interpretations to our geometric constraints, where now the pairing of the third data edge with a model edge must be consistent with both the pairing of the first data edge to some model edge and with the pairing of the second data edge to some model edge. The resulting set of consistent nodes is shown in Figure 1.9.

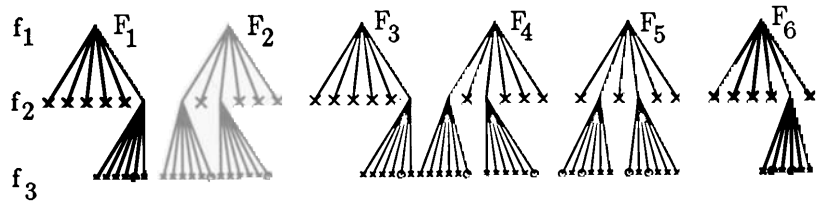


Figure 1.9
The set of nodes defining an interpretation for all three data edges that are consistent with relative angle constraints are shown circled, those inconsistent are crossed off.

As one can see, just using this simple measurement on the relative angle between edges has dramatically reduced the set of possible interpretations of the data, from 216 to 12, in this particular case.

Although the use of relative angle dramatically reduces the search involved in finding a solution, we can do more than this. Suppose we also consider distance constraints, that is, information about the distances between pairs of edges. As in the case of angle information, we can compute a table of relative distance information for the object model. Table 1.3 lists the range of squared distances between pairs of model edges, for example, the shortest distance between points on face F_1 and F_3 is $2(=\sqrt{4})$ and the longest distance between points on face F_1 and F_3 is $2\sqrt{2}(=\sqrt{8})$. We use the square of the distances rather than the distances themselves simply for convenience of representation.

	F_1	F_2	F_3	F_4	F_5	F_6
F_1	[0, 1]	[0, 5]	[4, 8]	[5, 13]	[9, 13]	[0, 10]
F_2	[0, 5]	[0, 4]	[0, 5]	[1, 10]	[1, 10]	[1, 10]
F_3	[4, 8]	[0, 5]	[0, 1]	[0, 2]	[1, 5]	[1, 8]
F_4	[5, 13]	[1, 10]	[0, 2]	[0, 1]	[0, 5]	[4, 13]
F_5	[9, 13]	[1, 10]	[1, 5]	[0, 5]	[0, 4]	[0, 13]
F_6	[0, 10]	[1, 10]	[1, 8]	[4, 13]	[0, 13]	[0, 9]

Table 1.3.
Table of ranges of squared distances between edges of the model in Figure 1.6. Each entry lists the minimum and maximum squared distance between any two points on the indicated edges.

A similar table holds for the range of squared distances between pairs of data edges, shown in Table 1.4.

	f_1	f_2	f_3
f_1	[0, 1]	[1, 5]	[1, 2]
f_2	[1, 5]	[0, 1]	[1, 5]
f_3	[1, 2]	[1, 5]	[0, 1]

Table 1.4.
Table of ranges of squared distances between the data edges of Figure 1.6.

We can add these new constraints to our search, by applying them in a manner similar to that used for the angle constraints. That is, for a given pair of data-model pairings, we can look up the range of feasible distances for the model edges from Table 1.3, and look up the range of distances for the data edges from Table 1.4. In order for these pairings to be consistent, the range of data distances must be contained within the range of model distances. By applying these constraints, we reduce the search tree to that shown in Figure 1.10.

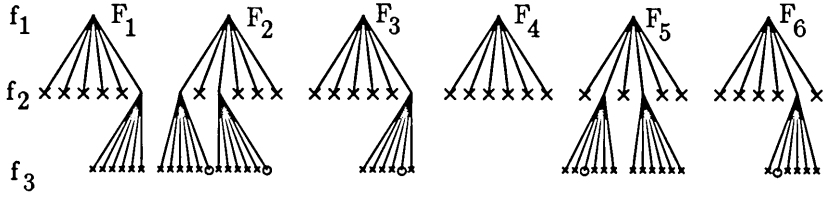


Figure 1.10
The set of nodes defining an interpretation for all three data edges that are consistent with relative angle constraints and with relative distance constraints.

This further restricts the set of feasible interpretations. Note, however, that these geometric constraints serve only to hypothesize feasible interpretations. We must still determine the pose associated with each interpretation and verify that it is globally consistent. In fact, of the 5 interpretations obtained in Figure 1.10, only 3 of them are actually globally consistent, as shown in Figure 1.11. We will consider how to use the geometric information implicit in the pairings of data and model features in an interpretation to deduce global consistency and to find the actual pose of an object in later chapters.

The point of this example is to illustrate the role of geometric constraints in reducing the search for feasible interpretations of the data. The goal of this work is to study the effects illustrated in this simple

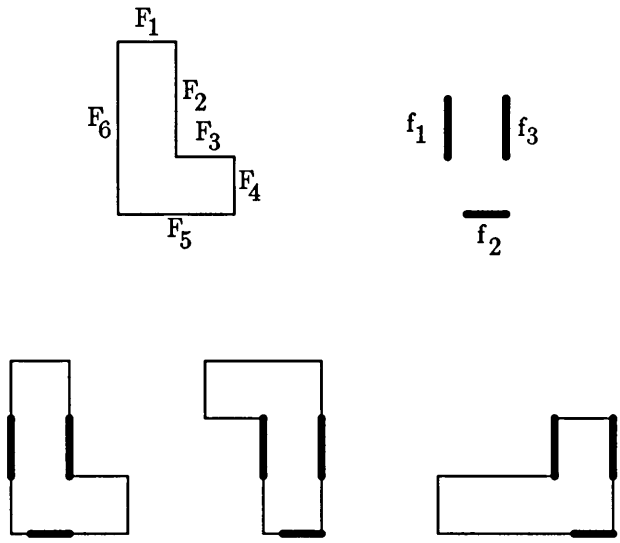


Figure 1.11
If the model on the top left is matched to the data on the top right, there are several feasible interpretations, as shown in the bottom set of figures.

example in detail. In particular, we will examine a variety of possible geometric constraints, and their effectiveness in isolating correct instances of objects and their poses.

It is important to note that one can separate the issue of how the tree of interpretations is search (e.g. depth first, breadth first, best first, beam search, etc.) from the role that the tree plays in defining consistent correspondences to consider. We will return in later chapters to methods of searching the tree. For the purposes of our example, we simply concentrated on the role of the tree in providing a method for exploring all possible solutions, while at the same time allowing geometric constraints to reduce wasted work.

1.3 What constitutes a good solution?

To conclude our introduction to the problem, we need to step back from the details. Our goal is to find feasible interpretations of data relative to a library of object models. We have suggested a basic approach to the problem in our simple example above, in which we tried to match data edges to model edges by searching through consistent interpreta-

tions in an interpretation tree. We want to explore a range of possible approaches to the problem of recognition and localization, however, using geometric constraints as our key tool. Hence, to focus discussion both of the method described in Section 1.2 and of alternatives to that method, we need a means of evaluating these different approaches. In our discussion, we will require that a solution to the recognition problem satisfy the following desired properties:

- **Efficiency:** A good recognition method must be as efficient as possible. This should be measured both in terms of run time efficiency and in terms of formal complexity.
- **Correctness:** Obviously, the method should find correct interpretations of the data. This means that it should not find false positive identifications (i.e. no cases of incorrectly claiming to have found an interpretation), and very few false negative identifications (i.e. few cases of missing a correct interpretation).
- **Robustness:** A recognition method must degrade gracefully with increasing noise in the sensory measurements, with a decrease in the amount of available relevant data, and with an increase in the amount of irrelevant data.
- **Scope:** The set of circumstances under which the method will meet the previous criteria should be as broad as possible. In particular, we argue that a realistic recognition method must work even when presented with partially occluded objects, with spurious data due to cluttered environments, and with relatively sparse information. As well, while particular optimizations will be possible for specific sensors, one would like a recognition framework that serves as a common core for recognition from many types of sensors, including tactile, range, and visual sensors.

Throughout the discussion that follows, we will indicate the evaluation of different approaches to recognition and localization based on these criteria. As well, we will show how these criteria can influence the development of different approaches.

1.4 Why is this a hard problem?

The criteria described above appear somewhat obvious, especially when considering the ease with which humans recognize objects in cluttered

environments. Why is this a hard problem for a computer, and what are the key difficulties to overcome?

When given perfect sensor data about an isolated object, there are many techniques that can identify an object and its pose. In realistic situations, however, there are three additional aspects of the problem that require careful consideration:

- occlusion,
- noise,
- and spurious data.

In most unstructured environments, recognition must proceed even when only some portion of the object's surface is visible, and when much of the data available from the scene does not come from the object of interest, i.e. one must be able to deal with both occlusion and spurious data. Thus, a good recognition system must both identify what data arises from the object of interest, and must use that data to determine the pose of the object. This implies that there are three separate parts to the process of interpreting sensor data:

- determining what object is present;
- determining the subset of the data to be matched with the object model;
- and determining the actual transformation that maps the model to this data subset, i.e. the pose of the object.

Thus, the example of Section 1.2 is simplistic in that all of the sensory data are assumed to have come from the object of interest, although the example does not assume that all of the object is visible in the data.

The recognition process is further complicated by the fact that in most realistic situations, the sensory data is corrupted by significant amounts of noise. The recognition process must show graceful degradation in the presence of bounded amounts of error. As a consequence, many methods that work well when confronted with perfect data from isolated objects do not extend well to real situations, and in evaluating approaches to recognition it is important to consider how well they deal with these factors. Again, the example of Section 1.2 is overly simplistic in that the sensory data are assumed to be perfect, so that we can compare angles between data edges and model edges exactly. In general, we must extend our methods to ensure that noise in the data measurements does not preclude our ability to recognize and locate an object.

Thus, in exploring the use of geometric constraints in object recognition, we focus on several factors:

- how can geometric constraints on relative shapes of object parts be used to control the search for consistent interpretations?
- how well do methods based on geometric constraints perform in the presence of noise, occlusion and spurious data?
- can we identify those situations to which such methods are well suited, and those situations in which poorer performance implies the need for additional or alternative techniques?

1.5 A view of things to come

Our goal in this book is to explore aspects of the problem of recognition and localization using geometric features. As we will see, there are many components of this problem in which there are several different means of solution, each trading off different advantages and disadvantages. Part of the goal of the book is to explore those tradeoffs, and to provide a framework in which to understand the impact of those choices. In exploring this area, we will consider a number of questions, including:

- how does establishing a correspondence between data and model features solve the recognition and localization problems?
- what are the choices of features to use?
- how can they reliably be extracted from raw sensory data?
- how does the relative geometry of features constrain the possible interpretations of them as instances of an object?
- how do we find correspondences of data and object features, and can we find them efficiently?
- how do we ensure that our solutions are correct?

We stress that recognition is an immense problem, and in the space of this book we cannot possibly deal adequately with all of it. Hence, this book should be read not so much as a manual for building an intelligent machine that can recognize objects with the same versatility as humans, but rather as an exploration of an important and useful subclass of that broad problem. In particular, we will focus on the use of local features and the constraints that the geometry of those features bring to bear on the recognition and localization of rigid objects. We feel that this is an important component of object recognition, and systems that solve

this problem can be of considerable utility despite the fact that they only solve part of the general recognition problem. At the same time, however, it is clearly only one component. There are many other aspects of recognition that are only touched on in this book.

In exploring the role of geometry in recognition and localization, the discussion will focus mainly on research performed by the author over the past seven years, partly in collaboration with Tomás Lozano-Pérez and partly in collaboration with Daniel Huttenlocher, as well as research performed by other members of the author's research group, especially David Braunegg, Todd Cass, David Clemens, Gil Ettinger, and David Jacobs. We are not alone in approaching recognition through geometric constraints, however, and throughout the book we try to indicate other alternatives that fit within the same framework and their relationships to that framework. Indeed, several strong research programs in geometrically constrained recognition have evolved contemporaneously with our own, and we wish to acknowledge the strong influence the work of Bob Bolles and collaborators at SRI, and the work of Olivier Faugeras and collaborators at INRIA. Although each of these efforts germinated independently, we especially feel that these efforts were the first to lay out a clear framework for recognition from geometric constraints, and much of the growth of this area follows from that seminal work. While our discussion in this book will focus on our own particular variation of this approach, interested readers are urged to explore the other variations discussed in the text.

1.5.1 A roadmap

We begin our exploration of the role of geometric constraints in Chapter 2, where we expand on our simple example of this chapter, by considering alternative methods of searching for instances of an object in the data. Having explored alternatives and their tradeoffs, in Chapter 3 we concentrate on the constrained search approach, as exemplified by the example of Section 1.2. This chapter lays out a general framework for constrained search, and in Chapters 4 and 5, we provided details of different types of geometric constraints that can be plugged into that framework. Chapter 6 considers the problem of actually finding the pose of an object associated with an interpretation, and verifying that the interpretation is globally consistent.

One of the main problems in constrained search approaches to recognition is controlling the inherent combinatorial explosion associated with the search. Variations on the search method used to find interpretations, and their effect on the combinatorics, are considered in Chapter 7. In Chapter 8 we consider other methods for controlling this explosion, mainly by restricting the portions of the search space to be explored. Empirical data summarizing the effects of these choices are given in Chapter 9. The first 9 chapters constitute the first part of the book, and provide a detailed exposition of a variety of search methods for utilizing geometric constraints in recognition.

The utility and practicality of the methods developed in the first part of the book have been demonstrated on a variety of real data, as summarized in Chapter 9. To explore the generality of these results, however, we also need a formal way of examining these methods, and we do this in the second part of the book. In particular, Chapters 10–13 develop a formal model of the recognition method, and derive analytic results on the complexity of constrained search approaches to recognition. These results carry some implications concerning the relative difficulty of different parts of the recognition problem, and these are discussed in Chapter 14.

The final part of the book deals with various extensions of the basic methods developed in the first part, and analyzed in the second part. Chapter 15 deals briefly with the problem of recognition from libraries of objects, Chapter 16 discusses extensions from rigid objects to broader classes of objects, Chapter 17 discusses briefly the role of grouping in recognition, and Chapter 18 explores the idea of sensing strategies. Finally, Chapter 19 briefly describes some representative applications of these recognition methods.