Model-Based Object Recognition
A Survey of Recent Research

Arthur R. Pope

Technical Report 94–04

January 1994

Abstract

We survey the main ideas behind recent research in model-based object recognition. The survey covers representations for models and images and the methods used to match them. Perceptual organization, the use of invariants, indexing schemes, and match verification are also reviewed. We conclude that there is still much room for improvement in the scope, robustness, and efficiency of object recognition methods. We identify what we believe are the ways improvements will be achieved.

Contents

1. Introduction

The object recognition problem is essentially this: given some knowledge of how certain objects may appear, plus an image of a scene possibly containing those objects, report which objects are present in the scene and where. This general definition admits many specific variations, not all of which will be considered here.

- Approaches differ according to what sort of knowledge they employ. We will be concerned with *model-based* object recognition, in which the knowledge of an object's appearance is provided by an explicit model of its shape. We won't be considering some other types of knowledge that may be used to recognize an object, such as knowledge of the context in which it may be found (Strat and Fischler 1991) or the function for which it may serve (Stark and Bowyer 1991).
- Approaches differ according to what restrictions they place on the form of the objects recognized. The objects may be two-dimensional—like symbols printed on traffic signs—or fully three-dimensional. Their shape may be restricted to a simple form—such as the class of polyhedra—or allowed to be more complex. They may be entirely rigid, composed of a few parts that can move rigidly with respect to each other ("articulate"), or capable of deforming throughout a whole range of shapes.
- Approaches differ according to the kind of image they are designed for. Some find objects in an intensity image, and others, in a range image or a combination of range and intensity images. Recognizing a 3D object in an intensity image of an unrestricted scene is, in many respects, the most difficult form of the problem. It requires dealing with the loss of depth information due to projection, with the possibility that objects may occlude each other, and with the fact that image intensity is only indirectly related to object shape.

Recognition is accomplished by finding a correspondence between certain features of the image and comparable features of the model. The two most important issues that a method must address are what constitutes a feature, and how is the correspondence found between image features and model features. Some methods use *global* features, which summarize information about the entire visible portion of an object. Examples of such features are area, perimeter length, compactness, and Euler number. Global feature methods are popular for those applications where lighting can be precisely controlled and occlusion does not occur, but because they presuppose perfect segmentation of objects from their background and from each other, these methods do not serve well in more general situations. We will focus, instead, on methods that use *local* features, such as edge segments and junctions.

In discussing object recognition methods, we ought to have in mind some criteria for judging how well a method performs. Most researchers have been concerned with the following criteria, as summarized by Grimson (1990):

scope	What kinds of objects can be recognized, and in what kinds of scenes?
robustness	Does the method tolerate reasonable amounts of noise and occlusion in the scene, and does it degrade gracefully as those tolerances are exceeded?
efficiency	Recognition requires that an enormous space of alternatives be considered. How much time and memory are required to search that space?
correctness	We cannot define correctness in any absolute sense because the object recognition problem is too loosely defined. Nevertheless, each method is based on a particular definition of the problem that engenders some criteria for ranking the alternate interpretations of a scene and deciding which of those interpretations to report. We can ask, then, does the method correctly implement the intended ranking and decision criteria, or, instead, does it sometimes miss an interpretation that it should have preferred.

The remainder of this survey is in three parts. Chapter 2 addresses the representation of object models and images for the purpose of recognition. Chapter 3 addresses methods of finding suitable matches between object models and images. In chapter 4 we conclude that there is still much room for improvement in the scope, robustness, and efficiency of object recognition methods, and we identify what we believe are the ways improvements will be achieved.

There have been other surveys published on object recognition. Extensive ones by Besl and Jain (1985) and by Chin and Dyer (1986) are now largely out-of-date, and omit research areas of recent importance such as geometric invariants and alignment matching. A survey by Brady, Nandhakumar, and Aggarwal (1989) is particularly about recognition in range images, and a recent one by Suetens, Fua, and Hanson (1992) covers a wide variety of matching techniques; however, both surveys also omit discussion of some of the most active areas of recent research, including invariants, grouping, and indexing. In contrast, one aim of this survey is to highlight areas where research is currently most active and promising.

2. Representation

Two representation schemes are needed for model-based object recognition: one to represent an object's model, and the other, an image's content. To facilitate finding a match between model and image the two representations should be closely related. In the ideal case there will be a simple relation between primitives used to describe the model and those used to describe the image. If the object is described by a wireframe model, for example, then the image might best be described in terms of intensity edges, each of which can be matched directly to one of the model's "wires". Yet, as in this example, the model and image representations often have distinctly different meanings—the model may describe the actual shape of an object while the image describes only visible manifestations of that shape.

Most model-based object recognition approaches have described objects only in terms of their shape, without detailing additional properties such as colour and texture. (An exception is Mahmood's system (1993), which employs colour and texture as well as shape.) Similarly, images have usually been described in terms of the visible manifestations of object shape—by the shape of intensity edges, for example, or the shape of range surfaces. Consequently, this survey only considers techniques for representing shape. Ways of possibly extending these techniques to other properties, however, are often readily apparent.

We will follow Marr and Nishihara(1978) in their use of the terms *shape*, *representation*, and *description*. *Shape* will mean the geometry of a locus of points, which will typically be the points on the surface of a physical object, or the points on an intensity edge or region in an image. A shape *representation* is a language for describing shape or some aspects of shape. It includes a set of shape *descriptions*, and a mapping between shape descriptions and shapes.

2.1 What makes a good shape representation?

Many researchers have prefaced their proposals for shape representations with a discussion of the criteria that ought to be satisfied by such a representation. These may be found, for example, in Marr and Nishihara 1978; Binford 1982; Brady 1983; Woodham 1987; Haralick, Mackworth, and Tanimoto 1988; and Mokhtarian and Mackworth 1992. The following have been mentioned frequently in some manner:

scope and sensitivity The representation must be able to describe all relevant shapes while preserving all important distinctions among those shapes.

uniqueness For any particular shape there should be a unique description. If this criterion is met, identical shapes will have identical descriptions and the problem of comparing shapes is simplified.

stability Small changes in shape should produce small changes in description. By ensuring that similar shapes will have similar descriptions, this also simplifies the problem of comparing shapes.

efficiency It must be possible to efficiently compute a shape description from input data, which, in the context of object recognition, may be either an image or an object model. Also, it must be possible to compare shape descriptions efficiently.

These criteria lead to several conclusions about the nature of a good shape representation. First, shape should be described by a combination of primitives with each primitive describing a limited portion of the overall shape. Such *local* primitives can be computed relatively efficiently since each is based on a limited portion of the input data. Moreover, a description composed of local primitives is relatively stable since only some of the primitives will be affected by any small change in shape. And, of particular importance for object recognition, a description composed of local primitives is only partly affected when its shape becomes partly occluded in an image. Thus the efficiency and stability criteria both argue for the use of shape primitives that are local.

A further consequence of the stability criterion is that the representation should describe shape over a range of scales while somewhat decoupling the descriptions at different scales. Two shapes that are similar on a large scale should have similar descriptions, even if the shapes differ in small-scale details. A *multi-scale* representation accomplishes this by composing a description from primitives having a range of different scales.

For the sake of convenient and efficient matching of shape descriptions, each primitive should bear a type or name denoting some limited category of possible shapes. So, for example, surface regions might be classified as being planar, cylindrical, elliptical, or hyperbolic. Later we will survey some commonly used types of shape primitives. The point to be made here is that the range of possible shapes, which is practically a continuum, must be divided into discrete categories by the representation.

Altogether, then, a representation should partition shape into discrete primitives according to three qualities: location, scale, and the category of shape at a particular location and scale. Unfortunately, however, this discretization leads to a conflict between the stability and uniqueness criteria. Instabilities are found at the thresholds between discrete categories. With small changes in shape, a surface that is sometimes classified as planar, for example, might at other times be classified as cylindrical. One way to reduce this source of instability is to allow some overlap and redundancy among the discrete primitives. Then a surface that lies near the threshold between planar and cylindrical would be represented by both types of primitives at once, and as the surface deforms to become more planar or more cylindrical, one of the two primitives will remain to provide some measure of stability. Note, however, that although it may provide greater stability, a redundant representation does not meet the uniqueness criterion, for various subsets of a redundant description may constitute alternate descriptions of the same shape.

Fortunately, the uniqueness property is not of great importance in the context of object recognition. Recall that the purpose of the uniqueness criterion is to permit straightforward tests for shape equivalence. In object recognition, however, we usually compare shapes to judge not whether they are identical, but whether they are sufficiently similar, with some allowance for mismatch due to noise and occlusion. In comparing shapes, then, the mismatch that may result from having alternate descriptions for a single shape can often be accommodated with little additional effort.

Almost any function of shape could be considered a shape primitive, and, indeed, many alternatives have been suggested. How do we decide, then, what makes a good primitive? As Saund has pointed out (1988), the qualities that various researchers have sought are essentially twofold. First, the primitives should make explicit whatever information is required for the task at hand. For a particular object recognition task, that means the primitives must somehow capture all distinctions needed to differentiate the objects. Second, the primitives should reflect the regularities and structures of the external world. For example, they should exploit properties that are invariant with respect to changing conditions. A common example is the intensity edge visible at a surface tangent discontinuity, which remains detectable across a range of illumination conditions; primitives based on intensity edges can exploit this illumination invariance. Both of these criteria lead to the conclusion that primitives ought to depend closely on the nature of the application: what task is being performed, and in what environment. Saund's thesis, that designing appropriate primitives provides a way to embed important knowledge about an application, is based on this understanding.

2.2 The choice of coordinate system

To describe the relative locations of various shape primitives, a shape representation must employ some coordinate system. There are principally two ways to define this coordinate system for a representation of three-dimensional shape.

2.2.1 Object-centred representation

One choice produces what is called an *object-centred* representation. A single coordinate system is affixed to the object and used to locate the various shape primitives. Or, if the object is represented by a hierarchical arrangement of parts, each part carries its own coordinate system and it is located using the coordinate system of its parent part. An object-centred representation yields the most concise shape descriptions, and usually the most accurate. It also gives a convenient way of describing objects composed of rigid parts that can move with respect to each other (as in Brooks 1981 and Lowe 1989). However, when an object-centred representation is used to describe models for object recognition, either of two difficulties must be faced.

- One can attempt to derive a similar, object-centred description from the image and match that description with various models. It is difficult, though, to segment objects or parts in an image and fix coordinate systems to them reliably. A strategy of fixing coordinate systems to the centres of objects, for example, will be foiled whenever those objects are partly occluded.
- Alternatively, one can derive a 2D description from the image, and use a matching procedure that accounts for the projection of a 3D object onto a 2D image. Because it must consider the effects of self-occlusion and perspective, this 3D/2D matching procedure faces a more difficult task than a 2D/2D or 3D/3D procedure.

Despite these difficulties, however, both approaches have been demonstrated experimentally. For example, Dickinson and Pentland (1992) recover 3D volumetric primitives before recognizing objects in terms of those primitives, while Lowe (1987a) matches 3D models directly to 2D image features.

2.2.2 Viewer-centred representation

The alternative to an object-centred representation is a *viewer-centred* or *multi-view* one, which describes the 3D object using a set of 2D *characteristic views* or *aspects*. Each characteristic view describes how the object appears from a single viewpoint, or from a range of viewpoints yielding similar views. Matching is simpler than with an object-centred representation because it involves comparing descriptions that are both two-dimensional; there is no need to project the model during matching, and the continuous space of view-points has been reduced to a discrete choice among characteristic views.

For even a moderately complex object, however, there are many qualitatively distinct views to be recorded. Thus this representation requires considerably more space than an object-centred one. Space requirements can be reduced somewhat by allowing views to share common structures (Minsky 1975; Burns and Riseman 1992) and by merging similar views after discarding features too fine to be reliably discerned (Petitjean, Ponce, and Kriegman 1992).

Another consequence is that there are, in effect, many more models to be considered during recognition since each characteristic view is a separate model. However, this may be more than compensated for by the fact that testing each model requires only a 2D/2D match rather than a 3D/2D or 3D/3D one, and 2D/2D matches can be performed much more quickly (Breuel 1992b). Systems that recognize 3D objects in 2D images using viewer-centred models have been described by Breuel (1992b), Burns and Riseman (1992), and Camps, Shapiro, and Haralick (1991).

There is some interesting evidence that the human visual system uses a viewer-centred representation for object recognition (Ullman 1989; Edelman and Bülthoff 1992). Humans are able to recognize objects more accurately and rapidly when they are seen from particular viewpoints, implying that those views of an object are readily available while others must be computed as needed.

A viewer-centred description usually provides only an approximation of object shape. Each characteristic view must represent an entire range of viewpoints over which the appearance of the object may vary somewhat. As more views are used, each view can cover a smaller range and do so more accurately. Hence there is a trade-off between the size of a description and its accuracy. Breuel (1992b) has quantified this trade-off by determining the number of views needed to represent a model of point features to within any desired precision.

Fortunately, when a shape is viewed from a range of different viewpoints some relations among its features appear to remain nearly constant. This sort of invariance can extend the range of viewpoints covered by a single characteristic view, thus improving the trade-off between accuracy and number of views. Certain relations between lines in three dimensions, such as cotermination (the proximity of endpoints), parallelism, and collinearity, appear approximately invariant when seen from various viewpoints (Lowe 1985). Even angles between general pairs of lines (not necessarily parallel ones), and ratios of line lengths, may be relatively stable with respect to viewpoint (Ben-Arie 1990; Burns, Weiss, and Riseman 1993).

Another technique for improving the space/accuracy trade-off of a viewer-centred representation is to interpolate among views. Ullman and Basri (1991) have shown that with three views of a rigid object whose contours are defined by surface tangent discontinuities, one can interpolate among the three views with a linear operation to produce a fourth view. If the object has smooth contours instead then the appearance of its rim is somewhat harder to predict; nevertheless, six views yield a good interpolation.

An advantage of a viewer-centred description is that it can be built relatively easily by accumulating images of the object. (To build an object-centred description from images, on the other hand, one must solve the difficult problem of reconstructing 3D structure from 2D images.) Moreover, a viewer-centred description can also be built from an object-centred one, either empirically or analytically. To build it empirically, a *view sphere* of viewing positions about the object is sampled uniformly or stochastically. From each viewpoint the object is rendered under orthographic projection, and similar views are clustered to obtain the characteristic views (e.g., Pathak and Camps 1993; Sato, Ikeuchi, and Kanade 1992; Zhang, Sullivan, and Baker 1993). To build a viewer-centred model analytically, the view sphere is subdivided into regions by identifying the boundaries where the object's self-occlusions begin and end, and one characteristic view is chosen from each region (e.g., Eggert and Bowyer 1993; Petitjean, Ponce, and Kriegman 1992). Algorithms exist for performing such analysis on a class of shapes that has been recently extended to include solids of revolution (Eggert and Bowyer 1993) and algebraic surfaces (Petitjean, Ponce, and Kriegman 1992).

2.2.3 Combining object- and viewer-centred representations

Some recent research has sought ways to combine the best features of both object- and viewer-centred representations. Dickinson, Pentland, and Rosenfeld (1992) first recognize generic parts by their characteristic views, and then assemble those parts into an object-centred description for comparison with object models. With this approach, a small number of characteristic views can support recognition of many different objects provided each object can be expressed as some composition of the generic parts. Zhang, Sullivan, and Baker (1993) use both object- and viewer-centred descriptions of each object for recognition. The viewer-centred descriptions are first used to find a quick, approximate match; that match is then verified using the object-centred description. Since the views are only used to suggest possible matches, this approach requires relatively few characteristic views and each view only has to contain a few prominent primitives.

2.3 Viewpoint invariants

Because recognition calls for identifying an object under varying conditions of pose and lighting, it is helpful to have primitives that are at least somewhat invariant with respect to changes in these conditions. This is especially true when recognizing 3D objects from 2D intensity images where the effects of lighting and pose are confounded.

Recently, considerable effort has been directed at identifying and employing primitives that are completely invariant with respect to viewpoint (Mundy and Zisserman 1992; Weiss 1993). These are based on properties of geometric structures, called *invariants* or *geometric invariants*, that remain constant over an entire class of transformations. Each is defined in terms of a particular kind of geometric structure and a particular class of transformations.

One example of a geometric invariant is the *cross ratio*. For any four distinct, collinear points A, B, C, and D, the value $(AC \cdot BD)/(AD \cdot BC)$, where AC denotes the distance from A to C, etc., is unchanged by a perspective transformation, or, for that matter, by any composition of perspective transformations (i.e., a projective transformation). Of most interest for object recognition, of course, are invariants for the Euclidean, affine, and projective transformations that are often used to model the imaging process.

The cross ratio, like certain other geometric invariants, is computed from the coordinates of a small group of primitives. Choosing that group is itself a challenging problem; it will be discussed in section 3.1. Interesting invariants can also be computed from the coefficients of planar algebraic curves, from a set of higher-order derivatives taken at one point on a curve, or from certain combinations of feature coordinates, algebraic curve coefficients, and derivatives. However, whereas the use of coordinates entails the grouping problem, the use of algebraic curve coefficients or derivatives entails the problem of accurately estimating these quantities. Some current research is directed at finding invariants, along with associated grouping and estimating methods, that optimize the tradeoff between having to choose groups and having to estimate coefficients or derivatives (e.g., Weiss 1993).

It has been shown that there is no invariant for 2D projections of a finite, unconstrained set of 3D points (Clemens and Jacobs 1991a; Burns, Weiss, and Riseman 1993; Moses and Ullman 1991). Consequently attention has focused on invariants for suitably constrained structures, particularly coplanar sets of points, lines, and curves. Four examples will illustrate the nature of this research.

- Three coplanar, non-collinear points define two vectors, which can be used as a basis for representing the locations of other points lying in the same plane. The resulting representation is affine invariant, and hence also invariant to weak perspective transformations (Lamdan, Schwartz, and Wolfson 1988).
- A pair of coplanar conics yields two projective invariants. These can be used to identify objects like gaskets, which are essentially coplanar and typically composed of circles and ellipses (Forsyth et al. 1991).
- A rotationally symmetric object has a silhouette that is essentially planar. Therefore invariants of coplanar features can be used to identify a rotationally symmetric object by its silhouette (Forsyth et al. 1992).
- There is an invariant of orthographic projections of four 3D points that applies only if the four points define three orthogonal vectors. A test can be used to first determine which sets of points meet that condition so that the invariant is only relied upon where appropriate (Wayner 1991).

Another approach is to use properties that, although not truly invariant, are nearly so. Ben-Arie (1990) and Burns, Weiss, and Riseman (1993) have developed probabilistic models that describe how certain image properties, such as junction angles and ratios of distances, vary with viewpoint. They have found that these properties are sufficiently stable over a wide enough range of viewpoints to be useful for recognition. Practically speaking, even true invariants must be considered only approximately invariant since the image measurements from which they are computed can only be obtained with limited precision.

2.4 Common shape primitives

We will now briefly survey the shape primitives that are commonly used in object recognition work. It is helpful to think of them as being of three general types:

a) Segments and patches

A 2D curve, such as an intensity edge, can be approximated by a series of straight line segments, and a surface can be approximated by a series of polygonal faces. Although these primitives are convenient, they produce unstable descriptions when used to describe curved shapes. To allow stable descriptions to be produced for a larger class of shapes, including some curves, the set of primitives can be extended by adding higher-order analytic curves or surfaces. By adding elliptical arcs, for example, one can have reasonably stable descriptions of the projections of a circle. Segments of parabolas, circles, ellipses, and general conics have been used to describe 2D curves, and spherical, cylindrical, and spline patches have been used to describe 3D surfaces. However, because they involve more parameters, higher-order curves are more difficult to extract reliably from sensor data. Furthermore, while they may extend the class of shapes that can be represented well with few primitives, higher-order curves will still fail to yield stable descriptions of some shapes. Many systems have been demonstrated that use segments or patches as shape primitives, including HYPER, which uses polygonal approximations of 2D curves (Ayache and Faugeras 1986), and 3D-POLY, which uses quadric approximations of 3D surfaces (Chen and Kak 1989).

b) Parts

With a limited set of parts one can construct a large variety of objects, especially if each part can be customized somewhat by choosing values for free parameters. Generalized cylinders, generalized cones, and superquadrics are three parameterized families of parts used to describe volumes.

A *generalized cylinder* or cone is a volume swept out by running a closed planar figure along a space curve. The figure, the space curve, and the relation between them are all restricted so that just a handful of parameters are needed to determine the part's shape. Two-dimensional analogs of the generalized cylinders are *ribbons*, *symmetric axis transforms*, and *smoothed local symmetries*. Generalized cylinders and their two-dimensional counterparts describe only certain elongated shapes well. Systems that have used them include ACRONYM (Brooks 1981), PARVO (Bergevin and Levine 1993), and a system by Dickinson, Pentland, and Rosenfeld (1992).

A *superquadric* is an equation of a particular form whose solutions define a closed surface. As the equation's two parameters are varied, the surface deforms through a range of shapes that includes cubes, diamonds, pyramids, and smooth, intermediate forms. Six more parameters are added to specify size along each axis, bending along two axes, and tapering along one axis, producing a family of parts more expressive than generalized cones and cylinders. Because so many parameters are involved, however, it has proven difficult to recover these superquadrics reliably, even from range images.

c) *Distinguished features*

A third approach is to describe shape only in terms of a limited set of distinguished features. Often these are chosen with a particular application in mind so that, although they may not describe a shape completely, they will capture the information that is important. The *curvature primal sketch* (CPS) is one example of this approach (Asada and Brady 1986). It represents 2D curves only in terms of the significant changes of curvature—including corners and inflections—that are detected at each scale. Among the motivations for the CPS is a hypothesis due to Attneave (1954) that, for the human visual system, a contour's extrema of curvature are more significant for recognition than are its other features. A second motivation for the CPS is that curvature changes provide a relatively stable description, as they remain detectable over a wide range of affine transformations.

In some cases, features have been designed to represent a very restricted class of objects. To demonstrate how knowledge can be brought to bear in the design of a shape representation, Saund (1992) has designed a representation for fish fin profiles using a vocabulary of approximately thirty features. Each feature captures just one fragmentary aspect of fin shape, such as the angle or curvature of a fin's leading edge. Collectively, though, the features provide a rich representation that is capable of distinguishing a wide variety of fin shapes. Face recognition systems provide another example of the use of distinguished features. In building her system, for example, Gordon (1992) has chosen to represent faces only in terms of the geometry of selected eye, nose, and cheek features.

One way in which distinguished features can be chosen for a particular application is to learn them from training images depicting the set of objects to be recognized. Segen (1989) has described a system that constructs a hierarchy of shape primitives by grouping point features in training images, measuring the geometry of each group, clustering these measurements to identify classes of common groups, and associating a primitive with each class. This produces primitives corresponding to configurations of point features that are particularly prevalent among the objects to be recognized. Others have used similar unsupervised learning techniques to automatically develop features that are 2D patterns of intensity or its derivative (e.g., Turk and Pentland 1991; Murase and Nayar 1993; Weng, Ahuja, and Huang 1993).

Thus far we have considered what types of shape primitives a representation might employ. Also important is the issue of what sizes or scales those primitives should have. Any shape feature much smaller than the smallest primitives simply cannot be represented, and any feature much larger than the largest primitives cannot be represented explicitly. Therefore primitives, regardless of their type, must be available in a range of scales if they are to make explicit all important features of shape.

2.5 Organizing shape primitives

There have been two common methods for organizing shape primitives into some sort of structure. The first is to arrange them hierarchically according to part/whole relations. In some cases, levels of the hierarchy represent degrees of descriptive accuracy, so that a single primitive at one level decomposes into several finer primitives at the next (Marr and Nishihara 1978; Brooks 1981). More commonly, levels of the hierarchy represent degrees of grouping and primitives occur only at the hierarchy's lowest level (Ettinger 1987; Connell and Brady 1987). A second method of organizing shape primitives, not necessarily incompatible with the first, is to arrange them according to adjacency relations so that each primitive is related to others nearby (Shapiro 1980; Wong, Lu, and Rioux 1989). Both part/whole and adjacency organizations are often represented as a graph (or hypergraph) in which nodes denote primitives or groups of primitives, and arcs denote the relations among them.

An advantage for these organizations is that they provide a convenient way to organize additional information about the relative geometry of primitives. Arcs can be annotated with attributes that record the position of a part with respect to a whole, or the position of a primitive with respect to its neighbour. For an object composed of rigid parts connected by articulating joints, it is especially convenient to represent geometry this way since the configuration of each joint need be recorded in only one place. Moreover, a part/whole decomposition supports an object recognition strategy, described in section 3.3.4, whereby object parts are first recognized, and then whole objects are recognized as configurations of those parts.

2.6 Representing models

The discussion has dealt thus far with representing the shape of a single, static object whose description has consisted of a collection of shape primitives plus information about the geometry of those primitives. When modeling an object for recognition, however, we may wish to describe an object whose shape can vary within certain limits, or a class of objects whose shapes may differ in certain ways. One approach is to use a parameterized model in which free variables or quantifiers are used to specify certain measurements (e.g., Brooks 1981; Lowe 1989; Grimson 1990). A model for pencils, for example, might describe the pencil body with a generalized cylinder whose length is a free variable; one for scissors might use a free variable to describe the rotation of each rigid blade with respect to the other. The model may include constraints limiting each parameter to a set of reasonable values (e.g., "pencil length is less than 10 cm"), and, in the case of the ACRONYM system (Brooks 1983), these constraints have even specified relations among parameters (e.g., "wing width is no more than half of wing length").

Parameterization of the model can be taken a step further. By using probability distributions rather than hard constraints to characterize parameters, one can represent uncertainty about an object's shape or, equivalently, a distribution of possible object shapes. For this purpose, Wong and You (1985) introduced a structure they called a *random graph*: an attributed graph in which a distribution is given for each attribute. McArthur (1991) has used this structure to represent a model of a 3D, rigid object in terms of point features whose locations are characterized by Gaussian distributions. Camps, Shapiro, and Haralick (1992) have used a similar structure to represent 2D characteristic views, also with Gaussian distributions.

Besides describing shape and perhaps shape's expected variation, an object model might also include information about the robustness of various object features and the expected cost of detecting them in images. A feature will be less useful for recognizing the object if it is not always part of the object, or if the feature has a poor chance of being detected. When recognizing the object, then, a good strategy is to first seek the features that are most robust and least costly. Feature robustness can be measured from actual images of the object, or it can be estimated by analyzing a model of the object. In performing this analysis some systems have simply estimated a feature's detectability as the portion of the viewsphere over which it is visible (e.g., Goad 1983; Kuno, Okamoto, and Okada 1991), while others have argued that much more complete models of lighting, sensors, and surfaces are needed to obtain useful estimates (e.g., Camps, Shapiro, and Haralick 1991; Chen and Mulgaonkar 1992; Sato, Ikeuchi, and Kanade 1992).

3. Recognition

This chapter considers the task of recognizing an object by finding a match between a model and an image. The inputs to this task are a library of object models and an image in which objects are to be recognized; the outputs specify the identity, pose, and perhaps certainty of any objects recognized in the image. This task is made difficult by several factors, including:

- not knowing which of many possible objects might be present in the image
- not knowing what pose an object might assume in the image
- the possibility that an object might be partly occluded in the image
- the possibility that the image may be cluttered with unknown objects as well as artifacts of the imaging process
- limits to the reliability and accuracy of sensor measurements

Following convention, we will use the term *feature* in this chapter to refer to a characteristic of appearance or form. What in this chapter is called a *feature* is essentially the same thing as what in the previous chapter was called a *shape primitive*. An object is recognized by its features, and an object model specifies those features.

To recognize a single object in an image, most methods perform something like the following sequence of steps:

feature detection	signal processing to detect features in the image and represent them as symbols
perceptual organization	identify stable groupings of features
indexing	use these features to select a likely model out of a library of object models
matching	find the best match between features of the image and those of the selected model
verification	decide whether that match suggests that the modeled object is present in the image

To recognize any number of objects in an image, most methods recognize single objects repeatedly until no further objects are found. After each object is recognized, its image features may be deleted and the next recognition cycle performed using whatever features remain.

The indexing and matching steps both involve choosing among alternatives—alternative features, models, or matches. Thus the entire process is essentially a large search through several stages of choices. One way that methods differ is in how they balance these stages in an attempt to minimize the overall workload. Some methods emphasize effective indexing to minimize the number of alternatives that must be considered by later stages, while other methods emphasize fast matching or verification so that indexing need not be so selective.

Feature detection, which includes such processes as image segmentation and edge detection, is not peculiar to object recognition; the topic is too extensive and too general to be surveyed here. The remaining four steps, from perceptual organization to verification, are the subjects of this chapter's four sections.

3.1 Perceptual organization

Perceptual organization is a process of grouping image features. Its purpose in object recognition is to produce group of features that are more informative than individual features, and therefore better able to guide the selection and matching of models.

The term perceptual organization has been widely applied to activities ranging from segmenting curves to recognizing geometric figures, but what qualifies all these processes as forms of perceptual organization is their generality. They are based not on knowledge of specific objects, but rather on general assumptions that hold for most objects and situations encountered. Segmentation may be based on the assumptions that most surfaces are smooth and most objects convex, for example. Insofar as the underlying assumptions are valid, the feature organization produced will be descriptive and stable; where those assumptions fail, the organization may be spurious or unreliable. Because it uses no specific knowledge, perceptual organization is predominantly a bottom-up process that proceeds iteratively from low-level groupings to high-level ones. Features at each level are selected, grouped, and/or abstracted to form those at the next level.

One important issue in this area is how to decide what features to group in an image. A decision about whether to group certain features is typically made by measuring various aspects of their relative geometry and by comparing those measurements to thresholds; ideally, the context of the features is also considered. So, for example, if two line segments are sufficiently close to each other, if they are approximately parallel, and if no prominent segments lie between them then a decision may be made to group the two segments, producing a new feature denoting a pair of parallel lines.

To be useful for object recognition, a group must not span objects; instead, its features must be derived entirely from a single object. While grouping methods try to produce groups that satisfy this condition, it cannot be guaranteed that all groups will. (Ensuring that each group involves only one object is impractical, for it first requires a perfect segmentation of the image.) The emphasis, therefore, is on producing groups that are *likely* to involve a single object, yet *unlikely* to arise otherwise, due to an accidental arrangement of objects. Subsequent object recognition stages must allow for missing and spurious groups by only assuming that some unknown (possibly empty) subset of the groups is actually correct.

Lowe (1985, 1990) first explicitly addressed the role of perceptual organization in object recognition. He suggested the following rationale (somewhat simplified here) for guiding grouping decisions. Because a viewpoint invariant structure projects the same pattern over a wide range of viewpoints, that pattern is more likely to occur in images than other patterns. Moreover, an occurrence of the pattern is more likely due to the presence of the 3D structure than to some accidental alignment or objects. Thus any arrangement of features that matches the pattern closely enough ought to be grouped. How close must the match be? Given a set of features, a viewpoint invariant pattern, and some assumptions about how features are distributed due to chance, one can estimate the probability that the features would match the pattern at least as well as they do due to chance alone. This is essentially the probability of the feature arrangement arising by accident. If that probability is below some threshold, then the features are grouped.

Except for work by Jacobs (1989) on grouping pairs of convex contours, there seems to have been no other effort like this to found grouping decisions upon first principles. Instead, it has been common practice to develop ad hoc criteria for deciding which sets of features should be grouped (e.g., Bergevin and Levine 1992; Horaud, Veillon, and Skordas 1990; Mohan and Nevatia 1992; Sarkar and Boyer 1990; Saund 1990; Stein and Medioni 1992a). Researchers concerned with the problem of how to identify groups efficiently have produced data structures for quickly locating related features (Sarkar and Boyer 1990; Saund 1990; Stein and Medioni 1992a), parallelizable algorithms (Mahoney 1987; McCafferty 1990), and other process improvements (Huttenlocher and Wayner 1992). Sarkar and Boyer (1993) have represented the grouping process using a Bayes network in order to allow a control strategy more flexible than the usual bottom-up process. With their approach a grouping hypothesis may lead to a top-down search for additional evidence in the image, and hypotheses may compete with one another to resolve choices among alternate groupings.

3.2 Indexing the model database

Given a library of object models and some features found in an image, we want to select a model that is likely to match those features. In this, the indexing problem, the goal is to do much better than simply trying each model in turn. Solutions generally involve a table that is indexed either by individual features or by small groups of them. Each table entry indicates a model (and perhaps viewpoint) that could produce the corresponding feature or group. Before recognition, the table entries are created for various viewpoints of each model by analyzing the model library, by rendering each model from a sample of viewpoints, or by processing a representative set of training images. During recognition, features chosen from the image are used to index the table, thus producing hypotheses about what objects are present in the image. Each hypothesis denotes a possible, partial match between model and image, which must be further tested to determine the full extent and quality of the match.

Many variations of this basic scheme are possible, and several have been investigated. We might index the table using all features from the image (Breuel 1990), a randomly chosen subset of features (Lamdan, Schwartz, and Wolfson 1988), or just features that are judged particularly likely to be derived from single objects (Clemens and Jacobs 1991b). We might test every hypothesis retrieved from the table for a possible match (Clemens and Jacobs 1991b), or instead treat the hypotheses as votes, and only test those that receive the most votes (Lamdan, Schwartz, and Wolfson 1988) or some minimum number of votes (Stein and Medioni 1992b). Votes may all bear equal weight, or they may be weighted according to the size of the lookup feature or how well the lookup feature matches each table entry (Beis and Lowe 1993; Rigoutsos and Hummel 1993; Sarachik and Grimson 1993).

If not all features are used to index the table, or if voting is used and only those hypotheses receiving the most votes are tested, then we must have some way of deciding when we have examined enough features or tested enough hypotheses. One approach is to estimate the probability of missing a model with each test and repeat testing until the cumulative probability drops below any desired error threshold (Lamdan, Schwartz, and Wolfson 1988).

How well indexing performs depends on how well entries are distributed throughout the table. The entries associated with any one object should occupy a relatively small portion of the entire table, and each entry should refer to relatively few objects. Poor distributions occur when objects are symmetrical or when they share common features (Flynn 1992).

Indexing performance is also greatly affected by uncertainty in feature measurements. Uncertainty may be accommodated by coarsely quantizing feature dimensions, by replicating table entries (with a range of duplicate entries representing a range of possible measurements), or by sampling a range of index values at lookup time. All three approaches reduce the selectivity of the index, and replication greatly increases its storage requirement as well. Jacobs (1992) has shown that, for indexing based on groups of point features and a particular type of projection, the table's storage requirement can be reduced by factoring the table into two tables, each having half as many dimensions.

A markedly different approach to indexing has been developed using networks of neuron-like units that compute functions called *generalized radial basis functions* or *hyper-basis functions* (Poggio and Edelman 1990; Brunelli and Poggio 1991; Edelman and Poggio 1990). A modeled object is represented by a network in which individual units represent distinct prototypical or characteristic views of the object. The input to the network is a vector of selected image feature measurements. The output represents either a normalized view of the object (which must be further tested against some standard view), or a graded yes/no recognition response. A library of object models is represented by a collection of networks, one for each model; for recognition, all networks are applied in parallel and the object's identity is indicated by the network producing the strongest response.

Although these networks have been described as performing object recognition, we consider them to be essentially an indexing scheme performing just one component of the complete object recognition task. Like other indexing schemes, these networks consider only a small, fixed-size subset of image features, and they must be applied to all such subsets to generate all hypotheses. They alone do not resolve the question of which image features are associated with an object. As indexing schemes, however, these networks do not compare well with other indexing schemes because, at one network per object, their time complexity scales linearly with the size of the model library. And although the networks have the advantage that they can be trained using images, there appear to be other trainable classifiers that can perform more quickly and about as accurately (e.g., a nearest–neighbor classifier, as used by Brunelli and Poggio (1992)).

Another recently proposed indexing approach organizes the model library hierarchically. Models of similar objects are clustered, and each cluster is represented in the library by a single prototype. This clustering may be repeated several times to form a model abstraction hierarchy. Recognition proceeds by descending this hierarchy while refining an object's identity along the way. However, a full search for a match between image and model features need only be performed at the hierarchy's top level—at each lower level, the match result from the level above provides an advanced starting point in the search for a more complete match. Variations of this approach have been described by Basri (1993) and by Sengupta and Boyer (1993).

3.3 Matching features

Given an image and an object model, both represented in terms of their features, we want to find a partial match between the two and estimate how the modeled object is positioned in the image. A match solution must satisfy the *viewpoint consistency constraint* (Lowe 1987b), which requires that the locations of the object's features in the image be consistent with some pose of the object.

Of all the consistent solutions to a matching problem, we usually want one that maximizes some appropriate measure of match quality. Such measures are often based on error models that describe how an image feature may differ from what the object model has predicted. Two common error models are:

- a *bounded error model*, which requires each matching image feature to be within some fixed range of its predicted location. The corresponding match quality measure is often just a count of the number of matching features.
- a *Gaussian error model*, which specifies that image features are distributed normally and independently about their predicted locations. The corresponding match quality measure usually considers both the number of matching features, and the sum of the squares of their normalized errors.

Match quality measures are now often defined using Bayesian probability theory (e.g., Wells 1992). Given such factors as the prior probability that each object is present in the image, the prior distribution of its pose if present, the conditional probability that each model feature is matched if its object is present, and the conditional distribution of matching errors described by an error model, one can estimate the posterior probability that a particular object is present with a given pose and a given set of feature matches. This posterior probability then serves as a match quality measure. Assumptions of feature independence are needed to keep the approach tractable.

One can classify various matching methods according to whether they search for a solution in *correspondence space*, *transformation space*, or both. Correspondence space is the space of *matches*, which are sets of pairings between model and image features. Transformation space is the space of possible object poses, viewpoints, or transformations between object and camera. Under the viewpoint consistency constraint and an appropriate error model, the two spaces are closely related—each match is consistent with a (possibly empty) set of transformations, and each transformation, with a (possibly empty) set of matches.

3.3.1 Correspondence space search

The interpretation tree approach (Grimson 1990) exemplifies those methods that search entirely in correspondence space. Its name refers to a search tree of choices concerning the interpretation of each image feature. Proceeding from the root of the tree, the match search examines an additional image feature at each level of the tree. Branches at each level represent different choices among model features that can be matched to that image feature, plus the choice of matching nothing at all to it. A complete interpretation of the image, assigning some subset of image features to corresponding model features, is associated with each of the tree's leaves.

The search can readily incorporate two kinds of matching constraints: *unary constraints*, which require that matching image and model features have similar properties, and *binary constraints*, which require that pairs of feature matches be geometrically consistent. Although these local constraints alone are not sufficient to ensure viewpoint consistency and consequently each complete interpretation must be subsequently verified, the local constraints do effectively prune the search. According to Grimson (1990), higher-order constraints provide no real improvement because of their higher computational cost.

Besides constraints, a branch-and-bound technique can also be used to prune the search tree. This technique requires a function that evaluates a score for a complete interpretation, plus an estimator that bounds the scores of any complete interpretations that might follow from a given partial interpretation. The search tree is pruned wherever the estimator shows no improvement possible over the best score yet obtained for a complete interpretation.

Correspondence space search has often been cast as a problem of graph matching (e.g., Ben-Arie and Meiri 1987; Bergevin and Levine 1993; Fan 1990; Shapiro and Haralick 1981; Wong 1992; Yang, Snyder, and Bilbro 1989; Zhang, Sullivan, and Baker 1992). In this framework, the task is to find a common subgraph isomorphism between two attributed graphs: one representing the image and the other, the model. Graph nodes represent features, graph edges or hyperedges represent the geometrical relations among them, and nodes and edges have attributes recording their properties or measurements. Usually an *inexact match* is sought, where the attributes of matching nodes and edges are allowed to differ somewhat to accommodate noise and distortion in the image. The exponential search for an optimal graph match is guided by some measure of graph match quality, which must evaluate both how well the two graph's structures match and how well their corresponding attribute values match. This measure serves the same purpose as the match quality measure discussed above, and it too may be based on Bayesian probability theory.

The biggest difficulty with correspondence space methods has been their computational cost, which is generally exponential in the number of image or model features. One way to avoid considerable computation is to relax the requirement that an optimal match be found and search instead for a near-optimal one. This is the tactic employed by *heuristic search termination* (Grimson 1990), which terminates the search as soon as a solution meeting some minimum requirement has been found. Relaxation labeling has also been used as a way to shorten the search while accepting a sub-optimal result (e.g., Bhanu and Faugeras 1984; Bray 1990; Kitchen 1980).

3.3.2 Transformation space search

The generalized Hough transform is an example of a method that searches transformation space. An array of bins, indexed by parameters of object pose, is first initialized as empty. Then, for each possible match between one image feature and one model feature, poses consistent with that match are determined and votes are cast in the bins corresponding to those poses. Finally, when votes have been placed on behalf of all matches, the array is scanned to identify and verify those poses that have received the most votes. Usually a bounded error model is used so that a finite range of poses and a corresponding finite range of bins are consistent with each matching; for reasonable error bounds, however, these ranges can be quite large.

The principle advantage of this and other transformation space methods is that, unlike correspondence space methods, they avoid exponential search. Unfortunately, however, not every bin collecting a large number of votes represents a correct match solution. In some cases, due to an accident of how the array tessellates transformation space, a bin may collect many votes that are not all consistent with a single pose. Also, when the image contains a great deal of clutter, random clusters of votes may overshadow a correct solution so that a large portion of the array must be examined before a correct solution is found (Grimson and Huttenlocher 1990).

Cass (1992) and Breuel (1992a) have developed transformation space algorithms that avoid the problems that are due to tessellation. Like the generalized Hough transform, they use a bounded error model to associate a range of poses with each possible match between one image feature and one model feature. By designing their features, error model, and transformation space carefully, they are able to ensure that these pose ranges are of a particularly simple form. For example, if the features are points, if error bounds are convex polygons, and if transformation space consists of 2D translations, rotations, and changes of scale, then the region of transformation space associated with each feature match is simply a convex polytope.

Each region can therefore be expressed as the intersection of several half-spaces, which in turn are defined by hyperplanes. To identify maximal sets of consistent feature matches, they search transformation space to find areas where maximal numbers of regions intersect. Because the regions are expressed in a convenient form (using hyperplanes), and because the number of regions is only polynomial in the numbers of image and model features, this search can be performed relatively efficiently.

Wells (1992) has shown how the transformation space search can be cast as an iterative estimation problem, solvable by an algorithm that is analogous to Newton's method. Using Bayesian theory and a Gaussian error model, he defines the posterior probability of a particular match and pose given some input image. This probability is then integrated over all possible matches, producing a marginal probability of pose that, for a given input image, is function only of pose. Limited experiments suggest that this function is relatively smooth, and that its maximum is usually near the correct pose. With an initial guess provided by indexing, an iterative procedure called *expectation-maximization* locates this maximum efficiently.

3.3.3 Using both search spaces

Instead of searching solely in either correspondence space or transformation space, some methods do some portion of their search in each space. Ullman's *alignment method* is one of these (Ullman 1989). It begins the search in correspondence space where it matches just enough *anchor features* to determine a viewpoint transformation. This requires three point features if the object is rigid, fewer if the point features have associated orientations (as junctions do, for example), and more if the object to be recognized is somewhat flexible. Once the viewpoint transformation is determined, it is used to project the remaining features of the model into the image. There additional matches are sought for each projected feature. Because there may be many combinations of anchor feature matches, this method relies heavily on having efficient techniques for computing and verifying transformations (Huttenlocher 1988).

The alignment method estimates a viewpoint transformation once for each set of anchor feature matches, and an error in localizing an anchor feature in the image yields an error in the transformation estimate. That error perturbs the locations of projected features, contributing to errors in matching those additional features. Whereas the alignment method requires that at least one set of anchor features produces a sufficiently accurate estimate of the viewpoint transformation, other methods attempt to overcome the inaccuracy of the initial estimate. Using additional matches involving projected features, they refine that estimate iteratively.

Lowe (1987a, 1987b) has described this iterative alignment approach and demonstrated it with his SCERPO system. As with the alignment method, a viewpoint transformation is first estimated from a small set of feature matches. This transformation is used to predict the visibility and image location of each remaining model feature. For each of these projected model features, potential matches with nearby image features are identified and ranked. This ranking considers both the likelihood that an image feature could occur by accident so close to its projected model feature, and the degree to which the match is rendered ambiguous by the presence of another, nearby image feature. The best ranked matches are then adopted, all matches are used to produce a refined estimate of the viewpoint transformation, and the process is repeated until acceptable matches have been found for as many of the model features as possible. Although back-tracking can be used to try alternate matches for projected features, Lowe reports that this is seldom necessary because the ranking scheme is usually successful in eliminating ambiguous matches, and because errors in the initial estimate of the viewpoint transformation are eliminated as additional matches are incorporated. A similar method of iteratively refining a viewpoint transformation estimate has been used by Ayache and Faugeras (1986) in their HYPER system.[1]

Another way of employing both correspondence and transformation spaces is first to identify interesting regions of transformation space by means of something like a coarse Hough transform, and then, within each region, to perform a correspondence search while considering only matches consistent with that range of transformations. One such method is described by Kuno, Okamoto, and Okada (1991). Their system begins with a Hough transform that accumulates evidence for various possible poses of the object. As various matches are hypothesized with each contributing evidence for certain poses, the system assesses the uncertainty of the growing accumulation of evidence. When that uncertainty drops below some threshold (i.e., a limited range of poses begins to appear particularly likely) the system begins searching instead for model features within restricted image regions predicted by the pose distribution. The threshold at which this transition takes place is determined according to the costs of the two search methods so that the overall search cost is minimized.

---

[1] The work of Huttenlocher, Lowe, Ayache, and Faugeras cited here has also contributed methods of estimating a viewpoint transformation from a set of feature matches. Viewpoint estimation, however, is not covered in this survey.

3.3.4 Ordering and structuring the search

The efficiency of a correspondence space search is significantly affected by the order in which features are considered for matching. At each stage of the search one would like to choose an unmatched model feature that is likely to be found in the image when the object is present, is relatively unique among features of the model, is not likely to be encountered when the object is not present, can effectively constrain other matches, and, according to what is already known about the object's pose, can be expected to lie within a small region of the image.

Goad (1983) suggested how one might use criteria like these to choose the next model feature for each stage of an interpretation tree search. To measure the criteria he proposed sampling a uniform distribution of viewpoints, estimating the object's appearance from each viewpoint using a 3D model, and tallying each feature's visibility and position. The tallies would be used to rank each feature according to how likely it is to be visible, how accurately its position can be predicted, and how much information about the object's pose would be provided by matching it. This analysis would be used to pre-determine the entire search tree for a given model before any recognition attempts. Goad's experiments demonstrated the idea of pre-ordering the search tree but they did not test his ranking criteria—features were ordered subjectively, by hand. Kuno, Okamoto, and Okada (1991) have described a similar approach that also considers the likelihood of features arising accidentally, the cost of detecting them, and the degree to which they may be distorted by perspective. Another variation, the *local feature focus method* (Bolles and Cain 1982), involves analyzing the entire database of object models to select for each model one or more focus features that, due to their specificity, can be used to initiate matching by alignment.

There has been some investigation of how to organize the match search hierarchically. In his SAPHIRE system, Ettinger (1987) uses both a structure hierarchy for dividing object models into subparts, and a scale hierarchy for representing each subpart at various levels of detail. Before recognition, models are decomposed automatically into subparts according to heuristics that look for "necks" and other shape properties. SAPHIRE's recognition algorithm first recognizes the subparts in an image and then recognizes objects in terms of those subparts (i.e., treating entire subparts as features). When recognizing a subpart, it first uses coarse features and then proceeds to finer ones. By decomposing the search in these ways, a large search is replaced by numerous smaller searches, altogether requiring less computation. The idea of decomposing the search according to a structure hierarchy and using that hierarchy to guide a bottom-up process is also found in systems by Burns and Riseman (1992) and by Dickinson, Pentland, and Rosenfeld (1992).

3.4 Verifying the match result

The last step in the recognition process is to decide whether an optimal match found by the match search actually represents an instance of an object in the image. There has been relatively little research on how best to make this decision. Many systems simply require that some fraction of the model's features be matched, and/or that some fraction of the edges projected from the model lie near image edges (e.g., Chen and Kak 1989; Gottschalk, Turney, and Mudge 1989; Lamdan, Schwartz, and Wolfson 1990). Some also assess negative evidence, such as image edges that cross projected model edges at large angles (e.g., Hansen and Henderson 1989; Huttenlocher and Ullman 1990). Match solutions are verified by testing these measures against empirically determined thresholds, and then ranked according to the measures to select the best, mutually-consistent solutions.

Grimson, Huttenlocher, and Sarachik have developed analytic models for estimating the probability that a particular match may be accidental—i.e., due to a conspiracy of random features rather than the actual presence of the object (Grimson and Huttenlocher 1991; Sarachik and Grimson 1993). Only if this probability is below some threshold is the match accepted. Chen and Mulgaonkar (1992) have used a similar analysis not only to accept or reject matches, but also to halt the match search as soon as a sufficiently reliable decision can be made.

Recently, Breuel (1993) has suggested that verification consider not only how much of a model is matched, but also the spatial distribution of its unmatched parts. In a valid match, the features that remain unmatched do so primarily because of occluding objects that typically cover contiguous regions of the image. Verification, therefore, should test how well these unmatched features can be explained away by hypothesizing a small number of contiguous occlusions.

4. Conclusion

Early in this survey we listed the criteria commonly used to evaluate object recognition methods: scope, robustness, efficiency, and correctness. We should now ask, how well have existing methods met these requirements, and what factors have contributed to the difficulties and successes in these areas?

Scope

Most methods are applicable only to highly restricted classes of objects, and none can cope with the full range of objects found in natural environments. Usually objects are required to have smooth contours and be rigid or articulate so that they can be readily matched against a simple, fixed model of shape. These restrictions may not be a problem in special situations, as in factories, but they effectively rule out recognition of most objects in most environments.

One factor that often seems to underlie this scope restriction is a reliance on some small, fixed set of shape primitives for describing all objects. Invariably these primitives embody strong assumptions about objects, and consequently they are only suitable for representing objects that fit those assumptions. For example, one scheme represents an object as composed of a small number of generic parts, such as generalized cylinders or superquadrics. The generic parts are kept simple so that they can be segmented and identified in images prior to recognition; as a consequence, however, the parts are of limited representational power and they are unable to portray many natural objects (the scheme also omits any representation of object markings). Another scheme is to recognize objects using geometric invariants; then objects must also be severely restricted because there are no suitable invariants for unrestricted configurations of points.

It appears that a solution to the scope problem will not come from finding some small, universal set of features capable of representing any object. We should expect, instead, that a rather broad range of features will be needed. Many of these features might apply only to certain objects but, collectively, they would be able to generate rich descriptions of a large class of objects. Furthermore, since it would be impractical for any designer to try to anticipate all features that could prove useful for recognition, it will be important for a general purpose recognition system to be able to coin new features as needed. So far there have been only a few efforts to construct recognition systems capable of learning their own representations; we might hope to see much more effort in this area.

Robustness

Since most methods have been tested only by their proponents, and apparently with just small, hand-picked sets of objects and images, it is difficult to assess how robust these methods truly are. Trials comparing alternate methods under similar conditions are extremely rare. We can speculate, however, as to which among the current crop of ideas seem most important for achieving robustness. They appear to be the following:

- *Effective use of redundant information.* An image of an object usually contains a considerable amount of redundant information. In some ways, this redundancy can compensate for missing or inaccurate features. For example, the indexing methods reviewed in section 3.2 use this redundancy to identify correct interpretations even when a large proportion of features are missing or inaccurate. Another use of redundancy is in representations that span a range of scales; large-scale features make explicit certain information that is already implicitly available in small-scale ones, but the large-scale features allow objects to be easily recognized as similar even when they differ in some details.
- *Use of probabilistic models*. A simple approach assumes that an object can be modeled as a single, idealized form, and that any departure of its actual form from this ideal is simply "noise". In practice, however, individual objects of the same type often differ in particular ways; some features are not shared by all individuals and feature geometry may vary. Pencils, for example, differ greatly in length and in the conditions of their two ends, but in other ways they are quite similar. Probabilistic models like those described in section 2.6 try to represent these kinds of variations more accurately, and their use should permit more reliable recognition of classes of similar objects.
- *Use of probabilistic reasoning*. Matching methods are now often formulated in terms of some similarity measure and, increasingly, that measure is based on probability theory. Probability theory provides a framework for integrating a probabilistic model, the observations of many noisy features, and other sources of knowledge. The framework contributes to robustness by allowing appropriate importance to be accorded each of many information sources.

This survey has not examined how feature detection is accomplished. Nevertheless, that is also an area where we should be seeking improvements to enhance the robustness of object detection. Of course, without robust and stable feature detection no recognition method can succeed, regardless of what else it may have going for it.

Efficiency and correctness

Efficiency and correctness are considered together in this discussion because one of the most effective ways of improving efficiency in object recognition has been to permit some relaxation of the assurance of correctness. That tactic has been responsible, in part, for the performance gains of indexing methods like geometric hashing, and of matching methods like alignment. There is much more work to be done exploring ways that recognition can be made fast by permitting answers to be approximate or occasionally suboptimal.

Efficiency remains an important problem despite continuing improvements in computational power and algorithms. Recognition systems that work with large databases of complex objects still appear to be beyond the reach of the practicable. However, some interesting ideas have been proposed for improving efficiency and, although these have been tried out individually, the task now appears to be to find effective ways of integrating them. For matching, different strategies such as coarse-to-fine, part-to-whole, and abstract-to-specific have been described. Each might be best in certain situations, but the best overall performance may come from integrating them or choosing among them as needed.

An object recognition system should fine-tune itself according to the contents of its model library and the images it is seeing. If an index table is used, for example, then the table's quantization level could be adjusted according to the number of entries it contains and the degree of accuracy found in image measurements. Grouping thresholds could be adjusted according to the prevalence of those groups among objects and the degree of clutter found in images. Such parameters are now established statically in most systems, but they, and the performance of the system, could be improved through automated learning.

References

Asada, H. and Brady, M. 1986. "The curvature primal sketch." *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-8(1): 2–14.

Attneave, F. 1954. "Some informational aspects of visual perception." *Psych. Rev.* 61: 183–193.

Ayache, N. and Faugeras, O.D. 1986. "HYPER: A new approach for the recognition and positioning of two-dimensional objects." *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-8(1): 44–54.

Basri, R. 1993. "Recognition by prototypes." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 161–167.

Beis, J.S. and Lowe, D.G. 1993. "Learning indexing functions for 3-D mode-based object recognition." In *Proc. AAAI Fall Symposium: Machine Learning in Computer Vision*, pp. 50–54.

Ben-Arie, J. 1990. "The probabilistic peaking effect of viewed angles and distances with application to 3-D object recognition." *IEEE Trans. Patt. Anal. Machine Intell.* 12(8): 760–774.

Ben-Arie, J. and Meiri, A.Z. 1987. "3D objects recognition by optimal matching search of multinary relations graphs." *Comput. Vis. Graphics Image Processing* 37: 345–361.

Bergevin, R. and Levine, M.D. 1992. "Extraction of line drawing features for object recognition." *Patt. Recogn.* 25(3): 319–334.

———. 1993. "Generic object recognition: Building and matching coarse descriptions from line drawings." *IEEE Trans. Patt. Anal. Machine Intell.* 15(1): 19–36.

Besl, P.J. and Jain, R.C. 1985. "Three-dimensional object recognition." *Computing Surveys* 17(1): 75–154.

Bhanu, B. and Faugeras, O.D. 1984. "Shape matching of two-dimensional objects." *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-6(2): 137–156.

Binford, T.O. 1982. "Survey of model-based image analysis systems." *Int. J. Robotics Res.* 1(1): 18–64.

Bolles, R.C. and Cain, R.A. 1982. "Recognizing and locating partially visible objects: The local-feature-focus method." *Int. J. Robotics Res.* 1(3): 57–82.

Brady, J.P., Nandhakumar, N. and Aggarwal, J.K. 1989. "Recent progress in object recognition from range data." *Image Vision Comput.* 7(4): 295–307.

Brady, M. 1983. "Criteria for representations of shape." In *Human and Machine Vision*, ed. by J. Beck, B. Hope and A. Rosenfeld, Academic Press, pp. 39–84.

Bray, A.J. 1990. "Object recognition using local geometric constraints: A robust alternative to tree-search." In *Proc. European Conf. Comput. Vis.*, pp. 499–515.

Breuel, T.M. 1990. *Indexing for Visual Recognition from a Large Model Base*. A.I. Memo 1108, A.I. Lab., Mass. Inst. Technol.

———. 1992a. "Fast recognition using adaptive subdivisions of transformation space." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 445–451.

———. 1992b. “Geometric Aspects of Visual Object Recognition.” Ph.D. dissertation, Dept. of Brain and Cognitive Sciences, Mass. Inst. Technol.

———. 1993. “Higher-order statistics in object recognition.” In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 707–708.

Brooks, R.A. 1981. “Symbolic reasoning among 3-D models and 2-D images.” *Artificial Intell.* 17: 285–348.

———. 1983. “Model-based three-dimensional interpretations of two-dimensional images.” *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-5(2): 140–150.

Brunelli, R. and Poggio, T. 1991. “HyperBF networks for real object recognition.” In *Proc. Int. Joint Conf. on Artificial Intell.*, vol. 2, pp. 1278–1284.

———. 1992. “Face recognition through geometrical features.” In *Proc. European Conf. Comput. Vis.*, pp. 792–800.

Burns, J.B. and Riseman, E.M. 1992. “Matching complex images to multiple 3D objects using view description networks.” In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 328–334.

Burns, J.B., Weiss, R.S. and Riseman, E.M. 1993. “View variation of point-set and line-segment features.” *IEEE Trans. Patt. Anal. Machine Intell.* 15(1): 51–68.

Camps, O.I., Shapiro, L.G. and Haralick, R.M. 1991. “PREMIO: An overview.” In *Proc. Workshop on Directions in Automated CAD-Based Vision*, Maui, Hawaii: IEEE Computer Society Press, pp. 11–21.

———. 1992. “Object recognition using prediction and probabilistic matching.” In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Raleigh, North Carolina: pp. 1044–1052.

Cass, T.A. 1992. “Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty.” In *Proc. DARPA Image Understanding Workshop*, pp. 693–704.

Chen, C.-H. and Mulgaonkar, P.G. 1992. “Automatic vision programming.” *CVGIP: Image Understanding* 55(2): 170–183.

Chen, C.H. and Kak, A.C. 1989. “A robot vision system for recognizing 3-D objects in low-order polynomial time.” *IEEE Trans. Syst. Man Cybernetics* 19(6): 1535–1563.

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

Clemens, D. and Jacobs, D. 1991a. “Space and time bounds on model indexing.” *IEEE Trans. Patt. Anal. Machine Intell.* 13(10): 1007–1018.

———. 1991b. “Model group indexing for recognition.” In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 4–9.

Connell, J.H. and Brady, M. 1987. “Generating and generalizing models of visual objects.” *Artificial Intell.* 31: 159–183.

Dickinson, S.J. and Pentland, A.P. 1992. "A unified approach to the recognition of expected and unexpected geon-based objects." In *Applications of Artificial Intell. X: Machine Vision and Robotics*, vol. 1708, SPIE, pp. 614–627.

Dickinson, S.J., Pentland, A.P. and Rosenfeld, A. 1992. "3-D shape recovery using distributed aspect matching." *IEEE Trans. Patt. Anal. Machine Intell.* 14(2): 174–198.

Edelman, S. and Bülthoff, H.H. 1992. "Orientation Dependence in the Recognition of Familiar and Novel Views of Three-Dimensional Objects." *Vision Res.* 32(12): 2285–2400.

Edelman, S. and Poggio, T. 1990. *Bringing the grandmother back into the picture: A memory-based view of object recognition*. A.I. Memo 1181, A.I. Lab, Mass. Inst. Technol.

Eggert, D. and Bowyer, K. 1993. "Computing the perspective projection aspect graph of solids of revolution." *IEEE Trans. Patt. Anal. Machine Intell.* 15(2): 109–128.

Ettinger, G.J. 1987. "Hierarchical Object Recognition Using Libraries of Parameterized Model Sub-parts." Masters dissertation, Dept. of Electrical Engineering and Computer Science, Mass. Inst. Technol.

Fan, T.J. 1990. *Describing and Recognizing 3-D Objects Using Surface Properties*. Springer-Verlag.

Flynn, P.J. 1992. "Saliencies and symmetries: Toward 3D object recognition from large model databases." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 322–327.

Forsyth, D.A., Mundy, J.L., Zisserman, A.P., Coelho, C., Heller, A. and Rothwell, C.A. 1991. "Invariant descriptors for 3-D object recognition and pose." *IEEE Trans. Patt. Anal. Machine Intell.* 13(10): 971–991.

Forsyth, D.A., Mundy, J.L., Zisserman, A.P. and Rothwell, C.A. 1992. "Recognising rotationally symmetric surfaces from their outlines." In *Proc. European Conf. Comput. Vision*, pp. 639–647.

Goad, C. 1983. "Special purpose automatic programming for 3D model-based vision." In *Proc. ARPA Image Understanding Workshop*.

Gordon, G.G. 1992. "Face recognition based on depth and curvature features." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 808–809.

Gottschalk, P.G., Turney, J.L. and Mudge, T.N. 1989. "Efficient recognition of partially visible objects using a logarithmic complexity matching technique." *Int. J. Robotics Res.* 8(6): 110–131.

Grimson, W.E.L. 1990. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press.

Grimson, W.E.L. and Huttenlocher, D.P. 1990. "On the sensitivity of the Hough transform for object recognition." *IEEE Trans. Patt. Anal. Machine Intell.* 12(3): 255–274.

———. 1991. "On the verification of hypothesized matches in model-based recognition." *IEEE Trans. Patt. Anal. Machine Intell.* 13(12): 1201–1213.

Hansen, C. and Henderson, T.C. 1989. "CAGD-based computer vision." *IEEE Trans. Patt. Anal. Machine Intell.* 11(11): 1181–1193.

Haralick, R.M., Mackworth, A.K. and Tanimoto, S.L. 1988. "Computer vision update." In *Handbook of Artificial Intelligence*, ed. by A. Barr, P.R. Cohen and E.A. Feigenbaum, Reading, Mass.: Addison-Wesley.

Horaud, R., Veillon, F. and Skordas, T. 1990. “Finding geometrical and relational structures in an image.” In *Proc. European Conf. Comput. Vis.*, pp. 374–384.

Huttenlocher, D.P. 1988. “Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image.” Ph.D. dissertation, Cambridge: MIT.

Huttenlocher, D.P. and Ullman, S. 1990. “Recognizing solid objects by alignment with an image.” *Int. J. Comput. Vision* 5(2): 195–212.

Huttenlocher, D.P. and Wayner, P. 1992. “Finding convex edge groupings in an image.” *Int. J. Comput. Vision* 8(1): 7–29.

Jacobs, D.W. 1989. *Grouping for Recognition*. A.I. Memo No. 1177, A.I. Lab., Mass. Inst. Technol.

———. 1992. “Space efficient 3D model indexing.” In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 439–444.

Kitchen, L. 1980. “Relaxation applied to matching quantitative relational structures.” *IEEE Trans. Syst. Man Cybernetics* 10(2): 96–101.

Kuno, Y., Okamoto, Y. and Okada, S. 1991. “Robot vision using a feature search strategy generated from a 3-D object model.” *IEEE Trans. Patt. Anal. Machine Intell.* 13(10): 1085–1097.

Lamdan, Y., Schwartz, J. and Wolfson, H. 1988. “Object recognition by affine invariant matching.” In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 335–344.

———. 1990. “Affine invariant model-based object recognition.” *IEEE Trans. on Robotics and Automation* 6(5): 578–589.

Lowe, D.G. 1985. *Perceptual Organization and Visual Recognition*. Boston: Kluwer.

———. 1987a. “Three-dimensional object recognition from single two-dimensional images.” *Artificial Intell.* 31: 355–395.

———. 1987b. “The viewpoint consistency constraint.” *Int. J. Comput. Vision* 1: 57–72.

———. 1989. *Fitting Parameterized 3-D Models to Images*. Tech. Rep. 89–26, Dept. of Computer Science, Univ. of British Columbia.

———. 1990. “Visual recognition as probabilistic inference from spatial relations.” In *AI and the Eye*, ed. by A. Blake and T. Troscianko, New York: John Wiley and Sons Ltd., pp. 261–279.

Mahmood, S.T.F. 1993. “Attentional Selection in Object Recognition.” Ph.D. dissertation, Cambridge: MIT.

Mahoney, J. 1987. *Image Chunking: Defining Spatial Building Blocks for Scene Analysis*. Tech. Rep. TR-980, A.I. Lab, Mass. Inst. Technol.

Marr, D. and Nishihara, H.K. 1978. “Representation and recognition of the spatial organization of three-dimensional shapes.” *Proc. Roy. Soc. London B* 200: 269–294.

McArthur, B.A. 1991. “Incremental Synthesis of Three-Dimensional Object Models using Random Graphs.” Ph.D. dissertation, Univ. of Waterloo.

McCafferty, J.D. 1990. *Human and Machine Vision: Computing Perceptual Organization*. Ellis Horwood Ltd.

Minsky, M. 1975. "A framework for representing knowledge." In *The psychology of computer vision*, ed. by P.H. Winston, pp. 211–277.

Mohan, R. and Nevatia, R. 1992. "Perceptual organization for scene segmentation and description." *IEEE Trans. Patt. Anal. Machine Intell.* 14(6): 616–635.

Mokhtarian, F. and Mackworth, A.K. 1992. "A theory of multiscale, curvature-based shape representation for planar curves." *IEEE Trans. Patt. Anal. Machine Intell.* 14(8): 789–805.

Moses, Y. and Ullman, S. 1991. *Limitations of Non Model-Based Recognition Schemes*. A.I. Memo 1301, A.I. Lab, Mass. Inst. Technol.

Mundy, J.L. and Zisserman, A. 1992. *Geometric Invariance in Computer Vision*. Cambridge: MIT Press.

Murase, H. and Nayar, S.K. 1993. "Learning and recognition of 3-D objects from brightness images." In *Proc. AAAI Fall Symposium: Machine Learning in Computer Vision*, pp. 25-29.

Pathak, A. and Camps, O.I. 1993. "Bayesian view class determination." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 407–412.

Petitjean, S., Ponce, J. and Kriegman, D.J. 1992. "Computing exact aspect graphs of curved objects: Algebraic surfaces." *Int. J. Comput. Vision* 9(3): 231–255.

Poggio, T. and Edelman, S. 1990. "A network that learns to recognize three-dimensional objects." *Nature* 343: 263–266.

Rigoutsos, I. and Hummel, R. 1993. "Distributed Bayesian object recognition." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 180–186.

Sarachik, K.B. and Grimson, W.E.L. 1993. "Gaussian error models for object recognition." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 400–406.

Sarkar, S. and Boyer, K.L. 1990. *An Efficient Computational Structure for Computing Perceptual Organization*. Tech. Rep. SAMPL–90–06, Ohio State University.

———. 1993. "Integration, inference, and management of spatial information using Bayesian networks: Perceptual organization." *IEEE Trans. Patt. Anal. Machine Intell.* 15(3): 256–274.

Sato, K., Ikeuchi, K. and Kanade, T. 1992. "Model based recognition of specular objects using sensor models." *CVGIP: Image Understanding* 55(2): 155–169.

Saund, E. 1988. "The Role of Knowledge in Visual Shape Representation." Ph.D. dissertation, Cambridge: MIT.

———. 1990. "Symbolic construction of a 2-D scale-space image." *IEEE Trans. Patt. Anal. Machine Intell.* 12(8): 817–830.

———. 1992. "Putting knowledge into a visual shape representation." *Artificial Intell.* 54: 71–119.

Segen, J. 1989. "Model learning and recognition of nonrigid objects." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 597–602.

Sengupta, K. and Boyer, K.L. 1993. "Information theoretic clustering of large structural modelbases." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 174–179.

Shapiro, L.G. 1980. "A structural model of shape." *IEEE Trans. Patt. Anal. Machine Intell.* 2: 111–126.

Shapiro, L.G. and Haralick, R.M. 1981. "Structural descriptions and inexact matching." *IEEE Trans. Patt. Anal. Machine Intell.* 3(5): 504–519.

Stark, L. and Bowyer, K. 1991. "Achieving generalized object recognition through reasoning about association of function to structure." *IEEE Trans. Patt. Anal. Machine Intell.* 13(10): 1097–1104.

Stein, F. and Medioni, G. 1992a. "Recognition of 3D objects from 2D groupings." In *Proc. DARPA Image Understanding Workshop*, pp. 667–674.

———. 1992b. "Structural indexing: Efficient 2-D object recognition." *IEEE Trans. Patt. Anal. Machine Intell.* 14(12): 1198–1204.

Strat, T.M. and Fischler, M.A. 1991. "Context-based vision: Recognizing objects using information from both 2-D and 3-D imagery." *IEEE Trans. Patt. Anal. Machine Intell.* 13(10): 1050–1065.

Suetens, P., Fua, P. and Hanson, A. 1992. "Computational strategies for object recognition." *Computing Surveys* 24(1): 5–61.

Turk, M.A. and Pentland, A.P. 1991. "Face recognition using eigenfaces." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 586-591.

Ullman, S. 1989. "Aligning pictorial descriptions: An approach to object recognition." *Cognition* 32: 193–254.

Ullman, S. and Basri, R. 1991. "Recognition by linear combination of models." *IEEE Trans. Patt. Anal. Machine Intell.* 13(10): 992–1006.

Wayner, P.C. 1991. "Efficiently using invariant theory for model-based matching." In *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pp. 473–478.

Weiss, I. 1993. "Geometric invariants and object recognition." *Int. J. Comput. Vision* 10(3): 207–231.

Wells, W.M. 1993. "Statistical Object Recognition." Ph.D. dissertation, Cambridge: MIT.

Weng, J.J., Ahuja, N. and Huang, T.S. 1993. "Learning recognition and segmentation of 3-d objects from 2-d images." In *Proc. Fourth Int. Conf. Comput. Vis.*, pp. 121–128.

Wong, A.K.C., Lu, S.W. and Rioux, M. 1989. "Recognition and shape synthesis of 3D objects based on attributed hypergraphs." *IEEE Trans. Patt. Anal. Machine Intell.* 11(3): 279–290.

Wong, A.K.C. and You, M. 1985. "Entropy and distance of random graphs with application to structural pattern recognition." *IEEE Trans. Patt. Anal. Machine Intell.* 7(5): 599–609.

Wong, E.K. 1992. "Model matching in robot vision by subgraph isomorphism." *Patt. Recogn.* 25(3): 287–304.

Woodham, R.J. 1987. "Stable representation of shape." In *Computational Processes in Human Vision*, ed. by Z. Pylyshyn, Norwood, N.J.: Ablex.

Yang, B., Snyder, W.E. and Bilbro, G.L. 1989. “Matching oversegmented 3D images to models using association graphs.” *Image Vision Comput.* 7(2): 135–143.

Zhang, S., Sullivan, G. and Baker, K. 1992. “Using automatically constructed view-independent relational model in 3D object recognition.” In *Proc. European Conf. Comput. Vis.*, pp. 778–786.

———. 1993. “The automatic construction of a view-independent relational model for 3-D object recognition.” *IEEE Trans. Patt. Anal. Machine Intell.* 15(6): 531–544.