# Computational Strategies for Object Recognition

PAUL SUETENS

*ESAT, Machine Intelligence and Imaging, K. U. Leuven, and Department of Radiology, University Hospital Leuven, Leuven, Belgium*

PASCAL FUA

*Artificial Intelligence Center, Computer and Information Sciences Division, SRI International, Menlo Park, California 94025 and INRIA Sophia-Antipolis, Valbonne, France*

ANDREW J. HANSON

*Artificial Intelligence Center, Computer and Information Sciences Division, SRI International, Menlo Park, California 94025 and Department of Computer Science, Indiana University, Bloomington, Indiana 47405*

This article reviews the available methods for automated identification of objects in digital images. The techniques are classified into groups according to the nature of the computational strategy used. Four classes are proposed: (1) the simplest strategies, which work on data appropriate for feature vector classification, (2) methods that match models to symbolic data structures for situations involving reliable data and complex models, (3) approaches that fit models to the photometry and are appropriate for noisy data and simple models, and (4) combinations of these strategies, which must be adopted in complex situations. Representative examples of various methods are summarized, and the classes of strategies are evaluated with respect to their appropriateness for particular applications.

Categories and Subject Descriptors: I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*architecture and control structures*; *modeling and recovery of physical attributes*; *representations, data structures, and transforms*; *shape*; I.4 8 [**Image Processing**]: Scene Analysis; I.5.4 [**Pattern Recognition**]: Applications—*computer vision*

General Terms: Algorithms, Design, Experimentation, Theory

Additional Key Words and Phrases: Image understanding, model-based vision, object recognition

## INTRODUCTION

This paper reviews practical computational strategies for recognizing objects in digital imagery. It provides both an introduction to the field of applied object recognition for the nonspecialist and a detailed guide to a representative body of literature and techniques for those beginning research in this domain. The focus is on mature techniques with explicit models and working applications.

The paper begins by defining the object recognition task and establishing the scope of the review. It then proposes a

CONTENTS



classification framework based on general characteristics of object recognition strategies. The central sections discuss the categories of the general classification, evaluate the applicability of the techniques, and present summaries of papers typifying each method. A list of related books and review articles, as well as more detailed analyses of selected papers, are presented in appendices.

## CHARACTERISTICS OF THE OBJECT RECOGNITION TASK

### Definitions

*Object recognition* is the task of finding and labeling parts of a two-dimensional (2D) image of a scene that correspond to objects in the scene. Figure 1 shows an example of the object recognition task as it might be carried out by a human observer with a marking pen; an aerial image of an industrial complex has been marked and labeled to show areas recognizable as buildings and roads.

*Photometry* usually refers to light intensities reflected from surfaces in a scene and recorded on camera film; on occasion, data originate from sources such as ultrasound or x-ray absorption instead of light. A digital computer image of a scene is a 2D array of numbers called *pixels* whose values represent the scene's photometry, that is, the strength of the signal arriving at a particular point on the recording medium. The image behind the markings in Figure 1 originated from an ordinary black and white photograph that was digitized into a 2D array of pixel values in computer memory. These numerical values contain all the photometric intensity information at our disposal.

To carry out the object recognition task, we must first establish *models*, or general descriptions of each object to be recognized. Typically, a model includes shape, texture, and context knowledge about the occurrence of such objects in a scene. For example, the mathematical description of a building model as a set of shaded rectangles might have been used to generate Figure 1. A three-dimensional (3D) building object could be modeled as a set of rectangular solids. Texture information might include colors or knowledge about the layout of a building's windows.

A *model label* is then attached to each occurrence (or *instance*) of a model in the

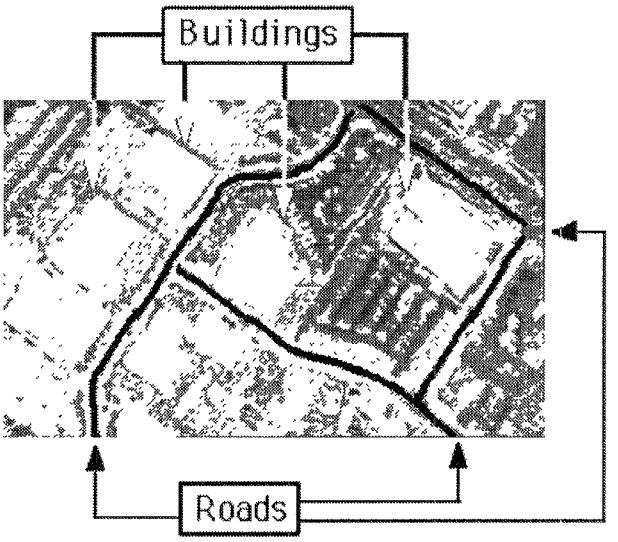



**Figure 1.** An aerial image of an industrial complex with labels attached to buildings and roads.

image. A model label can be thought of as a tag pinned to an area in the image that we believe shows an instance of the corresponding object model. The words *Roads* and *Buildings* in Figure 1 are examples of model labels; the outlined 2D areas indicate where we think the photograph shows 3D buildings. That is, a model may be two- or three-dimensional, whereas labels in an image always refer to 2D model instances. (We note that in certain cases the image and some of its labeled model instances may be three-dimensional.)

There are several important distinctions about the kind of information we deal with in a digital image and its corresponding scene. The most elementary type of information is *syntactic*, which deals only with the pixel values themselves, not their meaning. *Semantic* information, by contrast, deals with knowledge and meaning. Thus, when we talk about a syntactic image operator, we mean a procedure that blindly applies an algorithm to the pixel values; an example would be a procedure that assembles groups of adjacent pixels that have a high contrast with their other neighbors. A semantic operator, in contrast, uses models of the scene and the image production process that incorporate symbolic knowledge about the organization of the information, such as "parts may be lying on top of one another."

Closely related to the distinction between semantic and syntactic information are terms describing the spatial dependence of a procedure or concept. We frequently use the term *local* to refer to processes that look at a pixel and its very nearest neighbors but use no information about the rest of the image; local processes are typically syntactic. The term *global*, therefore, is used to refer to the opposite situation, in which context information from the entire image or scene, usually semantic in nature, is considered.

To sum up, we may think of object recognition as the process of drawing lines and outlining areas in an image and attaching to each such structure a label corresponding to the model that best represents it, as illustrated in Figure 1.

## Scope

In this paper we consider only object models that include knowledge of object structure such as shape or seman-

tic context. In particular, models that rely strictly on local photometry are not considered. Thus we do not treat in detail models defined only by syntactic statements such as "all contiguous pixels with intensity value above 128."

The effect of these restrictions is to focus our attention on object recognition strategies that are generally associated with the machine perception branch of artificial intelligence. Among the wide variety of object recognition problems that fall within the scope of this review are such tasks as locating single object instances, accurately determining object boundaries in an image, choosing an object's best class membership from among many possibilities, and extracting object labels from complex, cluttered scenes.

### Role of Context in Object Recognition

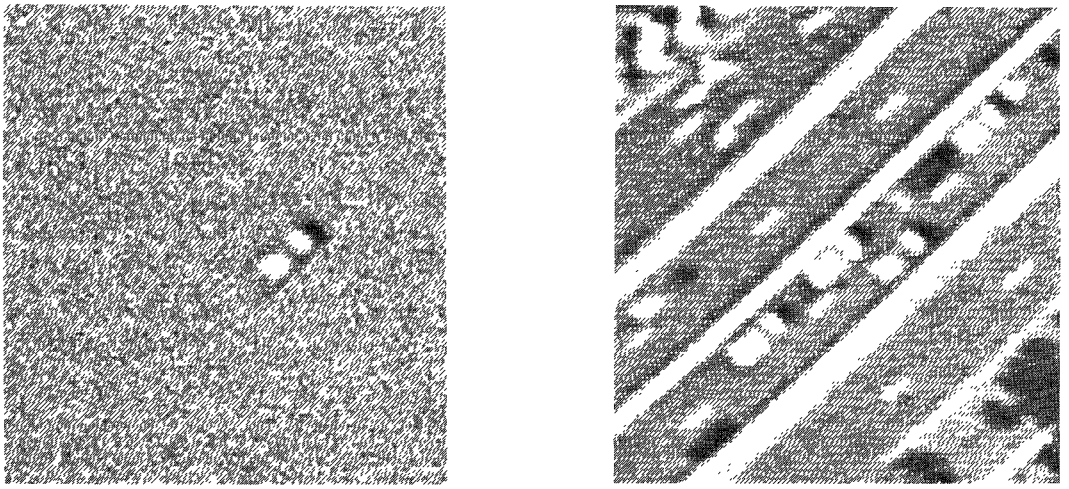
Object recognition is difficult because a combination of factors must be used to identify objects. These factors may include restrictions on allowable shapes, the semantics of the scene context, and the information present in the image itself. We next present two examples illustrating the importance of context in interpreting images. Note that the human reader will experience the same types of confusion that computer systems do if the scene context is not clearly understood; even human beings require training to interpret accurately images of the type presented.

Consider first the image in Figure 2a without reading the caption. In isolation, it is nearly impossible to identify the object in the center of this image. This same object also appears in the same position in the image of Figure 2b. With no further information, it is still difficult to identify the object. Finally, given the context information that the image in Figure 2b is an aerial image of a highway, the object is more easily recognized as an automobile. Cultural context plays a central role in enabling us to interpret the scene.

As our second example, consider the photograph in Figure 3a, showing a cluster of rocks lying on light-colored ground. What we want to illustrate here is the importance of context assumptions about lighting and shadows. If we have learned to expect sunlight to fall on the top of the dark-colored rocks, forming even darker shadows on the soil, we can form a very three-dimensional interpretation of Figure 3a. If, however, we *interchange* light and dark, as in Figure 3b, our expectation is confounded, and most of us will no longer see a consistent 3D picture.

Two more simple operations on the images can help us isolate what is important to our perceptual process: In Figure 3c, we show a thresholded binary image that effectively paints the shadows black. This cartoonlike image is relatively easy to see as a 3D scene; it may help to squint your eyes slightly. If, however, we paint the shadows *white* in the original gray-scale image, as in Figure 3d, most viewers will again find it impossible to recover the 3D shapes. The intended lesson is this: Although the thresholded Figure 3c retains much of the important visual information in the original image, Figures 3b and 3d have become uninterpretable because the 3D cues we have learned to expect have been obliterated. These examples argue strongly that computer systems (or humans, for that matter) need appropriate context models even at a very low level of the data processing procedure in order to carry out object recognition and scene interpretation.

Object recognition is analogous to another difficult problem—the interpretation of an audio signal as a sensible sentence. For simple cases, it may work to extract all the words from the audio signal in one pass, then send the words to a parser. In real-world applications, it may be absolutely essential to exploit the context of possible parses *during* the processing of the audio signal to get the correct set of idealized symbols and their interpretation. The visual context is equally important in object recognition.



(a) (b)

**Figure 2.** An illustration of the importance of context. (a) One automobile in isolation; (b) image of automobiles in an aerial highway scene.

### Illustrating the Drawbacks of Simple Approaches

The simplest approaches to object recognition rely entirely on local operators that analyze the photometric statistics of the image (e.g., measured light intensities). Since real objects are defined by their geometric and semantic characteristics as well as their statistical properties, these methods may fail to identify objects properly. Although we are excluding methods relying exclusively on local image statistics from our treatment, it is important to have a qualitative understanding of their characteristics.

For example, depending on arbitrary parameter settings, edge detector methods will either produce so many edges that relevant information cannot be perceived in the clutter or will fail to extract edges that are crucial for the interpretation process. Similarly, region segmentation algorithms will either undersegment (combine semantically meaningful objects) or oversegment (break coherent objects into unrecognizable pieces). These failure modes are inevitable because the statistical techniques used fail to take higher level geometric and semantic knowledge into account. Examples of these phenomena are shown in Figure 4 for three such methods: a histogram-based segmentation system [Laws 1984; Ohlander et al. 1978], an edge operator [Canny 1986], and the zero crossings of differences of gaussians [Marr 1982]. Many object recognition approaches depend on the assumption that the outlines appearing in some chosen *single image* defined by such methods will correspond directly to objects (buildings in this case); often, however, this assumption is simply not true.

To reiterate: Since a major task in the object recognition process is to outline areas in an image identifiable as model instances, it is tempting to use one of the above edge detection or region segmentation methods by itself. This does not work in general, because such methods have no conceptual *model* for what they are looking for—with a given set of parameters, one of these methods will draw the *same* outline whether it is looking for tadpoles or airports! These simple techniques may, however, be useful when we need a *starting point* for a more sophisticated analysis.

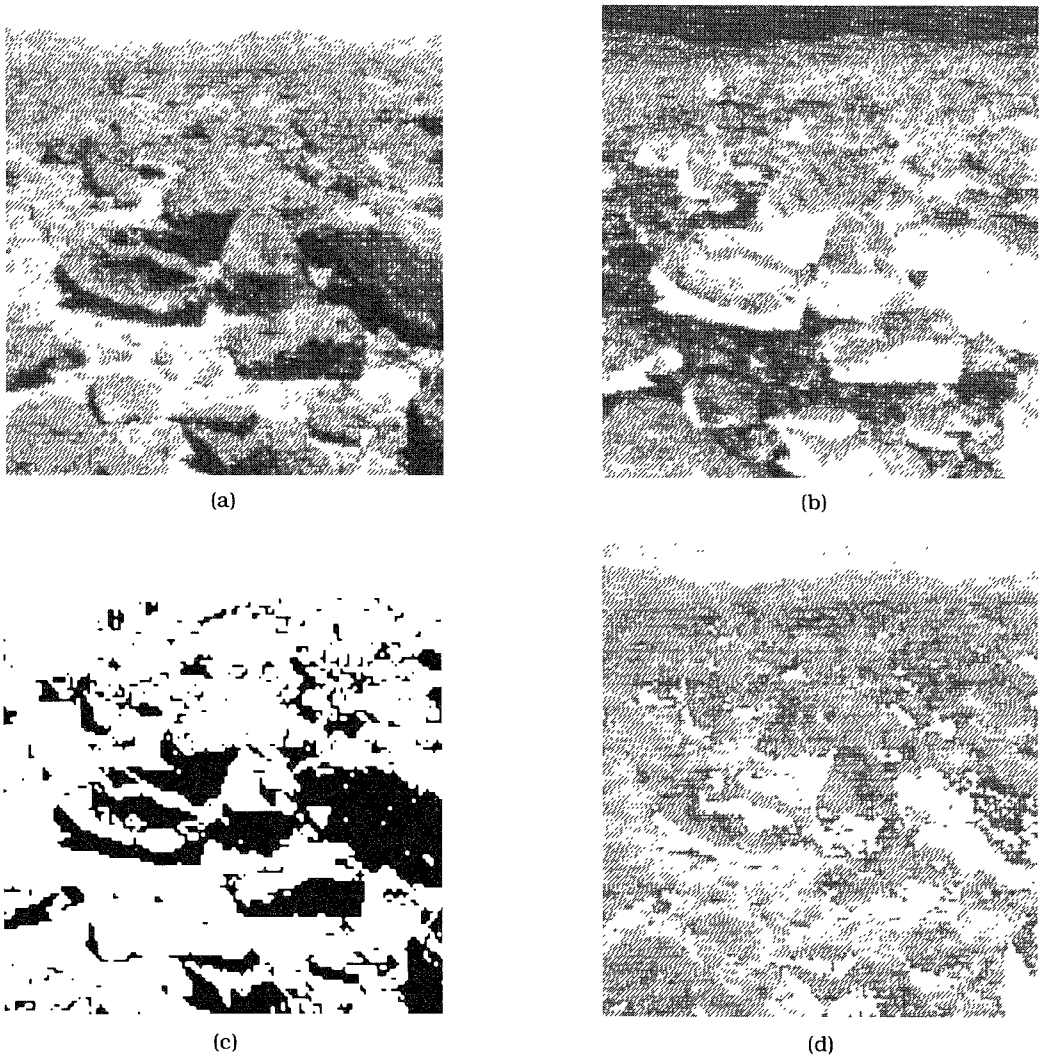



**Figure 3.** Four images of a rock scene: (a) Normal image; (b) image with the pixel intensities reversed; (c) thresholded image with shadows black; (d) image with shadows replaced by white. Shape cannot be deduced from shading alone in these examples, as all shape perception disappears in (b) and (d).

We have now argued that local photometry does not fully characterize objects in the real world, so effective object recognition procedures must incorporate model knowledge or context. In order to implement procedures that achieve this goal in a feasible fashion, we must adopt appropriate computational strategies. The classification and selection of such computational strategies is the subject of the remainder of this paper.

## CATEGORIZATION OF OBJECT RECOGNITION SYSTEMS

We classify the computational strategies used for object recognition according to two main characteristics: their suitability for complex image data and their suitability for complex models. The motivations for choosing these two features are the following:

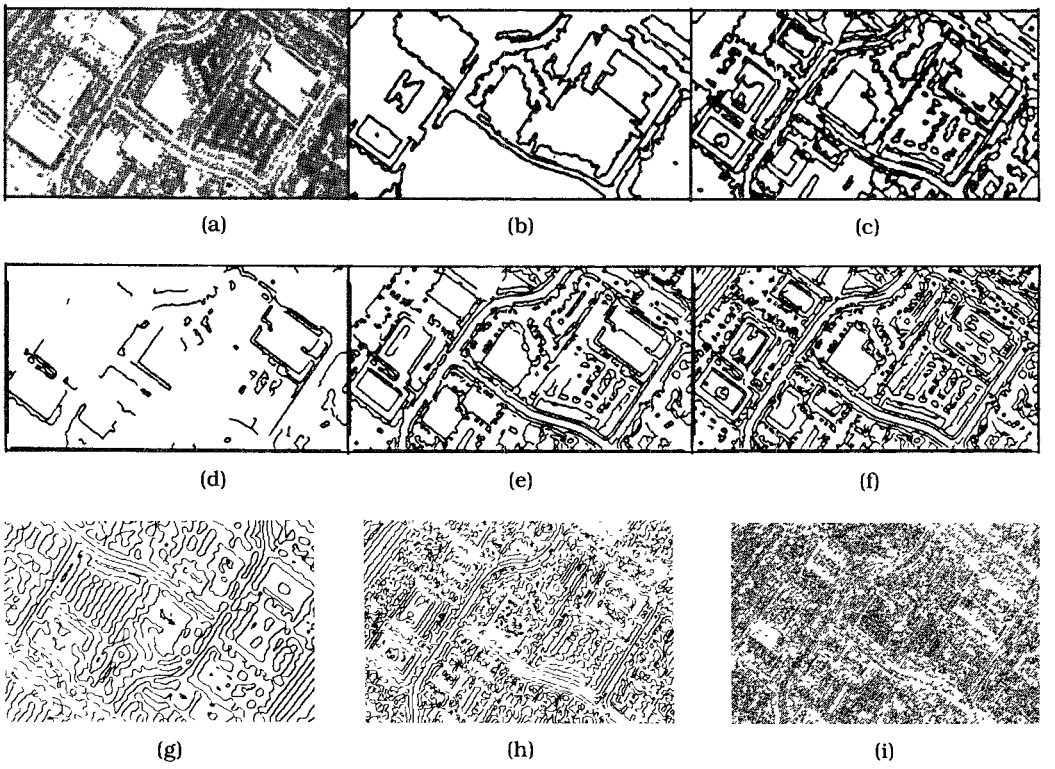



**Figure 4.** (a) An image of an aerial suburban scene; (b) a segmentation with undersegmented roofs [Ohlander et al. 1978; Laws 1984]; (c) oversegmentation resulting from a different parameter choice; (d) (e) (f) Canny edge images computed with progressively lower-edge strength thresholds [Canny 1986]; (g) (h) (i) zero crossings of differences of gaussians with progressively decreasing widths.

- **Complexity of the Image Data.** First, we define *data complexity* to correspond roughly to the signal-to-noise ratio in a digital image; an image with semantic ambiguity therefore corresponds to noisy, or complex, data. For example, if the data naturally have very good characteristics, we have the analog of an error-free sentence given to a sentence parser. Examples include data consisting of perfect (e.g., human-generated) outlines of model instances throughout an image or image data in which all houses in an aerial image have perfectly lit white roofs against a black background. Our only concern in this situation is to produce a correct set of labels without regard for how the set of symbolic outlines to be labeled were inferred from the data; in this case we call the data *simple*. If the modeled object characteristics are not unambiguously and completely encoded by an external process or by the photometric statistics, however, the task of extracting plausible model instances from the data is a major undertaking that often cannot be separated from the symbolic interpretation. Data with poor resolution, noise, or photometric anomalies (e.g., occlusions or cloud cover) typically require specially designed methods for the extraction of model instance hypotheses. Similarly, in images with easily confused false model instances, we need special methods to distinguish the correct objects from the false ones. In these latter cases, we refer to the data as *complex*.
- **Complexity of the Model.** If the model is defined by a simple criterion

like a single shape template or the optimization of a single function implicitly containing a shape model, no other context may be needed to attach model labels to the scene. If many atomic model components must be assembled or hierarchically related to establish the existence of the desired model instance, however, complex data structures and nontrivial search techniques may be required. Thus model complexity is indicated roughly by the levels of detail in the data structures and in the techniques required to determine the form of the data organization.

We are thus led to the four major classes of computational strategies that populate our category space; they are summarized schematically in Figure 5.

- **Feature Vector Classification.** Feature vector methods rely on a trivial model of an object's image characteristics and are typically applied only to simple data. Feature vector methods are well understood and treated in many textbooks [Duda and Hart 1973; Tou and Gonzales 1974]. However, for completeness we have chosen to include a brief description of these techniques because they can be very useful starting points for more sophisticated applications.
- **Fitting Models to Photometry.** When simple models are sufficient but the photometric data are noisy and ambiguous, a number of methods that extract simple model instances may be effective. Such methods search for features with predetermined global shapes and photometric properties. Methods may use rigid models, depending on a limited set of parameters, or flexible models, specified by a set of generic constraints on object characteristics. Detailed discussions of two typical examples of this method, the Hough transform and the snake method, are given in Appendix C, Sections C.1 and C.2.
- **Fitting Models to Symbolic Structures.** When complex models are required but reliable symbolic structures can be accurately inferred from simple data, procedures that tie these structures into complex model hierarchies may be appropriate. Such approaches typically look for instances of objects by matching data structures that represent relations among object parts and may use a hierarchy of intermediate models to prune the search tree. Detailed discussions of two typical examples of this method, the HYPER and ACRONYM systems, are given in Appendix C, Sections C.3 and C.4.
- **Combined Strategies.** When both the data and the desired model instances are complex, successful object recognition requires a combination of strategies. Detailed discussions of two typical examples of this method, the 3DPO and the minimal description length (MDL) method, are given in Appendix C, Sections C.5 and C.6.

Subsequent sections deal systematically with each of the major approaches to object recognition in the literature that fall within our scope. Appendix A tabulates and classifies the selected papers reviewed in the main text. Appendix B summarizes other reviews of our subject area and contrasts the approaches used with the one presented here. Appendix C contains more detailed discussions of selected papers representing each category in our classification space.

Now we turn to the task of analyzing and categorizing the literature on strategies for object recognition. We summarize a range of papers for each of our strategy categories in order to paint a broad picture of the possible applications as well as to illustrate the breadth of techniques that are available.

## 1. FEATURE VECTOR CLASSIFICATION

### 1.1 Summary of the Technique

The feature vector classification approach is a well-established strategy that has been described extensively in the litera-

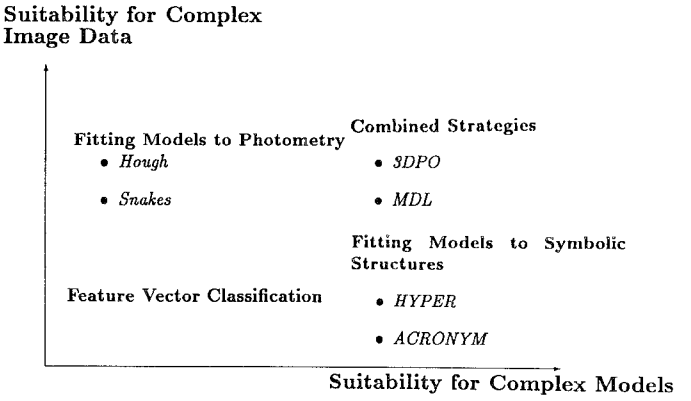



**Figure 5.** Classification space for object recognition strategies with illustrative examples. Appendix C contains detailed summaries of each example method.

ture [Duda and Hart 1973; Tou and Gonzales 1974] and has proven its usefulness in many industrial applications. Here, we give a brief summary to establish conventional terminology.

In this approach, objects are modeled as vectors of characteristic features, each of which corresponds to a point in the multidimensional feature space. Examples of features include gray value, color, infrared or ultraviolet intensity, area, perimeter, compactness, and number of holes. To use the feature vector approach, we must select which features are relevant, determine a way to measure them, and define a criterion for distinguishing the desired objects from others. For instance, to find chocolate doughnuts using a feature vector approach, we might construct groups of neighboring pixels that had similar chocolate color, compute the total number of pixels in each such group, and compute the total number of pixels in holes surrounded by each group. For each pixel, plot its color, the area of its group, and the area of its group's holes on separate axes. Pixels, all of whose values lie within acceptable ranges of these parameters, are then assigned the label "chocolate doughnut." There are, of course, many variations of such a procedure, with widely varying accuracy.

We see that once the feature space is defined, it must then be partitioned into regions corresponding to different object models; this allows the assignment of unknown objects to known object classes. The decision boundaries are usually constructed during a learning or training phase; for example, we might take a large number of sample objects with assorted feature values, make a density plot of their values, and note the boundaries of the clusters containing objects with different label names. Class selection may also be based on such techniques as bayesian decision analysis methods. The two major philosophies of feature vector classification are as follows:

- **Pixel Classification.** Pixel classification is the simplest and most straightforward application of the feature-space strategy. Each pixel is potentially a member of a different model class, and the classification of pixels is based solely on their intensities or frequency spectrum. Spectral analysis [Richards 1986; Wheeler and Misra 1980] is a well-known example of pixel classification. In this method, we might take two images of a large area of farmland using different color filters, then determine experimentally the average values of each of the two colors seen in known wheat fields. Unknown areas of the image would be labeled as wheat fields if the values of both their aver-

age colors were close to those of known wheat fields.

- **Classification of Labels.** Instead of the pixel-based approach of the previous method, we may use features that characterize regions of a partitioned image; such regions are typically obtained by some photometry-based method that groups adjacent pixels into coherent areas with homogeneous local characteristics. This requires images with simple photometric statistics. Examples of label classification can be found in the system of Groen et al. [1989] that classifies chromosomes in images of dividing cells based on features such as length of the chromosome, distance from the top to the last band, distance from the top to the darkest band, and so on, and in the system of Bergman and Mulgaonkar [1988] that uses a three-layer neural network to recognize destination address blocks in images of mail pieces using position and shape.

### 1.2 When to Use this Strategy

The feature-space approach works well when the problem involves simple models that do not include constraints relating different parts of a model and when we can restrict ourselves to either pixel classification or classification of labels with good photometry. A variety of local photometric methods and classification techniques suffice to produce accurate labels in such cases.

### 1.3 When to Avoid this Strategy

A major limitation of the feature-space approach is its inappropriateness for the representation and handling of models that include constraints on the relationships between the chosen features. The technique does not easily make use of more global information, such as spatial relationships and model context. Furthermore, unless local photometry is sufficient to distinguish the desired object completely, we cannot rely strictly on feature-vector approaches.

## 2. FITTING MODELS TO PHOTOMETRY

The most straightforward object recognition techniques are those that fit their models directly to the photometric data. These methods improve upon feature classification by incorporating more model knowledge into their procedures and replacing local pixel classification by more global considerations. As a simple example, we could tell the procedure to look for circles by finding portions of arcs with a given radius and center, as opposed to saying find any light–dark boundary in the intensity data. We divide the basic strategies into two categories:

- **Rigid Model Fitting.** The shape or photometry of the target object is known a priori; the model can be either rigid or parametric, depending on a limited set of free parameters. For more flexible models, a more sophisticated strategy is required.
- **Flexible Model Fitting.** The next level of complexity supports the use of models that are specified by generic constraints. These methods rely on an optimization procedure that finds the best fit between the model and the image data. Heuristics can be used to control the search and reduce the computation time at the possible risk of finding a nonoptimal solution.

We now examine each of these strategies in turn. The references reviewed below for each category are summarized in Appendix A.

### 2.1 Rigid Model Fitting

#### *2.1.1 Summary of the Technique*

Template matching, one of the oldest computational strategies, is the precursor of a range of more recent strategies described in this section. A *template* represents an object as a rigid curve or an image. A *metric* or similarity measure that reflects how well the image data match the template is used to find the optimal template location.

### 2.1.2 Quantifying Photometric Statistics

The simplest class of metrics quantifies similarities between two images by *correlation* (e.g., average absolute or squared differences of image pixels, normalized cross-correlation, statistical correlation) and are described in basic textbooks on image processing [Ballard and Brown 1982; Hall 1979; Rosenfeld 1969]. The basic idea is that when a (small) pattern matches up well with a local portion of a (large) image, the pixel-by-pixel differences are very small and therefore provide a clue that something special is happening in that particular local region. We refer the reader to the textbooks cited above for details of these fundamental statistical methods.

A classic example of the correlation approach is optical matched filtering. This technique actually implements correlation using optical devices that simulate the operation of an appropriate sequence of Fourier transforms [Reynolds et al. 1989]. These optical methods are interesting because they are examples of massively parallel, virtually instantaneous analog computing technology. The results are also easily simulated using (slower) digital correlation techniques.

To detect photometric similarities between the object template and the image, it is natural to use the raw image data. When other object features are more indicative, however, the raw image can be processed first (e.g., by performing a low-level operation such as edge filtering or line filtering).

Template matching is also applicable to binary images. Binary images can, for example, be obtained as the output of a low-level operator. Wallace [1988] applies boundary correlation to match rigid object-model contours geometrically with image contours. The contours are represented as tangent angle versus length curves ($\theta - s$), and the two $\theta - s$ descriptions are correlated in $s$-space. Mansouri et al. [1987] first predict the existence of a straight line segment of predefined length at each pixel location where the gradient magnitude is above a certain threshold; they then verify the prediction by applying template matching in the form of a set of statistical tests on the unthresholded gradient data in the predicted position.

This class of techniques is effective only when the model is rigid. Small changes in scale, orientation, and shape (depending on the template pattern used), and photometry (depending on the metric used) can strongly disturb the match.

### 2.1.3 Hough Transform Methods

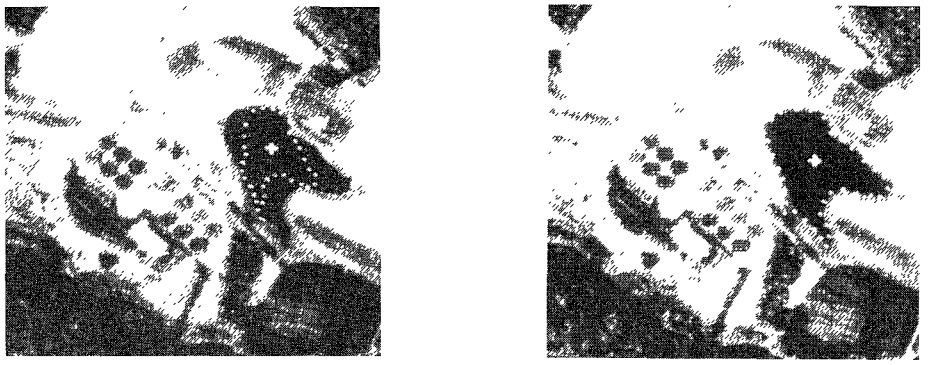
The *Hough Transform* uses templates described by a set of parameters, such as the slope and intercept of a line. By "voting" in parameter space, patterns in the image data conspire to produce local extrema at the most likely parameter values.[1] The results are relatively insensitive to partially occluded or slightly deformed shapes but take into account only the shape of the object outline.

The standard Hough transform [Ballard and Brown 1982; Rosenfeld 1969] detects curves whose shape can be described as an analytic curve. The method has been extended to detect arbitrary shape templates (the *generalized Hough transform*, [Ballard 1981]), represented as a list of boundary points. The method may incorporate parameters that translate, rotate, and scale the template.

The interested reader will find more details on the generalized Hough transform (GHough) in Appendix C. Figure 6 shows an example of the generalized Hough transform applied to a thresholded gradient image of a lake. The use of template matching of this sort is interesting because a match can still be found even with missing data.

The drawbacks of the method generally derive from the fact that a massive amount of memory and computation may be required to handle a general set of

[1] In this paper, we classify the Hough transform under template matching. Many textbooks and papers, however, consider them as two different techniques and restrict template matching to strategies carried out entirely in the image domain.



(a) (b)

**Figure 6.** (a) Image of a lake with overlaid Hough transform template. The template does not match the image; (b) generalized Hough transform match of a lake in an aerial image The correct location, scale, and orientation relating the template to the observed object in the image have been determined automatically by the Hough transform procedure

parameters. Extensive attention has been paid in the literature to methods for dynamically allocating sparse storage for the accumulators using various techniques to decrease parameter errors or reduce computation time [Niblack and Petkovic 1988] and improving performance using hardware implementations [Illingworth and Kittler 1988]. Grimson and Huttenlocher [1990] present a detailed evaluation of the reliability and other characteristics of the Hough transform.

#### 2.1.4 When to Use this Strategy

These techniques are without equal when the object's shape or photometry are *precisely* specified because they constrain the search space effectively. Furthermore, they are relatively insensitive to noise, thus making them useful in an application where occlusions may occur. In other words, these techniques work for rigid models applied to complex data.

#### 2.1.5 When to Avoid this Strategy

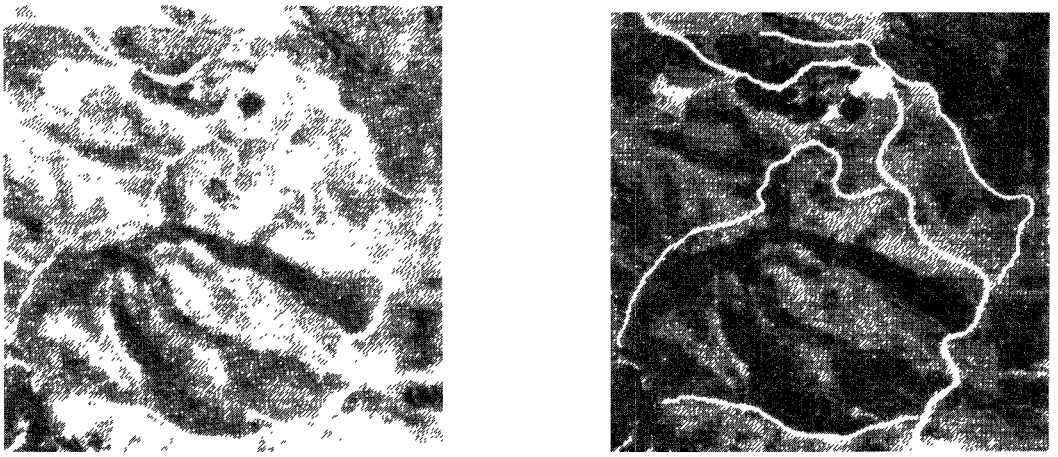
The power of template-based approaches stems from the exact knowledge of the target object's shape or photometry and disappears when such knowledge is not available. Another drawback is that it is difficult to handle a large number of model types at once; when a large number of models must be matched to the data simultaneously, we should consider variants such as geometric hashing [Kalvin et al. 1986; Lamdan and Wolfson 1988]. When the template style of model definition is not applicable, methods such as those described in the following section may be useful.

### 2.2 Flexible Model Fitting

#### 2.2.1 Summary of the Technique

Whereas template matching is restricted to rigid or parametric object models, our next class of computational strategies uses more *flexible* models, specified by a set of generic constraints on object characteristics such as smoothness, rectilinearity, curvature, compactness, symmetry, and homogeneity. The fit of the model to the image data is usually measured by an objective function, and matching is performed by minimizing this measure.

The basic idea of flexible model fitting is similar to least-squares fitting. As a simple example, suppose we have a collection of data points and a randomly chosen line; then the least-squares solution can be found experimentally by wiggling the line around until the sum of



(a) (b)

**Figure 7.** (a) Image containing thin linear features; (b) Result of $F^*$; coherent linear data structures have been automatically computed that closely correspond to the human perceptions of (a).

differences squared (this is the objective function) is minimized (i.e., an optimization procedure is carried out). See the description of Kass et al. [1987] in Appendix C for a more sophisticated example.

Like template matching, the optimization process in this method operates at the pixel level, but because of the flexibility of the model, the search may become computationally expensive. The papers discussed in this section typically require an initialization in the form of a limited search area and use only a small number of generic constraints.

#### 2.2.2 Dynamic Programming

Dynamic programming is an optimization process that is expressed as a *recursive search* [Bellman and Dreyfus 1962]. Dynamic programming is applicable only if the objective function can be expressed in terms of relationships among neighboring pixels alone.

- **Fischler et al.—F*: Iterative Path Finding in a 2D Array.** The $F^*$ algorithm described by Fischler et al. [1981] defines a path cost and iteratively finds an optimal path in an image from a starting pixel (or a set of candidate starting pixels) to a terminating pixel (or a set of candidate terminating pixels). The 2D image array is considered to be a graph in which each pixel is connected by a directed weighted arc to its eight immediately adjacent array neighbors. The pixels and arcs have an associated cost that reflects their local likelihood of belonging to the optimal path, that is, the path with minimum cost. The $F^*$ algorithm is used to delineate thin linear features such as roads and rivers on low-resolution aerial images precisely (Figure 7). The starting pixel and terminating pixel, as well as a search region, can be selected interactively from a map data base or automatically using some basic image processing operations. In this technique, costs are modified using the transform $\text{cost}' = \text{cost}^{\alpha} + b$. The constant bias $b$ tends to *smooth* and *straighten* the road track, whereas raising each cost to a power $a$ causes the path to favor strong intensities or derivatives.
- **Gerbrands et al.—Resampling the Search Region.** Whereas the $F^*$ algorithm is iterative, the dynamic programming algorithm proposed by Gerbrands et al. [1986] finds an optimal path in a cost matrix in one iteration. To achieve this, the image data in

a selected search region must be transformed into a rectangular matrix. The search region is preferably a thin, curved, band of pixels in the image; this strip is then viewed as a distorted rubber sheet rectangle. The pixel values of the new undistorted rectangle are typically quite different from grid points in the distorted pixel band, so the latter has to be *resampled*. That is, the pixel values of the new undistorted rectangle are obtained by interpolating (averaging or smoothing) the pixel values near the corresponding points in the distorted strip.

Gerbrands developed this method for the accurate detection of the left ventricular contour in cardiac scintigrams. As compared to $F^*$, the computation time of Gerbrands algorithm may be lower, depending on the computational cost of the resampling process and the computational gain obtained by avoiding iterations. A drawback, however, is that global shape constraints of the final trajectory (e.g., smoothness) in the original image may not be simply expressible in the resampled array.

- **Nuyts et al.—Parametric Search Region.** The only shape constraint that can easily be expressed in the Gerbrands algorithm is straightness in the resampled rectangular array, which is to be considered as a similarity constraint in the original image [Nuyts et al. 1989]. This means that the shape of the resulting path will closely resemble the shape of the selected search region. Nuyts et al. [1989] further extend this idea by developing an iterative dynamic programming method that finds a path similar to a *parametric curve*. The authors approximate the shape of the left ventricular wall on SPECT (single photon emission tomography) images by a piecewise elliptic curve. The search region is centered around this parametric model. After each iteration, the parameters of the elliptic curves are tuned to the shape of the detected contour. The algorithm then restarts with the updated model. This procedure iterates until the model parameters remain stable. In Figure 8 we see the results of applying this method to the quantification of the myocardium in SPECT images, where parts of the object may be missing.

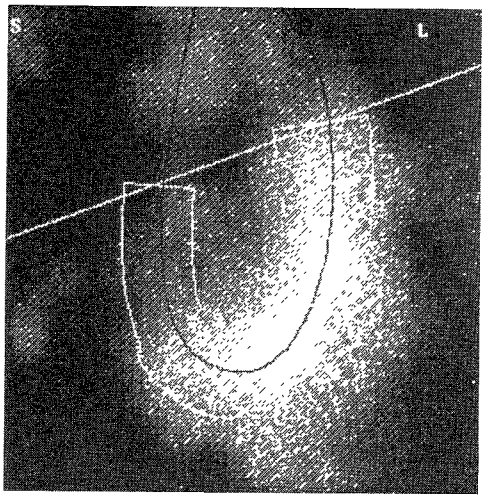


**Figure 8.** Quantification of the myocardium in SPECT images. The black center line represents the parametric model; the white lines represent the detected contour

- **Tenenbaum et al.—Optimal Path without Shape Constraint.** If there are no shape constraints, the optimal path in the resampled array can be found by taking the pixels with the minimum cost along subsequent lines perpendicular to the direction of the search region. Tenenbaum et al. [1979] used this procedure to monitor the water level of a reservoir from an aerial image, using elevation contours to guide the search for the land and water boundary. Tenenbaum also used this method to determine the precise location of a road guided by a rough prediction provided by a map and to detect, measure, or count objects whose possible locations and orientations in the image can be constrained by a map.
- **Yamada et al.—Noniterative Procedure without Resampling.** The dynamic programming matching

method by Yamada et al. [1988] does not require resampling and finds the solution in one iteration. The method is applied to extract kidney glomeruli in microscopic images. Glomeruli are more or less *circular* structures. The shape of the fitted curve is restricted to be *piecewise linear*. (The direction of each line segment is fixed.) Yet, the length of each vector is allowed to vary within a given range, and the number of segments can easily be increased in order to give the model the necessary shape flexibility. Because of this flexibility, the model is not comparable to the parametric object model used in template matching.

One difficulty of this application is that the fitted contour has to be closed. This constraint is added a posteriori by applying a distance criterion to the results of the dynamic programming process. Further improvement is obtained by cycling around the object more than once. Although the final result is theoretically not globally optimal, the method performs efficiently on the examples shown in the paper.

- **Maitre and Wu—Matching Segmented Images with Line Drawings.** Maitre and Wu [1987] applied dynamic programming to *binary* images in order to match them with line drawings such as cartographic maps or sketches. They show results of registering coast lines in satellite images, where the data are subject to noise, occlusions, and distortions. A binary image is obtained by gradient thresholding. Only the M highest edge candidates are retained (M on the order of 400). Due to the ambiguous photometry, this binary image contains noise, breaks, and distortions. Only planar translations between the binary image pattern and the model are further allowed, yet the concepts could also be directly applied to rotated and perspective views. Problems of distortions are easily overcome using dynamic programming to match the edge candidates to the lines in the map. To bridge existing gaps between the different parts of the path and to find the starting and final point automatically, the authors have introduced the concept of the "virtual state," which suspends broken paths until they can be reconnected.

  The method appears to be well suited to problems of image registration, in which the model is available as a line drawing. It is not applicable to models expressed by some generic constraints like smoothness and rectilinearity, which imply that the optimal path may not coincide with the edge candidates extracted by a local operator.

#### 2.2.3 Gradient Descent

The above tracking algorithms find a global optimum of the objective function in their search window but require the use of constraints that are local in the image data. In contrast, the energy-minimizing approach of Kass et al. [1987] can use nonlocal geometric constraints but may converge to local optima instead of global ones.

In this technique, contours are defined as curves, called *snakes*, that can deform themselves from a given initial position to the nearest local optimum of an objective function. This measure typically includes shape constraints, image constraints, and external constraints. Kass et al. [1987] use the shape constraints to enforce rigidity and elasticity by constraining the first and second derivatives of the curve. Other examples of shape constraints are rectilinearity, parallelism [Fua and Leclerc 1990], and radial symmetry [Terzopoulos et al. 1988]. The image constraints can be designed so that lines or maxima of the image gradient, for example, attract the curve. These constraints can also take the area enclosed by the curve into account and force the curve to find homogeneous regions [Fua 1989]. Finally, external constraints can be introduced to attract the curve toward particular places in the image; these constraints may correspond to either interactively specified forces or forces relating model compo-

nents [Kass et al. 1987; Witkin et al. 1987a]. The optimization procedure may even involve multiple snakes interrelated by constraints. Kass et al. [1987] mention the example of a stereo snake, which is a pair of correlated curves with smoothly varying disparities in a pair of stereoscopic images.

In the absence of shape constraints, gradient descent techniques converge slowly because there are an extremely large number of degrees of freedom. If snakes are treated as physical curves with linear shape constraints moving in the potential defined by the objective function, the optimization can be performed by solving the dynamic equations of the system. Other approaches to optimization can be found in Gardin and Meltzer [1988], where messages are passed between neighboring snake "molecules," and in Fua [1989], where global geometric constraints are applied at every iteration. All these implementations can be parallelized by allowing all the points on the curve to move simultaneously.

Snakes tend to get caught in undesirable local minima. One way to overcome this problem is *scale-space continuation* [Kass et al. 1987; Witkin et al. 1987b], which initially smooths the search space so gradient descent is likely to find a good approximation for the global minimum, then repeatedly reduces the smoothing. Another approach is *simulated annealing* [Kirkpatrick et al. 1983], which randomly chooses to change its state *up*, out of the local minimum, to see if there is another more desirable minimum nearby. More details about gradient descent methods are given in Appendix C.

#### 2.2.4 Closed-Form Solution

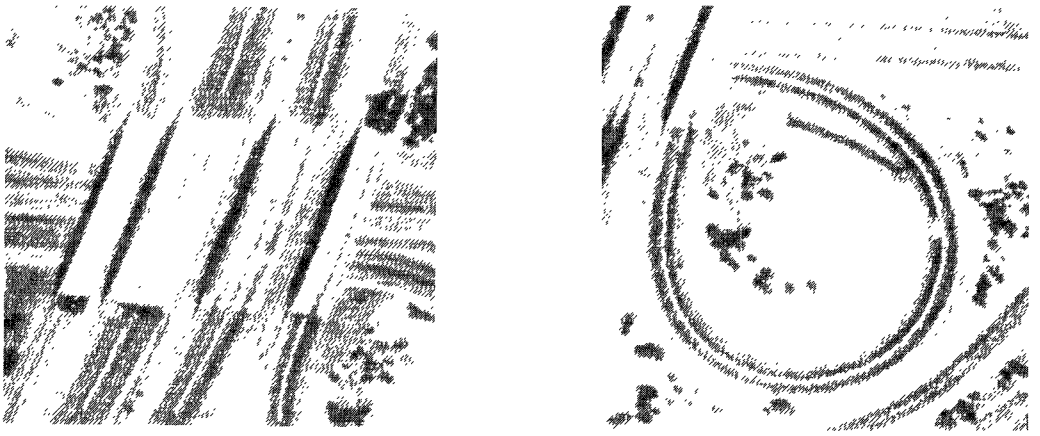
Both dynamic programming and snakes search for the optimum of an objective function derived from the pixel data. A closed form solution may exist when the constraints are carefully chosen.

• **Premoli et al.—KAMRI: Knowledge-Aided Minimum Radial Inertia.** Knowledge-aided minimum radial inertia (KAMRI) [Premoli et al. 1989] uses the following constraint functions: (1) radial inertia defined over the gradient image (the image resulting from numerically differentiating the pixel intensity values of an image), (2) the distance to a shape template of the searched contour, and (3) a smoothness constraint. (The radial inertia is, roughly speaking, the sum of the squared difference between the radius of a pixel in the gradient image and the corresponding radius of the shape template contour expected along the same line from a chosen origin.) Minimizing the radial inertia forces the curve to follow rapidly changing (high gradient) areas, while the shape template imposes a similarity constraint. This model is roughly comparable to the model used by Nuyts et al. [1989], except that the parametric template used by Nuyts is more flexible. The method of Premoli et al. uses a scale factor as the only parameter and furthermore requires that the centroid and orientation of the projected template in the image approximate those of the image pattern to be outlined.

The fitted curve must be a cubic spline. Theoretically, this further constrains the shape of the optimal contour, yet the high number of analytic curve parameters makes this constraint quite weak. Using the parametric form of the curve and all the constraints, Premoli et al. show that the objective function is a *quadratic form* in the unknown variables. Hence, the minimal value can be expressed as a closed form in terms of known values.

#### 2.2.5 Relaxation

The relaxation strategy iteratively locates and eliminates—or relaxes—the relational inconsistencies among the candidate interpretations. Its computational mechanisms are well suited to parallelization. Relaxation has been used for grouping coherent pixels with similar



(a) (b)

**Figure 9.** The results of Quam's road tracker are shown in the white overlays. Both the road trajectory and potential anomalies are marked. (a) When the tracker encounters a surface change, it extrapolates ahead and tries to reacquire the road; (b) result for a freeway interchange on-ramp loop; this example is interesting since the road curves rather tightly, and the road surface changes at approximately the same place where the road trajectory changes from a circular arc to a straight line.

characteristics into the most likely interpretation. By propagating the elimination of relational inconsistencies throughout the image, photometric ambiguities such as image noise can be resolved if the relaxation scheme converges to the desired optimum.

- **Murray and Buxton—Image Segmentation into Spatiotemporally Continuous Regions.** Murray and Buxton [1987] use *stochastic relaxation* to segment scenes consisting of a *fixed* number of moving planar surface patches. The algorithm looks for the interpretation of a field of optical flow data with the maximum a posteriori probability (MAP). As shown by Geman and Geman [1984], stochastic relaxation is a form of simulated annealing that converges if the annealing schedule is slow enough. The MAP criterion includes terms expressing how well the current interpretation explains the measured data and how well the interpretation conforms to the prior expectations of a sensibly organized flow field. To formulate the objective function, it is assumed that each surface patch in the scene is spatially and temporally continuous and that the optical flow data contains Gaussian distributed noise. An additional term is added to express the cost of introducing various line discontinuities, such as corners and T-junctions.

### 2.2.6 Heuristic Pruning

The previous strategies rely on analytic optimization techniques. When the search space is too large, intelligent heuristics are needed to constrain the search. In such cases, the objective function is not necessarily explicitly stated but may be embedded in the heuristic procedure (e.g., A*, described in Pearl [1985] and Fischler et al. [1981]). Below are some examples in which heuristics are used to *prune* the search space at the possible risk of finding a nonoptimal solution.

- **Quam—Heuristic Road Tracking Using Context-Adaptive Cross Correlation.** In Quam's [1978] procedure for tracking roads and detecting vehicles in aerial images, a context-adapting *heuristic search* method is used to support a *dynamically chang-*

*ing* model of road photometry. The approximate position of the road center ahead is defined based on past road center points and directions using parabolic extrapolation. This position is optimized by cross correlating (template matching) road cross-section intensities along a line perpendicular to the road direction with the current road cross-section model. Deviations from the model indicate potential road markings and vehicles. Figure 9 shows an example of the application of this approach; note particularly how dramatic changes in the road characteristics can be accommodated.

Quam's method uses a flexible model in the sense that the model template is dynamically updated based on the history of previously aligned road cross sections. *Similarity* between image areas centered around neighboring pixels is in fact the only constraint that characterizes the model.

Although the road tracker yields a *locally* optimal path, a globally optimal solution is not guaranteed. Typical of such heuristic line trackers is that slight intermediate displacement errors may extrapolate into large deviations. To avoid this effect, it is necessary that each step along the trajectory be reliable. In the case of Quam's road tracker, for example, it is assumed that the local shape of a road is approximately parabolic and that the photometry changes smoothly along the road trajectory (a few anomalies and road surface changes are allowed).

- **Zhang and Simaan—Model-Driven Seed Growing.** The system by Zhang and Simaan [1987] partitions seismic images into meaningful regions. In particular, they analyze a seismic image of the Gulf of Mexico. The image is partitioned into regions that differ in sediment compaction and regions that include shale ridges and salt domes. Initially, small clusters, called *islands*, are found by clustering discriminant texture features with high (.9) probability of belonging to a particular region of common signal character, that is, a region with a typical sediment compaction or a region that includes shale ridges and salt domes. The *islands* provide a context that is subsequently used as a constraint when growing the seeds into larger regions of common signal character. This context includes knowledge about the relative position of the regions, their size, and their topology. For example, regions composed of salt domes and shale ridges are not layered but are expected to have nearly vertical sides.

#### 2.2.7 When to Use this Strategy

Strategies that support flexible models are best adapted to situations in which an initial guess for a model shape instance can be easily supplied. Computation time is then reduced by an initialization in the form of a limited search region. The effectiveness of the method depends strongly on the appropriateness of the modeling primitives that are searched for and thus is a natural strategy when well-known models, described by a limited number of photometric and shape constraints, are available. Complex scenes may require models with multiple components; these components typically need to be combined in a nontrivial way in order to find a practical strategy for finding the optimal solution. This method can work well in the presence of incomplete or noisy data, provided natural limitations on the size of the search space can be imposed. Its effectiveness is further enhanced when appropriate algorithms for minimizing the cost function are available.

#### 2.2.8 When to Avoid this Strategy

Fitting flexible models directly to the photometric data is very sensitive to the completeness of the model and the appropriateness of the image data. For example, the snake evaluated using edge data alone will give spurious bleeding or premature termination if the situation requires a model that checks the area signature of the hypothesized object in addition to the edge signature. With-

out additional information or control of the search strategy, apparently optimal solutions that do not correspond to the desired objects are found. As the model complexity further increases, these strategies become computationally too expensive and multilevel (hierarchical) strategies become necessary. Approaches to remedying this deficiency are the subject of Section 4.

## 3. FITTING MODELS TO SYMBOLIC STRUCTURES

Fitting models to symbolic structures assumes that a set of features has been reliably extracted from the image data by some preprocessing operation. These features are usually found by a local statistics-based operator, without using shape information or contextual scene knowledge. This process is often referred to as *segmentation*. The features, however, may also be the raw pixel intensities or even labels produced by a method such as template matching or gradient descent, discussed in Section 2, thereby yielding a hybrid strategy. In these cases, the subsequent matching processes use the symbolic output of the initial process without referring back to the image. The major categories are as follows:

- **Graph Matching.** Objects are modeled as a relational structure or graph of primitives. The nodes are components of the object or scene, whereas the arcs denote relationships. Labels are assigned by searching for the optimal match between the model graphs and the graph derived from the image data. Heuristics can be used to prune the search tree and reduce the computation time at the possible risk of finding a nonoptimal solution.
- **Composite (Hierarchical) Model Fitting.** A reduction of the search space is obtained by working hierarchically, that is, by finding partial matches using a hierarchy of intermediate models and then refining them.

We now examine each of these categories in turn. Appendix A summarizes the references reviewed below for each category.

### 3.1 Graph Matching

#### *3.1.1 Summary of the Technique*

Relational matching overcomes the major inadequacies of pattern recognition by providing a representation for relational constraints. Objects or scenes are represented as relational structures whose nodes are subparts and whose arcs are relationships between the nodes they connect. The problem of matching relational structures is representable as one of optimizing some objective function. Heuristic search can be used to prune the search tree and reduce the computation time at the possible risk of finding a nonoptimal solution.

#### *3.1.2 Search*

The simplest form of relational matching techniques searches for sets of labels and relations that match subparts of the graph, assuming that an initial set of labels and relations has been extracted from the image by some preprocessing operation.

- **Murray—Depth-First Recursive Search.** The most straightforward approach to graph matching is to require that both graphs be identical (*isomorphic*). This strategy is used by Murray [1987] to recognize rigid polyhedral objects using sparse and error-prone point measurements of surface orientations and scaled depth. Relational constraints to be satisfied are of the following type: "If an interpretation pairs sensed data points $P_a$ and $P_b$ with model faces $i$ and $k$, respectively, then the range of angles between the vector in the direction between the two points and the sensed normal at $P_a$ must overlap the range of possible angles measured from the model." That is, points on a side face of a cube can only lie in a specific region when viewed from a point on the top face of the cube; families of

such relations place severe restrictions on what face a point can belong to. The process of matching sensed data points and model facets is performed by a depth-first recursive search.

Although Murray [1987], in principle, could be thought of as a hybrid model that also involves hypothesis verification, it is placed here because its treatment of graphs makes it an ideal example of the process of isomorphic graph matching.

- **Bolles and Horaud—3DPO: Maximum Clique Finding.** Slight photometric anomalies, such as occluded parts of 3D objects, may make the requirement of finding identical graphs too strong for many real-world applications. A less stringent criterion is to require that both graphs contain a *subisomorphism*, that is, an identical subgraph. Subisomorphisms in two graphs can efficiently be detected by maximum clique finding in an association graph. This method has been used in the 3DPO system [Bolles and Horaud 1986] to find the best match between features extracted from a range image and their corresponding interpretations.

  The elementary matching process of 3DPO is an ideal example of subisomorphic graph-matching methods; however, the system also has the capability of using more elaborate strategies, which are discussed in more detail in Section 4 and Appendix C.

The graph-matching techniques described so far are acceptable provided the graphs or subgraphs to be identified are identical; this is rarely the case unless the criteria for finding compatible nodes or arcs are weak. To compare nonidentical graphs or to compare identical graphs obtained with weak similarity measures we must use a distance measure to evaluate the similarity between graphs.

- **Mulgaonkar et al.—Matching Nonidentical Graphs.** The recognition scheme implemented by Mulgaonkar et al. [1984] uses relational and rough geometric information about 3D, manufactured objects (table, chair, etc.) to recognize instances of the objects in single, perspective normal views of scenes. An example showing how the pieces of a chair are recognized is given in Figure 10. All models are decomposed into three basic shapes: sticks, plates, and blobs. The model further consists of binary and ternary relations and related angles. For example, the back of a chair and the seat form a plate–plate connection in which the edges of the plates touch each other. Using the shape and relational constraints, graph matching is performed as a sequential tree search. The relational distance measure is defined as the sum of the number of relations of the model that fail to carry over to the image, normalized by the total number of relations in the model.
- **Horaud and Skordas—Ranking Maximal Cliques.** The method of Horaud and Skordas [1989] matches linear edge segments and their relationships in a stereo image to solve the correspondence problem. A relational graph is built from each image. Compatible subgraphs in both images are found as maximal cliques in a correspondence graph. Each maximal clique is evaluated by a benefit function, calculated as the sum of the individual benefits of the nodes. These individual benefits express the similarity between the corresponding line pairs, so the best maximal clique is not necessarily the largest one.

  The work of Jain and Hoffman [1988] is another example that makes strong use of a distance measure to match relational structures. Because the strategy is hybrid and hierarchical, it is described in Section 3.2.

#### 3.1.3 Dynamic Programming

Dynamic programming is a process that recursively searches for an optimal path in the graph [Bellman and Dreyfus 1962]. It allows the solution to be efficiently computed but requires that the graph

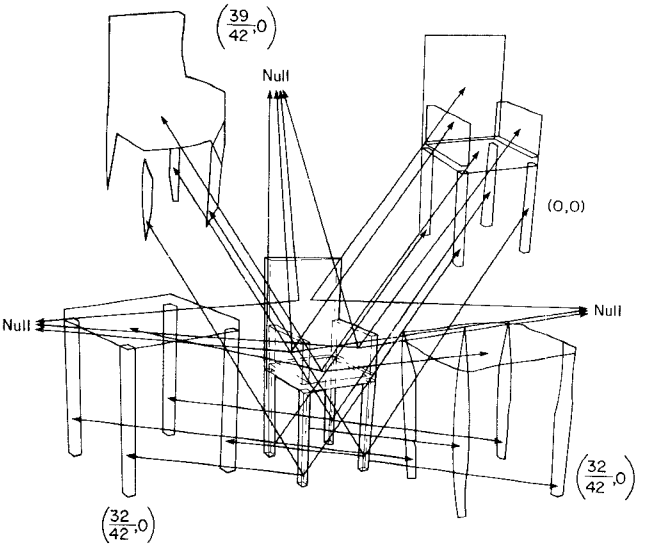



**Figure 10.** Matching results for a chair object. The lines from the model part to the various partitioned 2D views show how the parts map. The structural matching error is shown in parentheses as the first number. (Courtesy of P. Mulgaonkar.)

distance measure involve only local relationships among neighboring nodes in the graph. Dynamic programming may be used to increase the computational efficiency of the optimization procedure at the expense of using additional storage in the search.

- **Eshera and Fu—Inexact Contour Matching.** Eshera and Fu [1986] perform inexact graph matching by minimizing a graph distance measure. The contours of the object models and of the segmented patterns are represented as attributed relational graphs (ARG), that is, graphs whose nodes and branches can have attributes. The authors present two examples. The first deals with finding a 2D industrial part in an image of overlapping 2D objects. The second is concerned with the detection of an airport in synthetic aperture radar (SAR) images. Graph nodes are straight line segments, arc segments, and closed curves with length and span as attributes. Branches represent relations such as joint, intersecting, nonjoint and nonintersecting, and parallel, with attributes joint angle, angle of intersection, distance between the two line segments, and angle between the two line segments. Hence, objects are necessarily rigid 2D models. Costs are locally assigned proportional to the similarity between pairs of nodes and pairs of branches. The optimization problem can therefore be solved by means of dynamic programming.

- **Fischler and Elschlager—Heuristic Dynamic Programming.** Fischler and Elschlager [1973] represent a scene by a number of rigid components held together by springs. The springs joining the rigid pieces served both to constrain their relative movements and to measure the cost of the description by how much they are stretched. As shown in Figure 11, a face can be represented as a nose, mouth, eyes, and ears held by springs. The dynamic programming technique is used to match the various elements in the image and optimize their respective locations. Although the storage and time requirements for

dynamic programming in this work grow exponentially with the number of nodes in the graph, not all dynamic programming formulations lead to exponential algorithms. For example, if we could linearize the graph (not necessarily possible for spring-loaded templates), the dynamic programming algorithm is of polynomial complexity.

#### 3.1.4 Relaxation Labeling

Direct serial search can easily become combinatorially explosive but can be replaced by parallel techniques to make the computation feasible. Relaxation labeling [Kittler and Illingworth 1985; Rosenfeld et al. 1976] is such a technique. It is computationally identical to relaxation discussed in Section 2.2. It iteratively locates and eliminates inconsistent node interpretations. Because its computational mechanisms are well suited to parallelization, relaxation labeling has become an attractive strategy for grouping similar pixels, feature vectors, or data structures into the most likely interpretation. It has further been extended from *discrete* labeling to *probabilistic* labeling, in which the labels extracted from the image are assigned a probability that is iteratively increased or decreased based on its compatibility or incompatibility with related labels in the structure. As discussed by Faugeras and Berthod [1981], relaxation labeling can be an optimization process. The procedure is generally expected to converge to an optimal solution; however, in many of the proposed relaxation schemes this is not guaranteed. Even if the scheme converges, the result may be a *local* optimum that depends on the initial labeling. This may or may not be desirable. Stochastic optimization may be more appropriate if a *global* minimum is required [Kittler and Illingworth 1985].

- **Huffman and Clowes—Line Drawing Interpretion.** The line drawing interpretation approach associated with Huffman [1971] and Clowes [1971] is an early example of relaxation labeling [Ballard and Brown 1982; Mackworth 1973]. This strategy can analyze line drawings of complicated polyhedral scenes such as that in Figure 12. Initially, each trihedral corner in the line drawing and the lines meeting at that corner are considered as candidates for all possible interpretations. A line can correspond to a concave, convex, or a discontinuous edge in the 3D space, depending on the type of vertex with which it is associated. Conflicting line interpretations are eliminated by applying the *coherence rule*, which states that in a real polyhedral scene no line may change its interpretation (label) between vertices. By iteratively applying the coherence rule, this constraint propagates throughout the image and produces a consistent interpretation.

  Recently, the Huffman–Clowes approach has been further extended by Malik [1987] to deal with the more general class of line drawings of curved objects. Although this line-drawing work appears promising, its applicability to real-world applications is restricted because of the model simplicity and the unrealistic assumptions of good image data.

- **MSYS—Discrete Relaxation Labeling.** A more practical example of relaxation labeling is discussed by Tenenbaum and Barrow [1977]. The goal of this system is to partition an image into meaningful regions by merging small initial regions in accordance with their candidate interpretations. An example of such an interpretation for a scene of an indoor room is shown in Figure 13. Experimental results are reported in three scene domains: landscapes, mechanical equipment, and rooms. The system starts from an initial partitioning of the scene, in which the regions may have multiple interpretations. This initial interpretation set can, for example, be obtained from a training phase, during which a representative set of images is presented and pixels with the same set of possible interpretations are grouped into regions. Maps or a

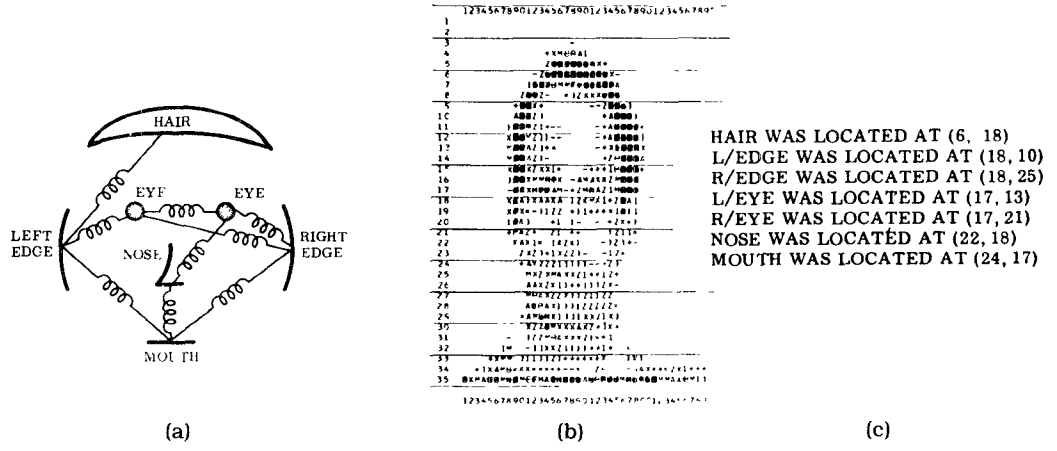



**Figure 11.** Dynamic programming approach of Fischler and Elschlager [1973]. (a) Model description of a face; (b) image of a face; (c) solution of the matching procedure. (Used with permission of IEEE; ©1987 IEEE.)

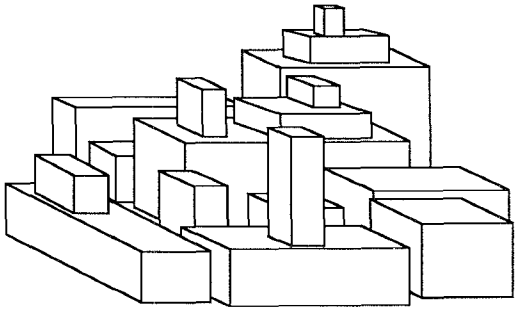


**Figure 12.** Example of the level of complexity in line-drawing scenes that can be dealt with using the Huffman-Clowes approach.

previous analysis of a similar image may also be used instead of the training samples. (An alternative method, not explicitly mentioned by the authors, would be to calculate local features, classify them using the feature-vector approach, and retain groups of pixels (islands or seeds) with a high probability of having a unique interpretation.) The result after relaxation is a set of regions with a unique interpretation, obtained by iteratively merging adjacent regions with the lowest contrast boundary and with nondisjoint interpretation sets. In contrast to line-drawing interpretation, this method has been tested on real image data and incorporates more sophisticated semantic constraints.

- **Mohan and Nevatia—Constraint Satisfaction Network.** The method described in Mohan and Nevatia [1988, 1989] is an example of relaxation where a cost function associated with a network of constraints is minimized. Linear segments extracted from aerial images are combined into structural patterns. The structural elements considered are lines, parallels, *U*'s, and rectangles. Initially, all possible structural elements found in the image are considered as candidates. Structural patterns that are consistent, such as a line and a *U* it belongs to, are mutually supportive. Inconsistent patterns, such as two overlapping *U*'s that share components, are mutually competitive. The structural patterns and the relationships of support and conflict among them define a network, with the structures serving as nodes and the relationships and compatibilities as arcs. A cost function is associated with the network, and the problem of locating the best groupings reduces to that of minimizing this cost. In Figure 14, we show the sequence of analysis. Beginning with a bare image, the approach


**Figure 13.** (a) An outdoor scene. (b) result of labeling the picture using MSYS.

first extracts every reasonable straight line candidate; these are grouped into patterns and the costs evaluated, resulting in the final optimal cluster of rectangles shown.

### 3.1.5 Heuristic Pruning

To speed up the optimization procedure, heuristics may be used to decide which among several alternative courses of action promises to be the most effective and should be explored first [Pearl 1985]. Moreover, heuristics can *prune* the search tree at the possible risk of finding a nonoptimal solution.

- **Amini, Weymouth, and Anderson—Hill Climbing.** The method of Amini et al. [1989] is an intriguing medical application of the heuristic pruning approach. The task is to distinguish inner-ear hair cells in images containing cross sections of the hair cells. An edge operator is used to find edge segments in the image. The centers of these edge segments then are treated as the vertices for groups of possible convex polygons. A depth-first search for a polygon is started in parallel for every edge segment in the image. The search ends if the initial segment and the final segment are identical. This search for polygonal structures is controlled by a heuristic rule that picks the *best* segment at every step. The cost function used to rank the edge fragments is a weighted sum of length, distance, and curvature. The intuitive idea is that segments that are longer and closer to the current segment should be more desirable because there will be less possibility of meaningless noise segments and empty gaps. In addition, the curvature of the segment decreases or increases the cost depending on its consistency with the shape of the hypothesized cell. After each step, every edge segment corresponds to a single cell-contour hypothesis.

  The advantage of hill climbing, that is, depth-first search with a heuristic procedure that orders choices, is the reduction of the computation time. The main drawback, however, is that the solution obtained by this sequence of locally optimal decisions is not necessarily globally optimal.
- **Ayache and Faugeras—HYPER: Heuristic Tree Pruning Including Hill Climbing.** As shown in Figure 15, the HYPER (HYpotheses Predicted and Evaluated Recursively) system of Ayache and Faugeras [1986] identifies and accurately locates touching and overlapping flat industrial parts in an image; the problem of handling such incomplete data is a common one in

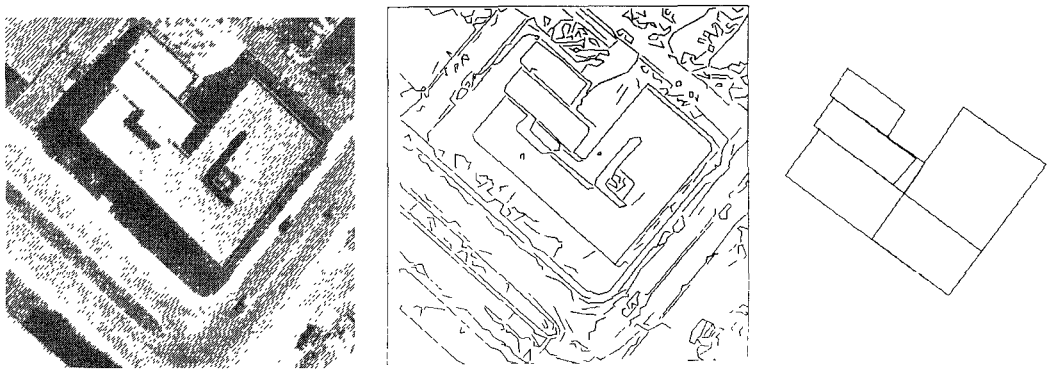



**Figure 14.** Results from Mohan and Nevatia [1989]. (a) Aerial image of a suburban scene; (b) linear segments detected in (a); (c) rectangles selected by the constraint satisfaction network. (Used with permission of R. Nevatia and IEEE; ©1989 IEEE.)

other applications as well. Object models and segmented image patterns are described by first-degree polynomial approximations to their contours. Matching is performed by a heuristic tree search procedure. The rigid model contour is iteratively matched to the image pattern segments by successively adding compatible segments to the available partial contour match. At each iteration, a dissimilarity measure between the active model segment and each image pattern segment is calculated. This dissimilarity measure is a weighted sum of three terms: the difference between the orientation of the model segment and the image segment, the euclidean distance between their midpoints, and the difference between their lengths. As in Amini et al. [1989], Ayache and Faugeras [1986] heuristically match the considered model segment with the *best* image segment, that is, the image segment with the minimal dissimilarity. More details of this approach are given in Appendix C.

#### *3.1.6 When to Use this Strategy*

The strategy of optimizing the match to information represented as a graph works best when a comprehensive graph model is available and if one has good local operators that reliably discover the features used as nodes and relationships of the graph.

#### *3.1.7 When to Avoid this Strategy*

This approach assumes that most elements of the relational structure are directly available, that is, nodes can be extracted from the data without the use of contextual knowledge. Although inexact graph matching may overcome some problems due to image ambiguities, such as occlusions [Ayache and Faugeras 1986], this assumption is generally unrealistic for images with ambiguous photometric statistics because local operators cannot be expected to achieve the required level of performance.

### 3.2 Composite (Hierarchical) Model Fitting

#### *3.2.1 Summary of the Technique*

In Section 3.1 we saw how simple heuristics were used to limit the range of labeling possibilities to be considered. In this section, we discuss a class of methods that uses hierarchical modeling techniques to limit the search; a sequence of intermediate and progressively more complete models is used to find and refine partial matches. The intermediate states in the computational process have an

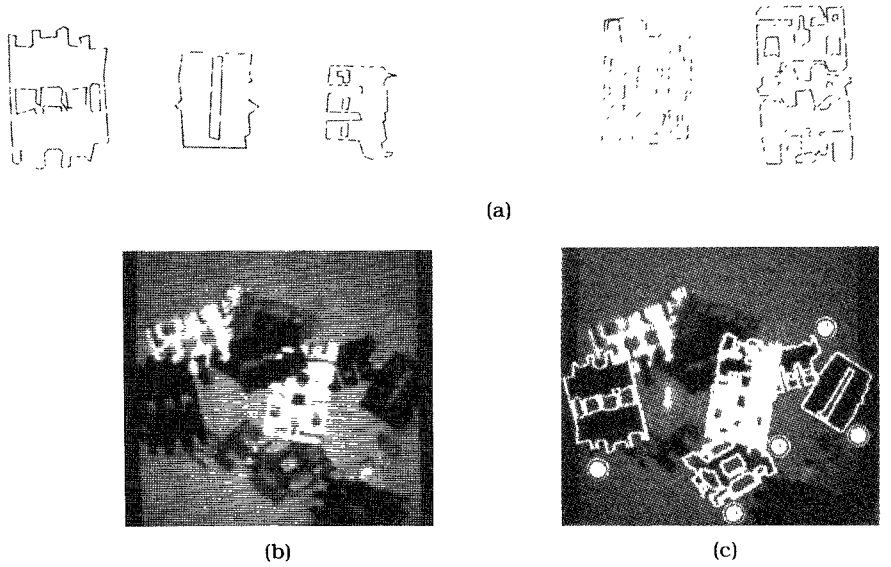



**Figure 15.** HYPER system of Ayache and Faugeras [1986]. (a) Model contours; (b) original image of overlapping flat electromechanical device parts; (c) highest ranked model instances (in white) superimposed onto original image For example, the leftmost model in (a) corresponds to the actual object seen at the left of the image (c) rotated half a turn (Used with permission of the IEEE; ©1986 IEEE )

obvious semantic meaning and create a *context* for the subsequent analysis. For example, if the problem is to recognize yellow cars in an aerial image, we might first look for all yellow patches in the image, then see which of the patches had the characteristics of a car.

### 3.2.2 Structural Grouping

The most basic methods of this class use a sequence of models that range from generic, with few attributes, to complex, with multiple attributes. The problem is divided into tasks, with particular models applied to solve each case in sequence. The gross features are dealt with first, the more specific ones next.

Since the semantic characteristics of objects are not necessarily independent, the process of finding an instance of a *partial* model is often heuristic. Unless these partial solutions are considered as hypotheses that can still be changed after verification, the final solution *may not be optimal*.

- **Lowe—SCERPO: Locating Perceptual Structures.** Hierarchical object recognition may or may not produce optimal solutions. This strategy is more likely to produce an optimal solution if the control structure is powerful enough to backtrack when necessary, thereby permitting the investigation of the complete state space. The SCERPO system [Lowe 1987], whose goal is to recognize and locate rigid 3D manufactured parts in a single gray-scale image, is a typical example of such a system. Figure 16 shows SCERPO's results for partial and final matches in an image of a bin of disposable razors. Pairs of straight lines are combined into *perceptual structures*, that is, instances of collinearity, end-point proximity, and parallelism. Next, these primitive relations are combined into

larger, more *complex structures*, such as trapezoid shapes. These "generic" structural patterns are finally used to limit the search by hypothesizing the position of the manufactured part (e.g., a razor, a stapler, etc.), which is then backprojected onto the edge data to verify the hypothesis.

- **Brooks—ACRONYM: Hierarchical Models and Constraints.** ACRONYM [Brooks 1981; 1983] has been used in applications involving a wide variety of models ranging from motors to aircraft. One of its most challenging applications is the location and identification of airplanes in aerial views of airfields. Three-dimensional geometric object classes, (e.g., airplanes) and specific objects (like a Boeing-747) are modeled as *generalized cones* and their spatial relationships. Initially, edges are combined into features such as ribbons and ellipses, which are the shapes generated by the body and the ends of generalized cones. The interpreter then looks for matches between the model as a set of generalized cones and the observed features based on the predicted ways the generalized cones could appear in the image. Interpretation proceeds by combining local matches of shapes to individual generalized cones into more global matches for more complete objects, requiring consistency among related families of constraints. In Figure 17, we show a typical ACRONYM application, with a bare image, a set of edges, the derived features, and the final consistent match to a particular aircraft model. The ability to handle families of aircraft models via the properties of their subparts in this way illustrates the use of a more complex modeling procedure than, for example, a system like SCERPO. This paper is discussed in detail in Appendix C.
- **Huertas and Nevatia—Finding Linear Structures.** The approach of Huertas and Nevatia [1988] uses a combination of information about linear cultural objects and their context in the identification process. In an application designed to detect runways in aerial images, Huertas et al. [1987] group line segments into *apars*, that is, antiparallel line pairs or parallel lines of opposing contrast. Broken apars are joined using some properties of connectedness and collinearity. The remaining *long apars* are candidate runways. Verification of these hypotheses is accomplished primarily by detection and identification of runway markings among the set of original apars and line segments. In Huertas and Nevatia [1988], the authors use a similar strategy to detect buildings in aerial images. Initially, edges are approximated by piecewise linear segments. Next, corners, defined as *near orthogonal L junctions*, are found and labeled as objects or shadow as a function of the direction of the illumination. Corners that share a line segment are grouped into more complex structures. Finally, a closed outline is classified as a building boundary if it contains a corner with a corresponding shadow.
- **Jain and Hoffman—Merging Adjacent Surface Patches.** The recognition method described by Jain and Hoffman [1988] matches models of 3D objects, described by a set of constraints on the relationships among their parts, to the detected structure of surface patches produced by a range image. (Typical constraints would involve the relative orientation of the faces of a rectangular solid.)

  The process used to produce these range-image surface patches typically produces an oversegmentation of natural object faces. The first task of the recognition system is therefore to merge adjacent surface patches into meaningful structures based on model knowledge about the boundary angles of the 3D objects. The result of this merging process is a separate relational structure of surface patches for *each* candidate object model.

  Next, the recognition system calculates a similarity measure between each object model and its correspond-

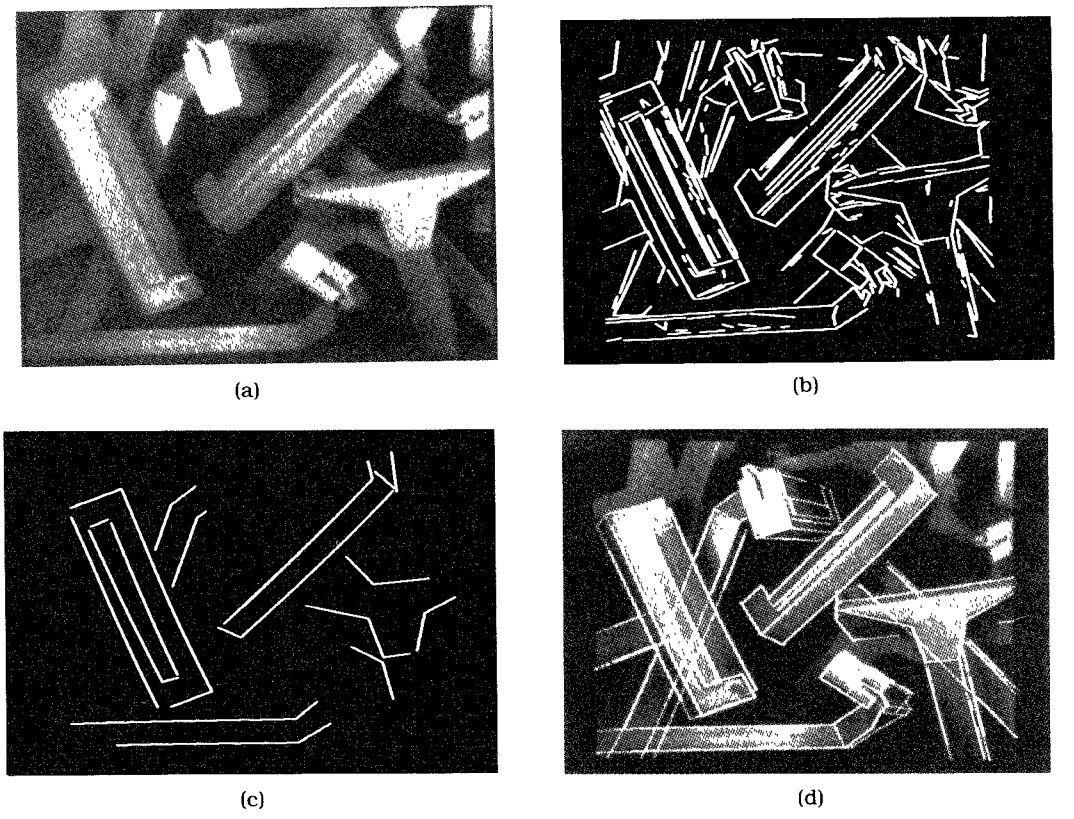



**Figure 16.** The SCERPO system [Lowe 1987]. (a) Original image of a bin of disposable razors; (b) straight line segments; (c) the most highly ranked perceptual groupings detected from among the set of line segments; (d) the model projected onto the image from the final calculated viewpoints. Model edges are shown dotted where there was no match to a corresponding image segment. (Used with permission of Elsevier Science Publishers.)

ing relational structure of surface patches. The model knowledge consists of a set of constraints on these relations giving supporting or refuting evidence for identification hypotheses.

The papers of Fua and Hanson [1987, 1988, 1991], McKeown et al. [1985], and Suetens et al. [1989] are other examples of methods that initially search for simple structural patterns in the image. They are described in Section 4 because their hybrid strategy makes strong use of methods for complex data.

#### 3.2.3 *Refining Matches Using Multiple Information Sources*

In this class of strategies, equivalent or complementary information sources are sequentially exploited. Further reliability in the object-labeling procedure can be achieved by integrating information from multiple sources. Such additional information can be either in the form of data (e.g., stereo images) or independent sources of semantic knowledge (e.g., using a library of alternately applicable road-finding operators). As in the systems just examined, the individual information sources themselves are typically used in a hierarchical fashion, with initial hypotheses being progressively refined by the application of further knowledge. Conflicts may occur and must be resolved.

- **Herman et al.—3D MOSAIC: Object Completion Using Additional Images.** The 3D MOSAIC system

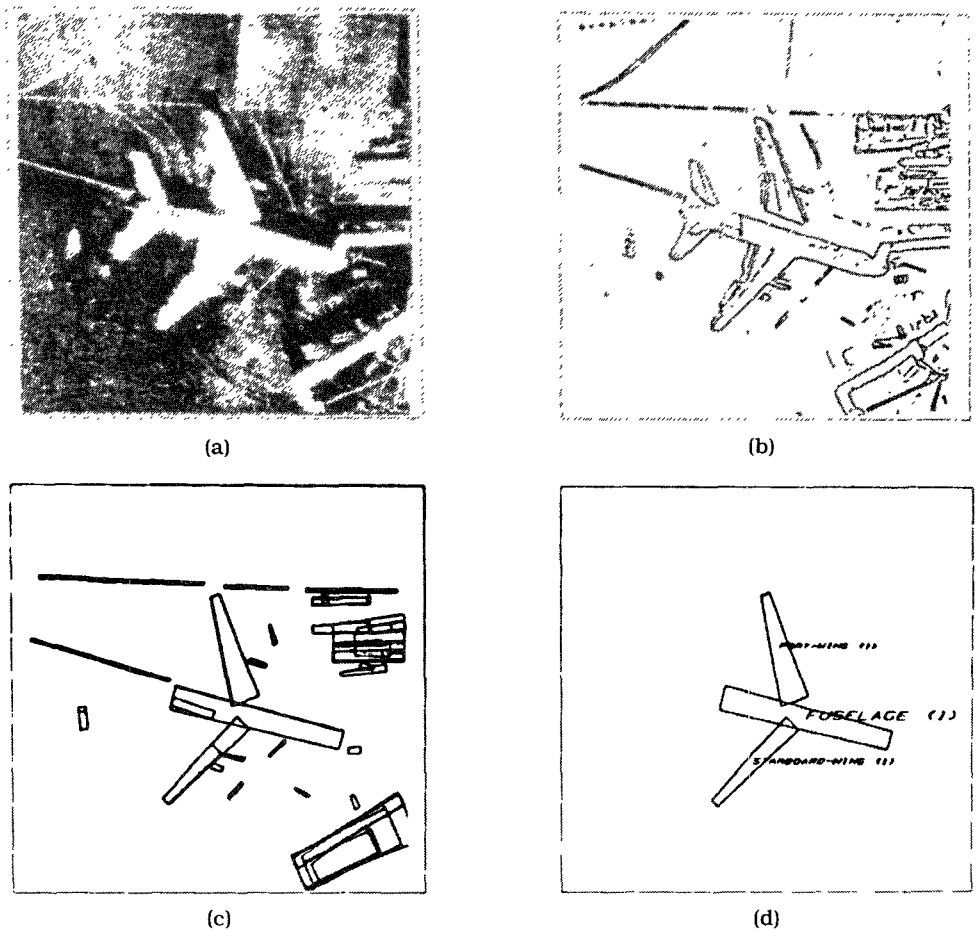



**Figure 17.** Application of the ACRONYM system [Brooks 1981, 1983] to aircraft identification. (a) Original image; (b) edges extracted from (a); (c) features such as ribbons; (d) ACRONYM result, identifying an aircraft by combining part models. (Used with permission of the IEEE; ©IEEE 1983.)

[Herman and Kanade 1986] reconstructs buildings from a sequence of monocular or stereoscopic aerial images taken from different viewpoints. It uses the multiple images as additional information sources that can be used to improve an existing interpretation.

Initially, an edge detector is applied to the images to extract straight lines. The shapes of the junctions formed by pairs of lines are labeled as an L, a T, an arrow, or a fork. Next, the 3D positions of the junctions and converging lines are calculated. If a stereo image pair is available, a cost optimization strategy is applied to find the optimal set of matching structural features. For single images (monocular analysis), new lines are first heuristically added to the image to form linear *connected* structures of junctions. Using depth cues that characterize scenes consisting of horizontal and vertical lines, the relative 3D positions of the junctions and linear segments are then calculated by propagating constraints among the line interpretations.

The result is a 3D wire frame description of the scene. It is elaborated into a surface-based description (object completion) in the next step.

Incomplete faces are completed as parallelograms or as polygons by hypothesizing missing vertices and edges. Finally, this result is improved when new views become available. Inconsistent hypothesized edges and vertices of the existing partial interpretation are replaced by newly obtained elements, and modifications propagate throughout the wire frame to maintain overall consistency. In this way the redundancy of image data partly corrects for the inaccuracies introduced by the heuristics of the system.

- **Fan et al.—Matching Image Interpretations Using Heuristic Search.** Like the 3D MOSAIC system, the system of Fan et al. [1988, 1989], matches the individual interpretations of two images, taken from different viewpoints, in order to arrive at an improved interpretation.

  The images are range images partitioned into surface patches. These patches are further grouped into graphs whose nodes represent the patches and whose arcs express geometric relationships between the patches. The result is several unlinked subgraphs that are supposed to correspond to the distinct physical objects in the scene. The subgraphs of both images are subsequently matched using heuristic search. A global match measure based on all the matched nodes defines whether the match is good enough to be accepted.

  This system uses some form of nonmonotonic reasoning; the initial grouping of surface patches into linked node structures may not be perfect. By examining the matches of different views, graphs may be merged and/or split to improve the correspondence and, as a result, also the interpretation. An example is shown in Figure 18.

- **Bobick and Bolles—Integration of Visual Information Over Time.** Bobick and Bolles [1989] incrementally construct the interpretation of an object as new views with changing resolution become available over long periods of time. The method implemented by the TraX system recognizes various outdoor objects when applied to a sequence of range images.

  The basic strategy is to consider different models when they become suitable; the selection of models is guided by the computed characteristics of the object. The models are arranged in a directed dependency graph, called the *representation space*. The TraX system, for example, includes 2D blobs, 3D blobs, superquadrics,[2] sticks, and several semantic representations including bush and tree. A new node in the representation space can become active only if one of its connecting nodes is valid. For example, if a reliable 3D blob description has been computed for the object, the superquadrics and sticks nodes can be activated. The principal indication of validity is stability over time, meaning that the same object description is computed repeatedly in subsequent images.

The work by Wang and Srihari [1988] and McKeown and Denlinger [1988] (automatic road follower, or ARF) are other typical systems that make use of multiple information sources. They are described in Section 4 because their strategy is hybrid and includes features to cope with ambiguous data.

#### 3.2.4 Knowledge-Based Systems

Solving problems by using a large amount of domain-specific knowledge has led to the notion of *knowledge-based*, or *expert*, systems. Typically, the system designer's knowledge about a complex domain evolves rapidly during development. For such applications, it is useful to state this knowledge in a form that

[2] See Barr [1981] Typical superquadric solid models are implicit functions of the form $|x|^{\alpha} + |y|^{\beta} + |z|^{\gamma} = 1$ that have spheres as limiting forms as the exponents approach 2.

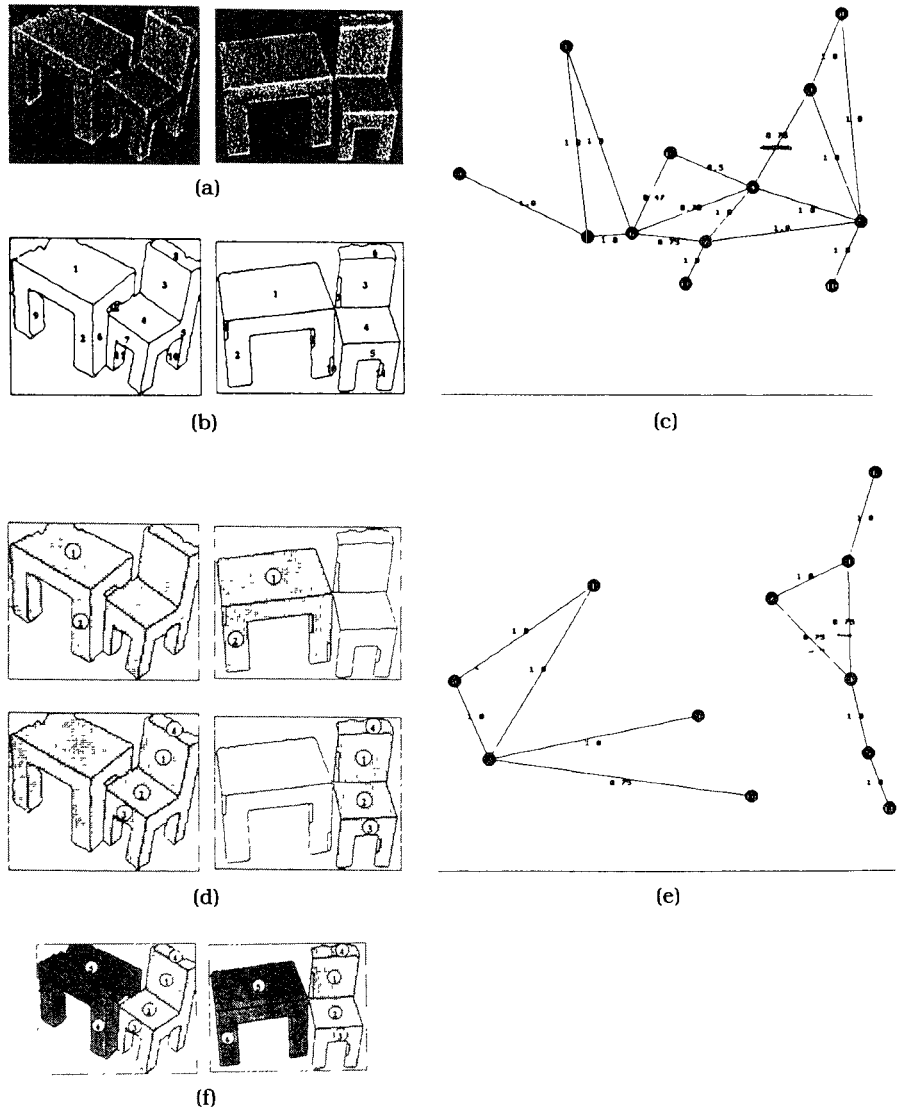



**Figure 18.** Example of the matching procedure of Fan et al. [1988, 1989]. (a) Original images; (b) segmentation; (c) graph of the left scene; (d) two possible matches before splitting; (e) graph of the right scene; (f) final match. The table and the chair in the left image touch each other In (d) there is only one object in the left view. By examining the graphs and the matches, however, it is possible to split the two objects in the left scene. (Used with permission of R. Nevatia and the IEEE; ©1988 IEEE.)

offers flexibility. Knowledge-based systems attempt to achieve this goal by using declarative languages, as in rule-based systems. Rules are to be considered as small pieces of domain knowledge, and their activation produces intermediate states with an obvious semantic meaning. Such systems therefore typically have a high degree of human understandability.

- **Ohta—Rule-Based System for Outdoor Scene Analysis.** The goal of Ohta [1985] is to interpret color images of outdoor natural scenes. The model is

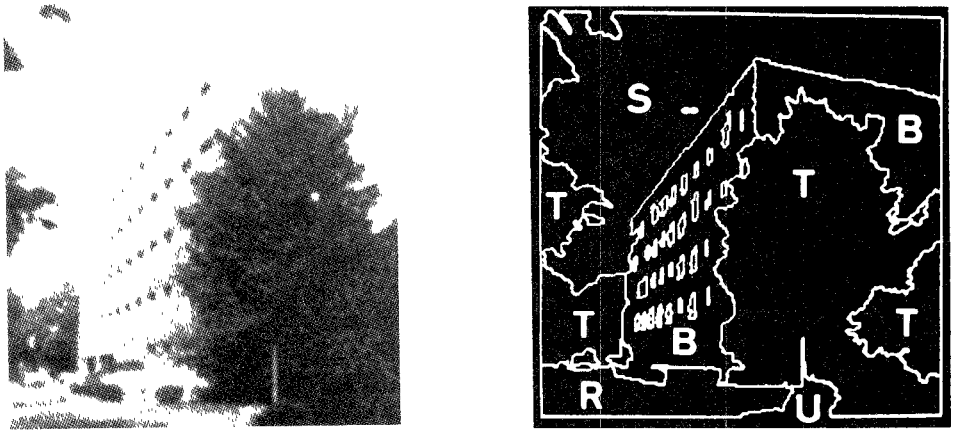
represented as a semantic net that contains properties of scene entities and their relational constraints. An initial set of labels is obtained by the feature-space approach. A rough interpretation—called a *plan*—is obtained by probabilistic relaxation labeling operating on large patches, tentatively merged with surrounding small patches into homogeneous compact regions. Subsequently, a set of heuristic rules operates both on the preliminary patches and the plan in order to produce a detailed interpretation. The control structure of the rule-based part distinguishes two phases—one for analyzing the overall structure without attending to details and one for analyzing detailed structures. In contrast with the plan generator, which uses an optimization strategy, the assignment of labels by the rule-based system is heuristic and may not be optimal. This disadvantage is partly overcome by feeding decisions back to the relaxation-labeling process to reevaluate and update the plan, thus maintaining the overall plan consistency. A typical result of Ohta's procedure, showing the effectiveness with which various confusing scene components is separated, is shown in Figure 19.

- **Wu, Suetens, and Oosterlinck—Rule-Based System for Chromosome Classification.** Wu et al. [1987] propose a rule-based system to classify chromosomes in metaphase images. The result of a feature-space analysis [Groen et al. 1989] is used as a hypothesis. Hypotheses are verified and modified by constraints imposed by the context and represented as if-then rules. The gross hierarchical strategy consists of a group classification, followed by a more specific type classification. The performance of the rule-based system is a clear improvement over conventional techniques based on the feature-space approach alone. This work has further been extended by using belief functions and evidential reasoning in order to achieve constraint satisfaction by probabilistic relaxation [Wu et al. 1989], a strategy that belongs to those described in Section 3.1.
- **Nagao and Matsuyama—Rule-Based Resegmentation.** Nagao and Matsuyama [1980] use a rule-based system to recognize various objects such as crop fields, forests, roads, rivers, cars, and buildings in color aerial images. The image is initially partitioned into regions based on the feature-space approach. This initial classification uses strict conditions in order to avoid false recognition. The acceptance thresholds are relaxed when additional contextual evidence is perceived in the environment. This strategy has the advantage that initial decisions using the feature space are reliable and need not be revised afterward. Regions giving strong photometric evidence for a particular object are classified first; they constitute a context for other regions with weaker photometric properties.

  An important property of this system is its ability to correct for some segmentation errors. Rules exist that activate a split-or-merge algorithm on irregularly shaped regions. Although this *resegmentation* is simple and operates on the contours of the regions instead of on the raw image data, it contains the beginnings of some of the methods discussed in Section 4.

In the previous systems, rules are grouped into larger knowledge blocks, called *phases* or *classes*, which are initiated by metarules and applied sequentially as the analysis progresses. The organization of these knowledge blocks defines the overall control structure of the system. Increasing the number of classes, each having a few complex coarse-grained rules, naturally leads to a *blackboard* style of control, which is characterized by a dynamically updated list of goals, tasks, and their subparts to keep track of what the system is trying to do and what it will do next.

- **Draper et al.—The Schema System.**




(a) (b)

**Figure 19.** (a) Image of an outdoor scene; (b) Ohta's [1985] resulting labeled image, with S = sky, T = tree, B = building, R = road, U = Unknown. (Used with permission of Pitman Publishing; © Pittman Publishing, 1985.)

The Schema System [Draper et al. 1988] has adopted much of the structure of the basic blackboard system, with a few significant adjustments. It is demonstrated in 2D images from natural domains (four road scenes and three house scenes). The Schema System partitions the available knowledge about the scene in terms of natural object classes. Each class of objects, object configurations, and object parts has a corresponding schema that stores the object and control knowledge specific to that class. Each schema is an expert at recognizing one type of object. General-purpose programs or tools, which are not object specific, are stored in separate knowledge sources and can be called by the schema strategies. Schema instances run concurrently and exchange information through a global blackboard mechanism.

Draper et al. [1988] serves here as an excellent example of the blackboard form. However, the authors state that the low-level knowledge sources, such as the region segmentation and the line extraction routines, may in principle be reactivated with new parameter values tuned to the specific image content that becomes available during the analysis. The possibility of integrating this knowledge-directed resegmentation [Draper et al. 1988] into the system puts this paper on the borderline of the combined strategies discussed in Section 4. Other knowledge-based systems such as those of McKeown et al. [1985] and Hwang et al. [1986] depend strongly on combining strategies in order to recover missing features in the image and are described in the next section.

#### 3.2.5 When to Use this Strategy

These methods are appropriate for scenes with moderately complex photometry combined with semantic complexity. Therefore, in controlled environments where we can depend upon low-level operators to extract relevant features, the techniques described in this section can be very efficient. We need only run the initial local operator once to get a set of symbols that can then be parsed quickly and reliably using semantic representations of the domain knowledge.

#### 3.2.6 When to Avoid this Strategy

This strategy should be restricted to situations with reliable intermediate states

in order to avoid backtracking that may result in a combinatorially explosive search. Furthermore, in uncontrolled environments, such as natural outdoor scenes, images will usually contain noise and ambiguities that may completely disrupt the reliability of most local operators. Although the semantic context may correct for the inaccuracies introduced by analyzing the local photometry, there is usually little reason to hope that the features found initially can be assembled into reliable object labels without using substantial additional domain knowledge. In the next section, we discuss alternative strategies for achieving this goal.

## 4. COMBINED STRATEGIES

For complex models with complex semantics, direct optimization as described in Section 2 may become computationally impractical. In this case, direct search can be replaced by more complex search strategies that systematically constrain the search space by finding and refining partial matches. This approach allows the feature extraction process to continuously refer to the image data and to be dynamically dependent upon the current context of the parse. The parsing process typically includes different methods for recovering missing features, such as template matching, gradient descent, and assorted low-level operators. We now consider in detail a number of systems exploiting hybrid methods and hierarchical search philosophies.

### 4.1 Refining Matches by Resegmentation

Refining matches by resegmentation effectively recovers missing object features when the image is initially undersegmented or when additional low-level feature data are available for use in later stages of the analysis. This technique exploits the fact that object features giving weak supportive evidence may be discovered by semantically associating them with strong (often sparse) features.

- **McKeown et al.—SPAM: Region Enlargement, Extension, Join/Merge, and Recovering Missing Regions by Low-Level Resegmentation.** The SPAM system [McKeown et al. 1985] is an example of a method that continually refers back to the original image data to refine its hypotheses of how the image should be divided into recognizable objects. SPAM's specific application is the interpretation of airport scenes using maps and domain-specific knowledge.

  SPAM begins with an image of the scene and a trial segmentation of the scene into regions. The basic premise of the system is that these initial regions are adequate building blocks from which to begin a rule-based analysis. Next, domain knowledge and image data are combined to *resegment* the image (i.e., make a new version of the segmentation regions). The processes that can be invoked by the context rules include *region enlargement* (add area to a region), *region extension* (grow in a particular direction), *join/merge* (coalesce multiple regions into a single one), and *recovering missing regions*. For example, given several linear image regions that are collinear with one another, a new linear region is hypothesized that encompasses each original region. The rules attempt to verify this hypothesis by invoking a linear feature extraction module using the new (hypothesized) linear region as a guide. If, for example, a terminal function area contains roads and parking lots, but no parking aprons, SPAM's rules invoke image analysis tools that look for regions whose shape and texture properties match the model of SPAM for parking aprons. Substantial additional progress on the process of acquiring and efficiently expressing the domain knowledge needed for systems of this type is described in McKeown et al. [1989].
- **Hwang, Davis, and Matsuyama—Recovering Missing Object Parts by Resegmentation Using a Different**

**Threshold or a Different Low-Level Operator.** In Hwang et al. [1986], an image-understanding framework is proposed, and its performance is demonstrated on a high-resolution aerial image of a suburban housing development in which houses, roads, and driveways are located.

The system creates an *initial context* by finding bright, compact, rectangular blobs (house hypotheses) and bright, elongated ribbons (road hypotheses). Using this context and the topological model knowledge, hypotheses and composite hypotheses are iteratively generated and verified. Missing parts (houses, road pieces, driveways) are searched for in the image by using a *different low-level segmentation operator* and/or *different threshold values* to obtain the necessary evidence.

- **Wang and Srihari—Repairing Oversegmentation and Undersegmentation by Rethresholding/Resegmentation.** Wang and Srihari [1988] find destination address blocks on mail pieces using a blackboard framework. Mail pieces include machine-written and handwritten letters, magazines, newspapers, and irregularly shaped parcels.

  The computational solution includes provisions for rethresholding and resegmenting a portion of an image using different parameters if the initial segmentation is found to oversegment or undersegment the object. For example, if the system examines the result of machine-generated text segmentation on hand-generated address data, it may find a cluster of neighboring small blocks or a block whose size is within the acceptable range for a hand-generated address but too large for a machine-generated address. In these cases, the system will invoke the hand-generated text segmentation tool on that area.

- **Nazif and Levine—Expert Segmentation System.** The goal of the rule-based system by Nazif and Levine [1984] is to outline structures that satisfy some basic grouping principles, such as similarity, proximity, uniform density, good continuity, and closure. The system starts from edges and homogeneous regions, extracted from the image using standard segmentation routines. Next, a collection of heuristic perceptual grouping rules is applied. A region can be split along an intersecting edge or be based on the histogram of a feature. Regions are merged or deleted based on continuity and good closure. A line, for example, may be extended by expanding the end point along the maximum local gradient. Lines can be joined if their end points are close together and/or if the lines are collinear.

### 4.2 Refining Matches by Template Matching

When the missing features have a precisely known shape or photometry but cannot be found by low-level feature extraction due to noise or occlusions in the image data, template-matching techniques may help solve the problem.

- **Shneier et al.—Model-Driven Feature Extraction.** The goal of the system of Shneier et al. [1986] is to maintain a description of a workspace that consists of moving industrial parts and fixed surfaces such as buffer tables and machine tools. The process computes how the workspace will appear at the next cycle of sensing and how it will be perceived by each individual sensor; then, it predicts the images—usually small regions—of features such as corners. Model-driven feature extraction is performed by processing the image in the windows where features are expected and by tailoring the feature detectors to the expectations. For example, if a corner is expected with a particular angle, lower thresholds can be set (resegmentation) to find it. If a corner is detected, the result of the matching process is a new feature with ideal properties, that is, the corner's angle is derived from the model

and the position and orientation are derived from the data. Detected features are combined into objects and objects into assemblies. This result is sent back to the predictive process for use in the next iteration.

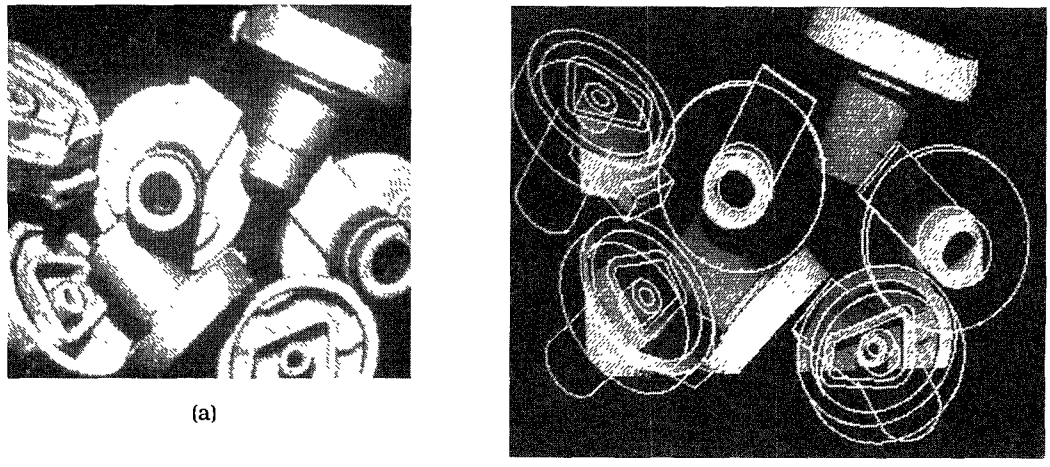
- **Bolles and Horaud—3DPO: Model-Driven Correlation-Based Hypothesis Verification.** The 3DPO system [Bolles and Horaud 1986] recognizes and locates 3D overlapping industrial parts jumbled together in a bin, as shown in Figure 20a. When the 3DPO system believes it has found the pose of an object, it verifies this hypothesis and refines the pose estimate by back-projecting the prediction onto the range data, as in Figure 20b. Figure 20 illustrates the complexity of the 3DPO problem domain and shows the system's ability to function with incomplete information. The approach resembles that of the SCERPO Vision System [Lowe 1987]. But whereas SCERPO backprojects the industrial part onto the segmented edge data, 3DPO compares the predicted data with the *original range data* based on *correlation*. This template-matching approach to hypothesis verification is restricted to rigid objects and requires a detailed model of the physics of the data acquisition, which is typically more straightforward for range data than for intensity images. More details are given in Appendix C.

### 4.3 Refining Matches by Flexible Model Matching

When complex models cannot be defined in terms of rigid shapes but must instead be specified by a set of generic constraints, template matching must be replaced by flexible model matching. Typically, selected model cues are used to initiate the search for the presence of missing model components and avoid combinatorial explosion.

- **Levy-Mandel, Venetsanopoulos, and Tsotsos—Model-Driven Line Tracking.** The rule-based system of Levy-Mandel et al. [1986] automatically localizes characteristic points (landmarks) on x-rays of the human skull. The system contains a heuristic line tracker that starts from an anatomical seed that provides a context to constrain the search. The strategy is hierarchical: The most important lines are tracked first, and the location of detected lines defines the appropriate location of the seed of the subsequent line.
- **Suetens et al.—Recovering Flexible Object Parts in the Image data.** The approach of Suetens et al. [1989] focuses on the recognition of the coronary blood vessels in single and in stereoscopic angiograms. At each level of the model hierarchy, an optimization procedure is started to find missing object attributes. Blood vessel segments are found in the image by propagating two wave fronts, starting from lines of maximum local intensity, until sharp edges are encountered. The method is similar to that used by Tenenbaum et al. [1979] to monitor the water level of a reservoir using aerial images. These segments form a set of largely disconnected blood vessels. To create a connected tree structure, the edges of each disconnected segment are extrapolated in the direction of that segment by means of the dynamic programming technique of Gerbrands et al. [1986]. To find long missing segments, the direction of the search is continuously updated using the history of the path, similar to the procedure of Quam [1978]. Finally, missing or spurious patterns in the image are detected by exploiting the anatomical model knowledge described in textbooks and acquired from cardiologists. Again, missing segments are recovered using a combination of dynamic programming [Gerbrands et al. 1986] and heuristic tracking [Quam 1978].

If stereoscopic images are available, they can be used at any time to improve the existing interpretation. We have seen this characteristic before in the



(a) (b)

**Figure 20.** (a) Optical image of a bin of industrial parts; (b) 3DPO parse of part bin overlaid on the range image. (Courtesy of R. Bolles.)

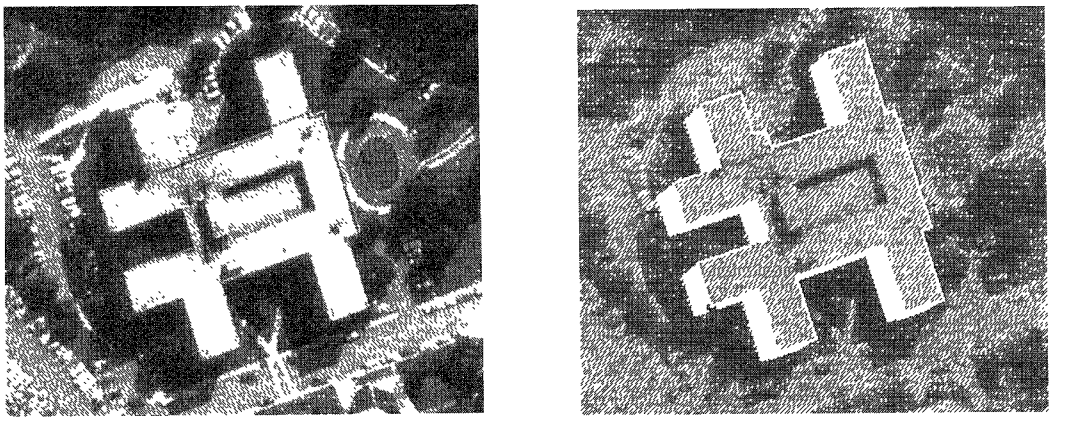
3D MOSAIC system [Herman and Kanade 1986]. In Suetens et al. [1989], it is shown how missing segments in one image are recovered using dynamic programming and the property that contiguous segments in one image necessarily correspond to contiguous segments in the other image.

- **Fua and Hanson—MDL: Finding Complete Generic Objects Using Model-Driven Optimization.** The approach of Fua and Hanson [1987, 1988, 1991] describes generic objects in terms of a language that specifies both photometric and geometric constraints on the objects and their appearance in the image. Figure 21 illustrates the ability of the generic model approach to generate a complex building model instance in 3D spontaneously.

  Buildings in aerial images are modeled as rectilinear structures whose internal gray level intensities are planar, whereas roads are modeled by pairs of parallel, smoothly curved edges enclosing a planar intensity area. To generate optimal descriptions, a hierarchy of increasingly complex models is fitted to the photometric data. These models range from elementary edges with the appropriate geometry to contours that enclose areas with specific photometric and geometric properties. This technique frequently produces sets of plausible but conflicting possible parses and therefore includes a mechanism based on the MDL criterion [Leclerc 1989] to choose the most likely scene labels. More details about this work are given in Appendix C.

- **Pentland—Recognizing a Generic Part Structure Using Optimization.** Pentland [1990] uses a general-purpose "parts" representation to recognize natural 3D objects in range images. Objects are described in terms of shapes of the component parts, which are modeled as deformable superquadrics. The system is illustrated on three range images, one of a ThingWorld, one of a goose, and one of a rabbit and book.

  A binary image is first obtained by automatic thresholding of texture, intensity, or range data, whichever is available. A set of 2D binary patterns, whose shapes are 2D projections of 3D superquadrics, is then fit to the binary image by template matching. The detected parts are considered as hypotheses, and the MDL criterion [Leclerc 1989] is used next to select the



(a) (b)

**Figure 21.** (a) Original image with a complex building; (b) highest scoring 3D roof hypothesis produced by the system, with projected walls.

subset of part hypotheses that best describes the binary image data. Given the segmentation into 2D patterns, the corresponding 3D parts of similar width, length, and orientation are subsequently deformed in order to minimize the error between the visible surface of the 3D object and the available range measurements.

- **McKeown and Denlinger—ARF: Road Tracking Driven by Multiple Road Models.** The automatic road follower (ARF) by McKeown and Denlinger [1988] invokes two different road trackers independently. The first road tracker is Quam's correlation-based technique [Quam 1978] with several improvements. The second is an edge tracker. Normally they generate the same center line; but if one of the road followers fails, the system is able to switch from one road tracker to the other. The authors state that "the combined tracker is better than either tracker alone in a significant number of cases."

  As compared to Quam's road tracker, ARF uses a second road tracker, which exploits additional knowledge (e.g., additional typical road characteristics) in order to improve the performance. The principle of using multiple complementary information sources (cf. 3D MOSAIC [Herman and Kanade 1986]) partly corrects for inaccuracies introduced by the highly heuristic nature of the search and the heuristic definition of the features used to recognize roads.

### 4.4 When to Avoid this Strategy

A combined strategy provides additional power but should be reserved for problems that cannot be solved in a simpler way. Because of the potential need for elaborate models, complex control strategies, and time-consuming computation in this method, every effort should be made to transform a particular application into a simpler domain. In other words, we should make a conscientious effort to understand each particular application and find the least complex way to solve it before resorting to the level of complexity required to apply combined strategies.

### 4.5 When to Use this Strategy

Use this strategy when all else fails.

## SUMMARY

When attempting to solve a complex object recognition problem, it is difficult to choose an appropriate strategy from

the wealth of available techniques. In this paper, we provided a basic introduction to this problem and proposed an organizational framework that gives some insight into making these difficult choices and understanding the tradeoffs involved.

We classified the strategies according to their suitability for complex models and for complex data. Using our classification framework, we delimited the domains of the various techniques and illustrated their characteristics with selected examples from the literature.

APPENDIX A. INDEX OF LITERATURE REVIEWED

| **Fitting Models to Photometry** | |
|---|---|
| Method and Author | Summary |
| **Rigid Model Fitting** | |
| Image Statistics | |
| Rosenfeld 1969; Hall 1979 | Image subtraction and correlation |
| Ballard and Brown 1982 | |
| Reynolds et al. 1989 | |
| Wallace 1988 | Template matching on segmented images |
| Mansouri et al. 1987 | |
| Hough Transform | |
| Ballard and Brown 1982 | Hough transform |
| Ballard 1981 | |
| Illingworth and Kittler 1988 | |
| Niblack and Petkovic 1988 | |
| **Flexible Model Fitting** | |
| Dynamic Programming | |
| Fischler et al. 1981 | F*: Iterative path finding |
| Gerbrands et al. 1986 | Resampling the search region |
| Nuyts et al. 1989 | Parametric search region |
| Tenenbaum et al. 1979 | Optimal path without shape constraint |
| Yamada et al. 1988 | Noniterative procedure without resampling |
| Maitre and Wu 1987 | Matching segmented images with line drawings |
| Gradient Descent | |
| Kass et al. 1987 | Snakes: Deforming a flexible curve |
| Fua and Leclerc 1990 | |
| Terzopoulos et al. 1988 | |
| Fua 1989; Witkin et al. 1987a | |
| Gardin and Meltzer 1988 | |
| Witkin et al. 1987b | |
| Closed-Form Solution | |
| Premoli et al. 1989 | KAMRI: Closed-form solution |
| Relaxation | |
| Murray and Buxton 1987 | Region segmentation using relaxation |
| Heuristic Pruning | |
| Quam 1978 | Heuristic road tracker |
| Zhang and Simaan 1987 | Model-driven seed growing |

| **Fitting Models to Symbolic Structures** | |
|---|---|
| Method and Author | Summary |
| **Graph Matching** | |
| Search | |
| Murray 1987 | Depth-first recursive search |
| Bolles and Horaud 1986 | 3DPO: Maximum clique finding (see also combined strategies) |
| Mulgaonkar et al. 1984 | Matching nonidentical graphs |
| Horaud and Skordas 1989 | Ranking maximal cliques |

| | |
|---|---|
| Dynamic Programming | |
| Eshera and Fu 1986 | Dynamic programming |
| Fischler and Elschlager 1973 | Heuristic dynamic programming |
| Relaxation Labeling | |
| Huffman 1971 | Line drawing interpretation |
| Clowes 1971 | |
| Malik 1987 | |
| Tenenbaum and Barrow 1977 | MSYS: Discrete relaxation labeling |
| Mohan and Nevatia 1988 | Constraint satisfaction network |
| Mohan and Nevatia 1989 | |
| Heuristic Pruning | |
| Amini et al. 1989 | Heuristic best-first search |
| Ayache and Faugeras 1986 | HYPER: Heuristic tree pruning |

Composite (Hierarchical) Model Fitting

| | |
|---|---|
| Structural Grouping | |
| Lowe 1987 | SCERPO: Locating perceptual structures |
| Brooks 1981 | ACRONYM: Invariant observables |
| Huertas et al. 1987 | Finding linear structures |
| Huertas and Nevatia 1988 | |
| Jain and Hoffman 1988 | Merging adjacent surface patches |
| Refining Matches Using Multiple Information Sources | |
| Herman and Kanade 1986 | 3D MOSAIC: Object completion using additional images |
| Fan et al. 1988, 1989 | Match image interpretations via heuristic search |
| Bobick and Bolles 1989 | Integration of visual information over time |
| Knowledge Selection by Rules | |
| Ohta 1985 | Rule-based system for outdoor scene analysis |
| Wu et al. 1987 | Rule-based system for chromosome classification |
| Nagao and Matsuyama 1980 | Rule-based resegmentation |
| Draper et al. 1988 | Schema System |

**Combined Strategies**

| Method and Author | Summary |
|---|---|
| Refining Matches by Resegmentation | |
| McKeown et al. 1985 | SPAM: Region recovery, enlargement, extension, join/merge by resegmentation |
| Hwang et al 1986 | Part recovery by rethresholding/ resegmentation |
| Wang and Srihari 1988 | ABLS: Repairing over and under-segmentation by rethresholding/ resegmentation |
| Nazif and Levine 1984 | Expert segmentation system |
| Refining Matches by Template Matching | |
| Shneier et al. 1986 | NBS: Model-driven feature extraction |
| Bolles and Horaud 1986 | 3DPO: Model-driven correlation-based hypothesis verification |
| Refining Matches by Flexible Model Matching | |
| Levy-Mandel et al. 1986 | Model-driven line tracking |
| Suetens et al. 1989 | Recovering flexible object parts in the image data |
| Fua and Hanson 1991 | MDL: Finding complete generic objects using model-driven optimization |
| Pentland 1990 | Recognizing a Generic Part Structure Using Optimization |
| McKeown and Denlinger 1988 | ARF: Road tracking driven by multiple road models |

## APPENDIX B. RELATED REVIEW PAPERS AND BOOKS

### B.1 Related Review Papers

The literature contains diverse survey papers covering different aspects of computational vision. The distinctive feature of the present article is that we systematically categorize a wider variety of computational strategies than previous articles, as well as analyzing their appropriateness to particular applications. Since there have been a number of other review papers, some of whose contents are similar to ours, it may be useful for us to summarize the salient features of these articles and to contrast our approach.

#### *B.1.1 Nagao: Control Strategies*

Nagao [1984] uses his own research results to illustrate merits and weaknesses of a number of computational strategies. The main strategies he discusses are the feature-space approach and hierarchical parsing. He recognizes the need for dynamically exploiting the image data. He says, however, that because a process that includes low-level image processing in the scope of control is complicated, most of the control structures are restricted to the symbolic level, and only a few have feedback to the image level. The paper by Nagao was written in 1982; in the meantime, the idea of feedback had been further elaborated and had given rise to additional computational strategies, which are discussed in our paper. Nagao further emphasizes the importance of declarative programming—which we do not consider as a computational strategy but rather as a programming methodology—and of the need for powerful software tools for the development of sophisticated control structures.

#### *B.1.2 Rao and Jain: Knowledge Representation and Control*

Rao and Jain [1988] discuss the pros and cons of different knowledge representation formalisms and different control strategies used in computational vision. Using their classification, they review some well-known systems such as Acronym and Visions. The discussion, however, is restricted to strategies for perfect data. The authors account for this limitation by pointing out that the number of papers that describe some form of top-down feedback referring to the image data is small. They state that "all vision systems use some form of feedback, in fact, but people have not made an effort to isolate and focus on this particular aspect." In this paper, we emphasize the role of this feedback in object recognition and clearly distinguish strategies dealing with interpretation of labels from other strategies that repeatedly exploit the image data at the pixel level to find model instances.

#### *B.1.3 Kanade: The Segmentation Problem*

Kanade [1980] gives a unified view of what he calls "the problem of segmentation." "Often," according to the author, "the ultimate goal of image analysis is to obtain a segmentation which separates out semantically meaningful objects or parts of objects." To discuss the problem of region segmentation, Kanade provides the following problem-solving paradigm: "Given an image, cues (picture domain cues or scene domain cues) are extracted, which are then used to access the generic model of the task world to generate hypotheses, which are verified by projecting them back to the picture level and by matching them with the input image." The picture domain cues are the features observed in the image, such as line segments, homogeneous regions, and intensity gradients. The scene domain cues are the features that give rise to the picture domain cues, such as edge configurations, surface orientations, and reflectance. Rather than dealing with computational strategies, this paper discusses the role of different information sources (signal, physical, and semantic) to obtain a "semantic region segmentation;" that is, to assign semantic labels

to pixels that best satisfy both the local image feature properties and the semantic constraints. In particular, the importance of exploiting the physical level of knowledge, "the bridge between a picture and a scene," is emphasized.

#### B.1.4 Pavlidis: Progress in Image Analysis

Pavlidis [1986] surveys major trends in the literature and identifies the reasons that have hindered progress. He focuses on typical works rather than striving for completeness. The author distinguishes image analysis (the topic of his paper) from pattern recognition or image understanding, the latter implying the assignment of name labels or descriptions by matching the results of image analysis to world models. Image understanding, such as line-drawing interpretation work (Huffman-Clowes, etc.) and symbolic reasoning systems (Acronym, etc.) are not covered by Pavlidis' paper. Our paper, on the other hand, does review such approaches since they encompass important computational strategies.

#### B.1.5 Mantas: Methodologies in Pattern Recognition

Mantas [1987] presents an overview of existing "methodologies in pattern recognition and image analysis." Like Pavlidis, he makes a distinction between image analysis, "the description of image features into a parsable string of numbers or characters," and pattern recognition, "the classification or parsing process of the created patterns." Segmentation and image-to-image matching methodologies are classified under image analysis. Statistical, syntactic, and hybrid classification methods are the author's main categories of pattern recognition methodologies. Unlike our paper, this paper does not review heuristic approaches to structural pattern recognition, nor does it discuss methodologies that integrate image analysis and pattern recognition.

#### B.1.6 Nandhakumar and Aggarwal: Contrast of Conventional and AI Approaches

Nandhakumar and Aggarwal [1985] review the approaches to pattern recognition, that is, the "automated extraction of information from signals," using a categorization into conventional techniques and the AI-based approach. The conventional approach combines the statistical methodology (template matching and feature space) with the structural methodology (syntactic pattern recognition and relaxation labeling). The AI-based approach emulates the hypothesize-and-test paradigm and uses heuristics to reduce the search space. Whereas the conventional techniques involve large amounts of numerical computation and analytically well-formed models, the AI approach is characterized by symbolic reasoning, which implies the importance of a suitable knowledge representation and control structure, and focuses on the efficient use of different knowledge sources in various forms. The paper by Nandhakumar and Aggarwal mainly emphasizes the distinctions between these two broad categories and does not discuss their domains of applicability, which is one of the principal goals of our treatment.

#### B.1.7 Binford: Model-Based Vision

Binford [1982] gives a good survey and critique of the state of the art (up to 1982) of what he calls "model-based image analysis systems." Model-based image analysis is to be considered as that part of computational vision that involves the use of high-level models, such as aircrafts, buildings, and ribs in chest x-rays. The author nicely summarizes the limitations of the systems that existed at that time. One major limitation, he says, is the poor performance of "segmentation." For example, few systems used shape information in segmenting regions. A second shortcoming is the weak definition and use of models. Models are usually image models and are viewpoint dependent (except for Acronym). Consequently, the ability to relate three-space models (world knowledge) to image structures is lacking. Furthermore, the models described are of specific objects, so hypothesis generation is limited to restricted scene domains.

Generally, this paper focuses on the modeling issue rather than on computational strategies. Since the publication of the paper, the field of computational vision has made progress toward overcoming each of the limitations mentioned above. In particular, recent developments have led us to introduce new classes of computational strategies not discussed by Binford.

#### *B.1.8 Rosenfeld: Scene Descriptions from Image Analysis*

The paper by Rosenfeld [1987] reviews the basic stages of an image analysis process, that is, the "construction of scene descriptions on the basis of information extracted from images or image sequences." The specific areas covered are feature extraction, texture analysis, surface orientation estimation, image matching, range estimation, segmentation, object representation, and model matching. For each of these areas, Rosenfeld summarizes the state of the art, then presents limitations and future directions. With respect to computational strategies, the author laments the lack of a general theory of control in image analysis and points out the need for incorporating geometric constraints and object semantics into the segmentation process. Our review attempts to alleviate both shortcomings: First, we categorize and discuss the computational strategies found in the literature. Second, we introduce additional categories of strategies that instantiate geometric and semantic constraints directly in the image.

#### *B.1.9 Wallace: Computational Strategies for Object Recognition in Line-Segmented Images*

The following computational strategies, applicable to segmented images, are discussed by Wallace [1988]: boundary correlation, generalized Hough transform, relational distance measures, graph matching, heuristic search, and relaxation labeling. To illustrate and compare these strategies, the problem of identifying a line-segmented image of a metallic bracket is used throughout the text. Using a single application has the advantage that a quantitative assessment of each of the strategies can be made. On the other hand, it narrows the study of available strategies in the literature to those that are useful for that particular application. For example, the class of computational strategies that supports flexible models (see Section 2.2) is not discussed. This derives from the author's particular view of the computational vision process: "Image interpretation may be considered as a three-stage sequential process, consisting of primitive extraction, grouping of primitives into extended features, and matching of scene descriptions to preformed models."

#### *B.1.10 Matsuyama: Categorization of Expert Systems for Image Analysis*

Matsuyama [1989] classifies expert systems for image analysis into four categories: (1) consultation systems for image processing for users with little experience, (2) knowledge-based program composition systems that build complex programs from abstract programs specified by the user, (3) rule-based design systems for image segmentation representing the various heuristics common to a segmentation method explicitly [Nazif and Levine 1984], and (4) goal-directed image segmentation systems that automatically extract image features, such as a rectangle with a specified area, by using knowledge about the image processing operators and the way to combine them. Except for the third category, the discussion is restricted to expert systems for image analysis that use only control knowledge about how to use image processing operators, as in program libraries. In this paper they are called expert systems for image processing (ESIP). They are different from the expert systems described in Section 3.2 and typically use a large amount of domain-specific knowledge of the scene and its objects. The knowledge in ESIPs is described explicitly and declaratively. The author states that most ESIPs were

developed to examine their feasibility and cites only one commercial ESIP. ESIPs do not solve basic computer vision problems but should rather be considered as a new programming style and as a step toward new flexible software environments for developing image analysis programs.

### B.2 Related Books

This paper provides a guide to the literature dealing with applied object recognition. Object recognition is a topic under intensive study in different research fields, such as image processing and artificial intelligence. Some familiarity with each of these broader domains can be acquired from the following textbooks:

## REFERENCES


1. D. H. BALLARD, C. M. BROWN, *Computer Vision*, Prentice-Hall, Englewood Cliffs, N.J., 1982.
2. A. BARR, P. R. COHEN, E. A. FEIGENBAUM, *The Handbook of Artificial Intelligence*, vol. 1-3, William Kaufmann, Los Altos, Calif., 1981–1982; A. Barr and P. R. Cohen, *The Handbook of Artificial Intelligence*, vol. 4, Addison-Wesley, Reading, MA, 1989; A. Barr and P. K. Cohen, *The Handbook of Artificial Intelligence*, vol. 4, Addison-Wesley, MA, 1989
3. B G. BATCHELOR, D. A. HILL, D. C. HODGSON, *Automated Visual Inspection*, IFS (publications) Ltd, Bedford, and North Holland, Amsterdam, 1985.
4. K. R. CASTLEMAN, *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1979.
5. R. O. DUDA, P E. HART, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973
6. M. A. FISCHLER, O. FIRSCHEIN, *Intelligence: The Eye, the Brain and the Computer*, Addison-Wesley, Menlo Park, Calif., 1987.
7. R. C. GONZALES, P. WINTZ, *Digital Image Processing*, Addison-Wesley, Reading, Mass., 1977.
8. E. L. HALL, *Computer Image Processing and Recognition*, Academic Press, London, 1979.
9. B. HORN, *Robot Vision*, MIT Press, Cambridge, Mass., 1986
10. D. MARR, *Vision*, W. H. Freeman and Company, San Francisco, Calif., 1982.
11. R. NEVATIA, *Machine Perception*, Prentice-Hall, Englewood Cliffs, N.J , 1982.
12. W. K. PRATT, *Digital Image Processing*, John Wiley & Sons, New York, 1978.
13. J. A. RICHARDS, *Remote Sensing Digital Image Analysis, An Introduction*, Springer-Verlag, Berlin, 1986.
14. A ROSENFELD, A. C. KAK, *Digital Picture Processing*, vols. 1-2, Academic Press, New York, 1982.
15. S C. SHAPIRO, ED., *Encyclopedia of Artificial Intelligence*, vols. 1-2, John Wiley & Sons, New York, 1987.
16. Y. SHIRAI, *Three-Dimensional Computer Vision*, Springer-Verlag, Berlin, 1987.
17. Y. SHIRAI, J. -I. TSUJII, *Artificial Intelligence, Concepts, Techniques and Applications*, John Wiley & Sons, New York, 1984.
18. S T. TANIMOTO, *The Elements of Artificial Intelligence Using Common Lisp*, W. H. Freeman, New York, 1990.
19. J T. TOU, R. C. GONZALEZ, *Pattern Recognition Principles*, Addison-Wesley, Reading, Mass., 1974.
20. P. H. WINSTON, *Artificial Intelligence*, Addison-Wesley, Reading, Mass., 1984, 2nd ed


## APPENDIX C. DETAILED DISCUSSION OF SELECTED KEY PAPERS

### C.1 Ballard — GHough: The Generalized Hough Transform

The Hough transform [Hough 1962; Rosenfeld 1969] was originally developed as a statistically reliable technique for finding parameters of straight lines in data consisting of collections of points. The basic concept is to take an equation described by a certain number of parameters, for example, the straight line

$$x \cos \theta + y \sin \theta = c$$

parameterized by $(\theta, c)$ and plot the values of $(\theta, c)$ for all the lines passing through a particular data point $(x_1, y_1)$. For each additional data point $(x_i, y_i)$, we plot the corresponding curve in $(\theta, c)$ space. An example of such a plot is shown in Figure 22. The point where the density of intersecting curves is the highest is the best candidate for the values of the straight line parameters describing the data. In practice, the $(\theta, c)$ space is quantized as an array in computer memory, and the memory cells are treated as counters that are incremented whenever the $(\theta, c)$ curves pass through the particular cell. This array is called the *Hough accumulator*, and it is clear that its cells contain, in effect, the number of votes cast by the data points for each sampled value of the line parameters $(\theta, c)$.

The concept of the Hough transform can be extended in several directions. The simplest is to treat other algebraic curves with a manageable number of parameters; circles and ellipses are typical examples. The idea is always to map the data into a weighted space of parameters describing the shape to which the data are expected to conform. The Hough transform takes input in a data space and produces output in an abstract parameter space.

There is a conceptual problem, however, when we want to locate a shape that has no straightforward parameterization as a space curve. How do we formulate the accumulator array for a shape that is a complex collection of points rather than a straight line or a circle? This problem was first solved by Ballard [1981], who noted, in effect, that the relevant parameter space corresponds to the space of spatial transformations that can be made on an arbitrary rigid curve. For an arbitrary curve what is relevant for the object recognition problem is the *location*, *rotation angle*, and *scale* of the curve in the image relative to the object's defining shape template. If an object has orientation-determining and/or fixed-scale features, these properties can be exploited to restrict the possible angles and scales of the template candidates, thus decreasing the dimension of the required Hough accumulator. For 3D objects, we can in principle extend this concept to the space of 3D transformations of a rigid object [Ballard and Sabbah 1981]; in practice, this may be difficult and is less likely to be successful than the 2D method.

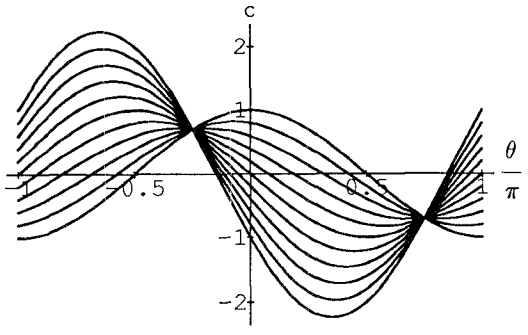



**Figure 22.** A plot of the allowed values of $c$ and $\theta$ in the equation $x \cos \theta + y \sin \theta = c$. Each curve corresponds to a different data point $(x, y)$; we may deduce that the set of data points considered lies on the straight line $y = x - 1$.

#### C.1.1 Implementation

For objects that appear in image data as 2D rigid curves, Ballard's generalized Hough transform (often abbreviated as GHough) is formulated as follows ([Ballard 1981], or Ballard and Brown [1982] pp. 123–131: beware of possible misprints):

- **Digest the shape template.** Define a shape template as a discrete set of points lying on the desired shape, choose a reference point as the template center, and record the angle and distance of the reference point relative to the points chosen on the shape outline. Finally, group these into bins with the *same gradient direction*, determined, for example, by measuring the normal to the template curve at each sample point. This is called the R-table.
- **Define the range of the parameter space.** We may know some restrictions on the location, orientation, and scale of the expected objects in the image relative to the template. If so, define a Hough accumulator array that has the appropriate parameter ranges and quantization steps. For straightforward implementations of GHough, the smaller this array, the better off you are.
- **Process the image data.** Run an edge operator such as the Sobel derivative over the image, producing both an edge strength and a direction at each pixel. Typically, some sort of threshold is applied to select a strong set of edges.
- **Compute the index into the Hough accumulator.** Loop through the selected image edges $(x_i, y_i)$, noting the direction of $\theta_i$ of each edge. When this direction falls in the same direction as an edge in the template

description, look in the R-table for the possible relative locations $(r, \alpha)$ in polar coordinates of the reference point as seen from the image edge. Compute the predicted template reference point using the basic formula

$$x_c = x_i + sr\cos(\alpha + \phi)$$
$$y_c = y_i + sr\sin(\alpha + \phi),$$

where $(s, \phi)$ are the discrete values of scale and orientation, respectively, being considered. If a range of these parameters is being considered, compute separate values of $(x_c, y_c)$ for each value of scale and orientation.

- **Increment the Hough accumulator.** For each image edge, we now have the coordinates $(x_c, y_c)$ and possibly $(s, \phi)$ of a cell in the Hough accumulator array. Increment this cell by one count. Note that several template edges might have the same gradient angle, so one data point might cause several increments to the Hough accumulator, one for each distinct value of $(r, \alpha)$ in the R-table bin corresponding to that gradient direction.

If the Hough accumulator has indistinct peaks, we may achieve better results either by changing the parameter quantization or by performing local averaging in the accumulator to produce broader but higher peaks enabling a clearer choice of preferred parameter values.

This method works well for particular applications such as the location of uniquely shaped lakes or similar landmarks in high-altitude aerial imagery, where the object is essentially two dimensional and rigid. Because the process of voting into the Hough accumulators is statistical, the results continue to be reliable even in the presence of noise, partial data, and occlusions that can disrupt techniques that use semantically driven matching techniques. Thus GHough is appropriate for simple models and complex data situations.

A careful study of the limits of applicability of the Hough transform has recently been presented by Grimson and Huttenlocher [1990], to which we refer the reader for additional evaluation information.

### C.2 Kass, Witkin, and Terzopoulos — Snakes: Active Contour Models

Snakes [Kass et al. 1987] are deformable curves that can be used to delineate salient image contour edges, lines, and subjective contours. These curves are implemented as splines that deform themselves under the influence of *image constraints* designed to attract them toward features of interest and of *internal continuity constraints* that force them to remain smooth, except at a selected number of discontinuity points. Both of these constraints are represented as additive energy fields; the best compromise between them is achieved by deforming the curve so as to minimize its total energy.

#### *C.2.1 Constraint Formulation*

The *image constraint field* $E_e$ used in the snake approach is a weighted sum of three terms:

- A term proportional to the image intensity that attracts the snake toward either black or white lines, depending on the sign of the weight.
- A term proportional to the image gradient that attracts the snake toward edges.
- A term proportional to the curvature of lines of constant gray level in a smoothed image that attracts the snake toward edge terminations.

The *internal constraint field* $E_i$ is the sum of two terms: one proportional to the integral of the square of first derivatives, the other proportional to the integral of the second derivatives along the curve.

Given a starting point, the snake is described by its vertices and is viewed as a massless object embedded in a viscous medium and moving under the influence of the image and internal constraint fields. Its optimal position is found by recursively solving the dynamic equa-

tions until it stabilizes. Given the fact that the smoothness term is quadratic and its derivatives are linear, this reduces to solving a linear system of equations at every iteration. This system is in fact sparse and can be solved quickly (the computation time grows linearly with the number of vertices).

#### C.2.2 Implementation

In the implementation of the snake method, the curves are described as polygons with $n$ equidistant vertices $X = \{(x_i, y_i), i = 1, \ldots, n\}$. The total energy $E$ is the weighted sum of $E_i$ derived from the internal constraint field and $E_e$ from the external image constraint field.

The explicit form of $E_i$ is

$$E_i = \mu_1 E_{i1} + \mu_2 E_{i2},$$

$$E_{i1} = \sum_i (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2,$$

$$E_{i2} = \sum_i (2x_i - x_{i-1} - x_{i+1})^2 + (2y_i - y_{i-1} - y_{i+1})^2,$$

where $E_{i1}$ and $E_{i2}$ account for the first and second derivatives along the curve, and $\mu_1$, $\mu_2$ are weights.

$E_e$ is computed by integrating the image field along the curve $\mathscr{C}$. For example, when image gradients are used,

$$E_e = -\frac{1}{|\mathscr{C}|}\int_0^{|\mathscr{C}|} |\nabla \mathscr{I}(\mathbf{f}(s))|\, ds$$

where $\mathscr{I}$ represents the image intensities and $\mathbf{f}(s)$ is a vector mapping arc length $s$ of the curve $C$ to points $(x, y)$ of the image. In fact, $\nabla\mathscr{I}$ can be precomputed, allowing for fast optimization.

To perform the optimization, the curve is imbedded in a viscous medium and the dynamical equation of the resulting system is written as

$$\frac{\partial E}{\partial X} + \alpha\frac{dX}{dt} = 0,$$

where $E = E_e + \lambda E_i$ and $\alpha$ is the viscosity of the medium. Since the internal energy $E_i$ is quadratic, its derivative with respect to X is linear and, therefore,

$$\frac{\partial E_i}{\partial X} = KX,$$

where $K$ is a pentadiagonal matrix. Thus, each iteration of the optimization amounts to solving the linear equation:

$$KX_t + \alpha(X_t - X_{t-1}) = \left.\frac{\partial E_e}{\partial X}\right|_{X_{t-1}}.$$

Because $K$ is pentadiagonal, the solution to this set of equations can be computed efficiently in $O(n)$ time using LU decomposition and backsubstitution. Note that the LU decomposition need be recomputed only when $\alpha$ changes.

#### C.2.3 Properties

Snakes have two key properties that make them especially useful for delineating linear features:

- Geometric constraints are used at the lowest level to guide the search.
- The information is integrated along the entire length of the curve, providing a large support while ignoring the irrelevant information from points not belonging to the actual contour.

The snake method has proven useful for interactive specification of image contours and can be applied to a wide range of problems such as motion tracking and interpretation of seismic data.

### C.3 Ayache and Faugeras — HYPER: Heuristic Pruning

The HYPER system [Ayache and Faugeras 1986] is an example of a robust tree-pruning approach. It identifies and accurately locates touching and overlapping flat industrial parts in an image. Object models and segmented image patterns are described by first-degree polynomial approximations of their contours. The number of model segments is typically less than 100 for effective operation of the system.

#### C.3.1 Issues in Search Heuristics

Search efficiency can be improved if heuristics are used to explore the most promising paths first. For example, depth-first search becomes equivalent to hill climbing if the choices are ordered according to an accurate heuristic measure of the remaining distance. Beam search explores only the best $n$ nodes at each level of the search tree. Best-first search improves the efficiency of a breadth-first search by choosing the best open node, no matter where it is in the partially developed tree. In systems such as HYPER, the heuristic measure is used to *prune* the search tree, incurring the possible risk of excluding the desired solution and arriving at a dead end. The appropriateness of the search heuristic is therefore of critical importance.

#### C.3.2 Implementation

The HYPER system assumes that both the model and the image descriptions are given by a set of linear segments, $M_i = (x_i, y_i, l_i, a_i)$ and $S_j = (x'_j, y'_j, l'_j, a'_j)$, respectively; $x$ and $y$ are the coordinates of the segment midpoint; $l$ is the segment length, and $a$ is the segment orientation relative to the horizontal axis. It further assumes that an image description of an object can be transformed into a model description by a rotation, a scaling, and a translation. This transformation is described by the parameter vector

$$v = (k \cos\theta, k \sin\theta, t_x, t_y)$$

that transforms an arbitrary model point $(x, y)$ into an image point $(x^*, y^*)$ as follows:

$$x^* = t_x + xk\cos\theta - yk\sin\theta,$$
$$y^* = t_y + xk\sin\theta + yk\cos\theta.$$

The method consists of three phases:

- **Initialization.** Given a model description and an image description, an initial hypothesis about the position of the model instance in the image is generated by matching a privileged segment, that is, one of the 10 longest segments of the model, to an image segment. Matching is performed by solving the above equations for the two pairs of end points of the model segment and the image segment, yielding a value for the parameters $k$, $\theta$, $t_x$, and $t_y$ and an initial estimate of the transformation vector $v_0$. Typically, a few hundred hypotheses are generated. By comparing local intrinsic features, the compatibility between the pairs of matched segments is calculated and only the best hypotheses are considered for further evaluation.
- **Heuristic Pruning.** Each generated hypothesis is verified by a heuristic tree search procedure. The rigid model contour is iteratively matched to the image segments by successively adding compatible segments to the available partial contour match. At each iteration $i$, a dissimilarity measure $d_{ij}$ between the active model segment $M_i$ and *every* image segment $S_j$ is calculated. The active model segment $M_i$ is by definition the one closest to $M_{i-1}$. The active segment $M_i$ is first transformed into $M_i^*$ by using the transformation vector $v_{i-1}$; $M_i^*$ is then compared to each of the candidate image segments $S_j$ using both local intrinsic features and positional constraints imposed by the available partial contour instance. The dissimilarity measure therefore includes terms that encode the absolute value $a_{ij}$ of the difference between the orientation of the two segments $M_i^*$ and $S_j$, the euclidian distance $D_{ij}$ between the midpoints of the segments, and the absolute value $l_{ij}$ of the relative difference between their lengths, that is, $l_{ij} = (l_i^* - l_j)/l_j$. The terms $a_{ij}$, $D_{ij}$, and $l_{ij}$ have empirical upper bounds $a_{\max}$, $D_{\max}$, and $l_{\max}$, respectively; $d_{ij}$ is then computed as follows: If one of the terms $a_{ij}$, $D_{ij}$, or $l_{ij}$ is *above* its corresponding upper bound, then $d_{ij} = 1$; otherwise,

$$d_{ij} = \frac{pa_{ij}}{a_{\max}} + \frac{qD_{ij}}{D_{\max}} + \frac{rl_{ij}}{l_{\max}},$$

where $p$, $q$, and $r$ are positive weights whose sum equals 1. The model segment $M_i$ is heuristically matched with its best corresponding image segment $S_j$; that is, the segment whose dissimilarity $d_{ij}$ is minimal subject to the constraint that it be less than 1. A recursive least-squares technique is then used to update the estimate of the transformation vector $v_i$.

- **Terminating the matching process.** For each of the hypotheses a quality measure $Q_i$ at each iteration of $i$ of the search measures the length of the identified model relative to the total model length. The quality is maximal if the model is perfectly identified in the image. It decreases if there are occlusions or other anomalies such as noise, tilted objects, or segmentation errors. At the end of the heuristic search procedure, a final test is done on the hypothesis with the highest quality measure by restarting the whole evaluation procedure with the more accurate value of the transformation vector $v$. This process is repeated until no additional model segments can be matched. The hypothesis is accepted if the quality measure is above a prespecified threshold; otherwise it is rejected.

The above method is robust in the presence of bad lighting conditions, partial occlusions up to 60 percent, and scale variations up to 40 percent. It was successfully tested on a large number of industrial scenes and was implemented on a vision system coupled to a pick-and-place robot to grasp and reposition unoriented and partially overlapping industrial parts.

### C.4 Brooks — ACRONYM: 3D Image Interpretation Guided by Invariant Model Relationships

The ACRONYM system [Brooks 1981, 1983] approaches model-based vision using the following basic concepts:

- **Generic Model Classes.** ACRONYM's models are volumetric, 3D models based on conjunctions of generalized cylinders. At the lowest level, object parts are generalized cones with data structures containing slots called *spine*, *cross section* and *sweeping rule*. A specific object such as a motor is then constructed from a set of these elementary units and their geometric relationships. ACRONYM's modeling philosophy, however, allows not only specific models, it also specifies *generic* classes of objects constructed by replacing numbers in the structure definition by variables obeying *constraints* on their values. Since these constrained variables may describe relationships among subparts, as well as the slots parameterizing generalized cones, a wide variety of spatial relationships characterizing generic model classes may be supported.
- **Generic Scene Constraints.** Besides supporting flexible local models, ACRONYM's constraint system allows the description of the entire scene in terms of relevant constraints. Constraints on the camera coordinate system or constraints imposed by the terrain can limit the values of the coordinate origins of individual objects. Thus the knowledge that the camera was pointing straight down from a range of altitudes constrains the sizes of object model instances on the ground, as well as their relative positions—airplanes, for instance, would then be constrained to lie at the same elevation in the world and to have only the freedom to translate at that elevation and to rotate about the vertical axis.
- **Availability of preprocessed images.** Although ACRONYM is not limited to preprocessed images at the conceptual level, it has in practice required data derived from images by an independent process. This process has no relevant knowledge of the use to which ACRONYM will put the processed data. Lines and simple line structures were found by a universal line finder and grouped into 2D ribbons and ellipses—the basic structures that a 3D generalized cone might pro-

ject to the 2D image. The original image data are not incorporated into the analysis, so errors in the preprocessing persist throughout the procedure; the original implementation of ACRONYM is thus best suited for images that are relatively simple photometrically, although the scene itself may be complex.

- **Geometric reasoning about invariant features and relationships.** ACRONYM's analysis relies heavily on a geometric reasoning system that allows constraints on objects and their components to be translated into predictions about image-invariant relationships. These predictions are then used to drive a hierarchical local matching system that suggests ranges of likely locations for clusters of semantically related ribbons and/or ellipses. When matches consistent with the model constraints are located in this way, the position and orientation of 3D model instances are determined. ACRONYM thus deduces a 3D interpretation of the scene, along with constraints on the camera position.

The core of ACRONYM is the prediction procedure, which is organized into four principal sections, each with many complex subcomponents. The basic concepts involved can be summarized as follows:

- **Constraint Manipulation System.** Constraints are already supported at the level of model data structures by allowing the replacement of numerical values by expressions involving algebraic variables. Constraints then take the form of possibly nonlinear rules relating these variables. The system then handles numeric and algebraic bounds of the form

$$\begin{aligned}&\text{Lower-bound}\,(v_1, v_2, \ldots)\\ &\quad \leq \text{expression}\\ &\quad \leq \text{upper-bound}\,(v_1, v_2, \ldots).\end{aligned}$$

Specific examples might include a constraint such as

$$10 \leq \text{DISK-RADIUS}*\text{DISK-RADIUS}*\pi$$

to indicate a disk of area less than 10 in appropriate units.

- **Prediction Process.** A backward-chaining control program sets up goals and invokes a database of about 280 rules to achieve the goals. Multiple parameters can be passed to and from rules, and modifications to the prediction data structure itself are carried out as side effects. The task of these rules for a specific problem domain (e.g., finding airplanes in an airport) is to gather the coordinate transformations that relate objects to one another and find features such as generalized cone pairs that have a viewpoint-invariant characteristic.
- **Shape Prediction.** The discovery of ribbons corresponding to projections of generalized cylinders into the image is carried out in several steps: Possible contours of visible object parts consistent with the camera position are determined, and their dependence on camera parameters is estimated. Then, geometric relationships within the parts of a generalized cone are found. Finally, a coarse filtering process that uses backprojection of image features to the model is invoked to select acceptable shape matches.
- **Feature Relation Prediction.** At this point, families of single image features have been identified and collected into *prediction nodes*. The next step is to create *prediction arcs* that relate multiple shapes on a single generalized cone, as well as shapes from different generalized cones that characterize composite objects. The arc types include (1) exclusive arcs—relating features that cannot coexist, like opposite ends of a cylinder, (2) collinear arcs—relating collinear features, (3) coincident arcs—when two features must touch, (4) angle arcs—relating two generalized cones (like a pair of vertically viewed airplane wings) that must obey a relative orientation constraint

in the image, (5) approach ratio arcs—relating parts that must divide one another in fixed proportions (like the position of a wing on a fuselage), (6) distance arcs—for object parts that have rigid but noncoincident relationships, and (7) ribbon-contains arcs—when one ribbon in the image must contain another.

Once the preprocessed image data have been analyzed in this way and matches of invariant features have been carried out, the scene can be labeled according to the model instances that have been discovered. An example of ACRONYM's analysis is shown in Figure 17. Typical results also include constraints on the camera position. For example, in an aerial image of an airport, ACRONYM may find alternative interpretations corresponding to a large airplane and a high camera altitude or a small airplane and a low camera altitude; either is consistent with the image and some generic airplane model classes. The main strengths of an ACRONYM style approach are the generality of its modeling philosophy and the concept of constraining its search based on local invariance of feature relationships. Its weaknesses stem from the complexity of the constraint-based modeling and constraint manipulation system and the lack of a hypothesis verification step that refers back to the original image data; thus the range of data complexity that can be treated is limited. This limitation, however, can in principle be overcome by incorporating techniques like those of the 3DPO system, which we describe next.

### C.5 Bolles and Horaud—3DPO: Model-Driven Correlation-Based Hypothesis Verification

The 3DPO system [Bolles and Horaud 1986] addresses the question of how to find objects having straightforward CAD models in a bin full of overlapping parts (Figure 20a). They assume that both monocular gray-level imagery and range data from a structured-light system are available and concentrate on the question of finding and gripping a particular part using rapid heuristics that would be practical in a controlled industrial environment.

#### *C.5.1 Object Models*

Much of 3DPO's speed derives from the fact that its hypothesis generation step relies on considering a few distinctive features, then verifying whether other expected features of the object are found in the original image. Such features are determined by carrying out a preliminary planning step to digest the best feature clusters and evaluate their use. This eliminates a great deal of time-consuming computation during the search process.

Object models in 3DPO also have some unique characteristics. In particular, each model provided to the recognition system is analyzed to provide answers to the following questions:

- How many features are there of each type and size?
- Which surfaces intersect to form a particular edge?
- Do other features lie in a given plane?
- What neighboring features are available to make one of a class of similar features distinct from its neighbors?

The answers to these questions are essential in constructing feature clusters that can be used to distinguish a good match from a bad hypothesis efficiently with minimal effort.

Another aspect of 3DPO's object modeling philosophy is to incorporate multiple object models for use in different phases of the search strategy. A complete model consists of a CAD model, a wire-frame model, a planar patch model, and a set of feature classification networks generated by preprocessing the model to isolate distinguishing features and feature groups. Three types of features are used extensively in 3DPO, although for different applications one might choose other types. The three types are straight

dihedrals (straight edges at the intersection of two planar surfaces), circular dihedrals (like the edge around the top of a cylinder), and straight tangentials (like the occlusion edge at the side of a cylinder). Each feature has its own peculiar signature; for example, a straight dihedral is characterized by its length, the size of the included angle, and the properties of the adjacent surfaces, such as their width and areas. To detect such features, the system begins by detecting discontinuities in the range data, linking the discontinuities into edge chains, and finding those that lie in a plane; the procedure then analyzes the surfaces adjacent to planar arcs and lines and refines the edge positions based on surface information.

#### C.5.2 System Design

To implement the 3DPO philosophy, we would carry out the following steps:

- Primitive feature detection
- Feature cluster formation
- Hypothesis generation
- Hypothesis verification
- Parameter refinement

The system starts by extracting edges from a range image, that is, locations where orientation changes sharply or is discontinuous. The thresholds for accepting edges are set relatively low in order to avoid missing any features. Then the three types of edges—straight dihedrals, circular dihedrals, and straight tangentials—are extracted. The recognition strategy first locates a key feature, then adds additional related features to form a cluster.

The accumulated feature clusters are then used to generate hypotheses about the potential locations of a part. This is a critical step in the search process since, if the chosen clusters are not sufficiently unique, combinatorics can quickly become overwhelming. Once a set of hypotheses has been put forward, the system looks back in the original image data to locate verifying information; in principle, we could either predict the location of other features and search for them or could take the features already tabulated and see which match the predictions. The latter is less reliable in principle since it does not reanalyze the data and therefore is not as likely to find things that were lost in the original analysis. 3DPO uses the planar patch model of the object to predict surface normals and compare them to the range data; this technique is effective when key verifying features are missing due to noise or occlusion by other parts.

The final step, parameter refinement, is needed to improve the accuracy of the part location and uses a method such as least-squares fitting to accomplish its goal. If multiple objects are to be recognized in the image data, the features of each object are deleted from the global list of features as they are found in order to reduce the search space for subsequent objects.

This procedure appears to be well suited to industrial applications because it is fairly robust. The use of a high-level strategy that can generate good hypotheses when the feature detection system loses features is combined with a low-level verification step that can go back and check hypotheses in the raw data. The low-level comparisons complement the feature-level hypothesis verification procedures and reduce 3DPO's dependence on the feature extraction step. Furthermore, direct comparison of the hypothesis with featureless regions in the range data can handle smooth objects for which few distinguishing features can be found in the original model.

### C.6 Fua and Hanson—MDL: Finding Complete Generic Objects Using Model-Driven Optimization

Fua and Hanson [1991] describe generic objects in terms of a language that specifies both photometric and geometric constraints on the objects and their appearance in the image. Buildings in aerial images are modeled as rectilinear structures whose internal gray level

intensities are planar, whereas roads are modeled by pairs of parallel, smoothly curved edges enclosing a planar intensity area. The authors define an information-theoretic objective function that expresses the quality of fit to the models in an algebraic form, then treat the problem of finding objects as one of generating an optimal description of the image in terms of both the language and the objective function.

### C.6.1 Theory

The problem of generating the best image description in terms of a set of models is phrased as one of maximizing the probability $P = p(m_0, m_1, \ldots, m_n \mid e_1, \ldots, e_n)$ that, given the evidence $E = \{e_i;\ i = 1 \ldots n\}$, describing the scene in terms of a particular set of model instances $M = \{m_i;\ i = 1 \ldots n\}$ and a background $m_0$ is correct. Each $m_i$ is taken to be a model instance, whereas $e_i$ is the measurable evidence specific to the $i$th model instance, typically a set of associated pixel intensities. Assuming that the objects' photometric properties are independent, it can be shown that the probability of the parse can be rewritten as

$$P \propto p(m_1, \ldots, m_n) \prod_{i=1}^{n} \frac{p(e_i \mid m_i)}{p(e_i)},$$

where $p(m_1, \ldots, m_n)$ is the prior probability that these $n$ instances appear in the scene.

The objective function is taken to be $S = \log_2 P$ and can be interpreted in terms of encoding cost [Hamming 1985; Shannon 1948]:

$$S = \log_2 P = F - G,$$

where

$$F = \sum_{i=1}^{n} F_i = \sum_{i=1}^{n} \{-\log p(e_i) + \log p(e_i \mid m_i)\}$$

$$G = -\log p(m_1, \ldots, m_n).$$

$F$ is defined to be the *encoding effectiveness* of the set of models. The $-\log p(e_i)$ terms give the number of bits needed to describe the evidence in the absence of the model, whereas the $-\log p(e_i \mid m_i)$ terms give the number of bits needed to describe the evidence using the modeling language. The use of the term effectiveness is thus motivated by the fact that $F$ represents the number of bits saved by representing the evidence using the model; $F$ increases as the fit improves. For example, the interior intensities of an 8-bit image region are modeled by a smooth intensity surface with a gaussian distribution of deviations from the surface. A region of area $A$ can be described in terms of a model requiring $p$ bits to describe the parameters of the surface and $kA$ bits to describe the image intensities. Here we define

$$k = \log \sigma + (1/2) \log (2 \pi e),$$

where $\sigma$ is the measured variance of the deviations from the smooth surface. Describing the same intensities without the model would require 8 bits per pixel, and the value of $F$ therefore is $(8 - k)A - p$.

$G$ is the number of bits needed to encode the evidence-free model representation information and quantifies the elegance of the chosen set of model instances with respect to the model language as well as their dependencies. For example, $G$ can be taken to be proportional to the length of the instance boundaries, thereby favoring compact objects. Because all the measures are expressed in terms of bits, distinct sources of information can be used simultaneously and their output made commensurate.

### C.6.2 Implementation

To generate optimal descriptions, a hierarchical procedure carries out the following steps: (1) Extract edges with the appropriate geometry, (2) find elementary geometric relationships between edges (such as corners or parallels), (3) build closed cycles of related edges that enclose areas with acceptable photometric and geometric properties, (4) invoke a

contour completion procedure that generates closed contours, optimizes their location, and computes their elevation, and (5) select the highest scoring contours.

Each parsing step is designed as a filtering process that both enforces some model constraints and limits the size of the search space, thereby preventing combinatorial explosion of the search. Multiple information sources (edge data, interior pixel intensities, stereographic information, and geometric constraints) are combined to build and rank hypotheses for generic objects of arbitrary complexity, such as the one in Figure 21.

### C.6.3 Properties

The framework described here accomplishes the following objectives:

- **Generic shape extraction.** For many important tasks, the exact shapes of objects of interest are not known. The models used in this approach describe common cartographic objects that obey specific geometric constraints but can be arbitrarily complex. The objective function balances the goodness of fit of model instances to the image data against their geometric quality. The system can therefore pick the best object hypotheses without using rigid geometric constraints or templates.
- **Integration of multiple data sources.** In general, objects are not characterized solely by their edge or area signatures. As a result, data-driven edge and region segmentation processes often fail to extract objects as such. Geometric constraints are combined with the photometric characteristics of the enclosed areas and their boundaries to generate and evaluate shape hypotheses simultaneously. When two or more images are available, stereoscopic construction can also be carried out. All available information is thus effectively exploited.

## ACKNOWLEDGMENTS

Paul Suetens' research has been supported in part by National Fund for Scientific Research, Belgium. Pascal Fua's research has been supported in part by the U.S. Defense Advanced Research Projects Agency and INRIA, France Andrew J. Hanson's research has been supported in part by the U.S Defense Advanced Research Projects Agency.

This work was supported in part by the NATO Collaborative Research Grants Programme under reference 0518/88 and by the Belgian National incentive-program for fundamental research in artificial intelligence, initiated by the Belgian State, Prime Minister's Office, Science Policy Programming Additional support was provided by INRIA, France, and by the U.S. Defense Advanced Research Projects Agency under Contract Nos MDA903-86-C-0084 and DACA76-85-C-0004. The scientific responsibility is assumed by the authors.

Numerous figures in this review are based on research databases of images available to the authors at SRI International or at ESAT, K.U. Leuven; we gratefully acknowledge these sources of illustrative material. Where noted, additional figures are reproduced with permission of the authors and/or publishers of the original articles.

Finally, the authors would like to thank Dr. Martin Fischler for his hospitality at SRI International and for his encouragement and suggestions.

## REFERENCES

AMINI, A. A., WEYMOUTH, T. E., AND ANDERSON, D J. 1989. A parallel algorithm for determining two-dimensional object positions using incomplete information about their boundaries *Pattern Recogn. 22*, 1, 21–28.

AYACHE, N AND FAUGERAS, O. 1986. HYPER: A new approach for the recognition and positioning of two-dimensional objects *IEEE Trans. Pattern Anal. Mach. Intell. PAMI 8*, 1, 44–54

BALLARD, D. H. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recogn. 13*, 111–122.

BALLARD, D. H. AND BROWN, C. M 1982 *Computer Vision*, Prentice-Hall, Englewood Cliffs, N.J.

BALLARD, D. H. AND SABBAH, D. 1981. On Shapes. *International Joint Conference on Artificial Intelligence*, 607–612.

BARR, A. H. 1981. Superquadrics and angle-preserving transformations. *IEEE Comput. Graph. Appl.* 1, 11–23.

BELLMAN, R. AND DREYFUS, S. 1962. *Applied Dynamic Programming*, Princeton Univ Press, Princeton, N. J

BERGMAN, A AND MULGAONKAR, P. G. 1988. Neural networks for address-block ranking: A comparison with classical techniques. In *Proceedings of USPS Third Advanced Technology Conference* (Washington, D C., May, 3–5).

BINFORD, T. O. 1982. Survey of model-based image analysis systems. *Int. J. Rob. Res. 1*, 118–64.

BOBICK, A. F. AND BOLLES, R. C. 1989. Representation space: An approach to the integration of visual information. In *Proceedings of Computer Vision and Pattern Recognition* (San Diego), IEEE Computer Science Press, Washington, 492-499.

BOLLES, R. C. AND HORAUD, P. 1986. 3DPO: A three-dimensional part orientation system. *Int. J. Rob. Res. 5*, 3, 3-26.

BROOKS, R. A. 1981. Symbolic reasoning among 3-D models and 2-D images. *Artif. Intell. 17*, 285-348.

BROOKS, R. A. 1983. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI 5*, 2, 140-150.

CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI-8*, 6, 679-698.

CLOWES, M. B. 1971. On seeing things. *Artif. Intell. 2*, 1, 79-116.

DRAPER, B., COLLINS, R., BROLIO, J., HANSON, A., AND RISEMAN, E. 1988. The schema system. Univ. of Massachusetts at Amherst, Computer and Information Science Technical Rep. COINS TR88-76, Sept.

DUDA, R. O. AND HART, P. E. 1973. *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York.

ESHERA, M. A. AND FU, K.-S. 1986. An image understanding system using attributed symbolic representation and inexact graph matching. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI-8*, 5, 604-618.

FAN, T.-J., MEDIONI, G., AND NEVATIA, R. 1988. Matching 3-D objects using surface desriptions. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation* (Philadelphia), pp. 1400-1406.

FAN, T.-J., MEDIONI, G., AND NEVATIA, R. 1989. Recognizing 3-D objects using surface descriptions. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI 11*, 11, 1140-1157.

FAUGERAS, O. AND BERTHOD, M. 1981. Improving consistency and reducing ambiguities in stochastic labeling: An optimization approach. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI 3*, 4, 412-424.

FISCHLER, M. A. AND ELSCHLAGER, R. A. 1973. The representation and matching of pictorial structures. *IEEE Trans. Comput. C-22*, 1, 67-92.

FISCHLER, M. A., TENENBAUM, J. M., AND WOLF, H. C. 1981. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Comput. Graph. Image Process. 15*, 201-223.

FUA, P. 1989. Object delineation as an optimization problem: A connection machine implementation. In *Proceedings of the 4th International Conference on Supercomputing* (Santa Clara, Calif.), pp. 476-484.

FUA, P. AND HANSON, A. J. 1987. Using generic geometric models for intelligent shape extraction. In *Proceedings of the AAAI 6th National Conference on Artificial Intelligence*, (Los Altos, CA, July), Morgan-Kaufmann, pp 706-711.

FUA, P. AND HANSON, A. J. 1988. Extracting generic shapes using model-driven optimization. In *Proceedings of the DARPA Image Understanding Workshop*, pp. 994-1004.

FUA, P. AND HANSON, A. J. 1991. An optimization framework for feature extraction. *Mach. Vision Appl. 4*, 59-87.

FUA, P. AND LECLERC, Y. G. 1990. Model driven edge detection. *Mach. Vision Appl. 3*, 45-56.

GARDIN, F. AND MELTZER, B. 1988. Analogical representations of naive physics. *Artif. Intell. 36*, 91-123.

GEMAN, S. AND GEMAN, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI 6*, 721-741.

GERBRANDS, J. J., BACKER, E., AND VAN DER HOEVEN, W. A. G. 1986. Quantitative evaluation of edge detection by dynamic programming. In *Pattern Recognition in Practice II*, E. S. Gelsema and L. N. Kanal, Eds Elsevier Science Publishers B. V., Amsterdam, pp. 91-99.

GRIMSON, E. AND HUTTENLOCHER, D. 1990. On the sensitivity of the Hough transform for object recognition. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI 12*, 3, 255-274.

GROEN, F. C. A., TEN KATE, T. K. SMEULDERS, A. W. M., AND YOUNG, I. T. 1989. Human chromosome classification based on local band descriptors *Pattern Recogn. Lett. 9*, 3, 211-222.

HALL, E. L. 1979. *Computer Image Processing and Recognition*, Academic Press, New York.

HAMMING, R. W. 1985. *Coding and Information Theory*, Prentice Hall, Englewood Cliffs, N.J.

HERMAN, M. AND KANADE, T. 1986. Incremental reconstruction of 3D scenes from multiple, complex images. *Artif. Intell. 30*, pp. 289-341.

HORAUD, R. AND SKORDAS, T. 1989. Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI 11*, 11, 1168-1180.

HOUGH, P. V. C. 1962. Methods and Means for Recognizing Complex Patterns, U.S. Patent 3069654.

HUERTAS, A. COLE, W., AND NEVATIA, R. 1987. Detecting runways in aerial images. In *Proceedings of the DARPA Image Understanding Workshop*. pp. 272-297.

HUERTAS, A. AND NEVATIA, R. 1988. Detecting buildings in aerial images. *Comput. Vision Graph. Image Process. 41*, 131-152.

HUFFMAN, D. A. 1971, Impossible objects as nonsense sentences. *Mach. Intell. 6*, 295-323.

HWANG, V. S., DAVIS, L. S., AND MATSUYAMA, T. 1986. Hypothesis integration in image under-

standing systems. *Comput. Vision Graph. Image Process. 36*, 321-371.

ILLINGWORTH, J. AND KITTLER, J. 1988. A survey of the Hough transform. *Comput. Vision Graph. Image Process. 44*, 87-116.

JAIN, A. K. AND HOFFMAN, R. 1988. Evidence-based recognition of 3-D objects. *IEEE Trans. Pattern Anal. Mach. Intell., 10*, 6, 783-802.

KALVIN, A., SCHONBERG, E., SCHWARZ, J. T., AND SHARIR, M. 1986. Two-dimensional model-based boundary matching using footprints. *Int. J. Rob. Res. 6*, 4.

KANADE, T. 1980. Survey—Region segmentation: Signal versus semantics. *Comput. Graph. Image Process. 13*, 279-297.

KASS, M., WITKIN, A., AND TERZOPOULOS, D. 1987. SNAKES: Active contour models. *Int. J. Comput. Vision 1*, 4, 321-331.

KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. 1983. Optimization by simulated annealing. *Science 220*, 671-680.

KITTLER, J. AND ILLINGWORTH, J. 1985. Relaxation labeling algorithms: A review. *Image Vision Comput. 3*, 4, 206-216.

LAMDAN, Y. AND WOLFSON, H. J. 1988. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*. IEEE Computer Society Press, Washington.

LAWS, K. I. 1984. Goal-directed texture segmentation. Tech. Note 334, Artificial Intelligence Center, SRI International, Menlo Park, California. This method is an extension to the Ohlander et al. [1978] segmentation method.

LECLERC, Y. G. 1989. Constructing simple stable descriptions for image partitioning. *Int. J. Comput. Vision 3*, 73-102.

LEVY-MANDEL, A. D., VENETSANOPOULOS, A N , AND TSOTSOS, J. K. 1986. Knowledge-based landmarking of cephalograms. *Comput. Biomed. Res. 19*, 282-309.

LOWE, D. G. 1987. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell. 31*, 355-395.

MACKWORTH, A. K. 1973. Interpreting pictures of polyhedral scenes. *Artif. Intell. 4*, 121-137

MAITRE, H. AND WU, Y. 1987. Improving dynamic programming to solve image registration. *Pattern Recogn. 20*, 4, 443-462.

MALIK, J. 1987. Interpreting line drawings of curved objects. *Int. J. Comput. Vision 1*, 73-103.

MANSOURI, A.-R., MALOWANY, A. S., AND LEVINE, M. D. 1987. Line detection in digital pictures: A hypothesis prediction verification paradigm. *Comput. Vision Graph. Image Process. 40*, 95-114.

MANTAS, J. 1987. Methodologies in pattern recognition and image analysis: A brief survey. *Pattern Recogn. 20*, 1, 1-6.

MARR, D. 1982. *Vision*, W H. Freeman, San Francisco, Calif.

MATSUYAMA, T. 1989. Expert systems for image processing: Knowledge-based composition of image analysis processes. *Comput. Vision Graph. Image Process. 48*, 22-49.

MCKEOWN, D. M. AND DENLINGER, J. L. 1988. Cooperative methods for road tracking in aerial imagery. In *Proceedings of the IEEE Computer Vision and Pattern Recognition* (Ann Arbor, Mich.), IEEE Computer Society Press, Washington, pp. 662-672.

MCKEOWN, D. M., HARVEY, W. A., AND MCDERMOTT, J. 1985. Rule-based interpretation of aerial images. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI 7* 5, 570-585

MCKEOWN, D. M., HARVEY, W. A., AND WIXSON, L. E. 1989. Automating knowledge acquisition for aerial image interpretation. *Comput. Vision Graph. Image Process. 46*, 37-81.

MOHAN, R. AND NEVATIA, R. 1988. Perceptual grouping for the detection and description of structures in aerial images In *Proceedings of the DARPA Image Understanding Workshop* (Los Altos, CA), Morgan-Kaufmann, pp. 512-526.

MOHAN, R. AND NEVATIA, R. 1989. Using perceptual organization to extract 3-D structures. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI 11*, 11, 1121-1139.

MULGAONKAR, P. G., SHAPIRO, L. G., AND HARALICK, R. M. 1984. Matching "sticks, plates and blobs" objects using geometric relational constraints. *Image Vision Comput. 2*, 2, 85-98.

MURRAY, D. W. 1987. Model-based recognition using 3D shape alone. *Comput. Vision Graph. Image Process.* 40, 250-266.

MURRAY, D. W. AND BUXTON, B. F. 1987. Scene segmentation from visual motion using global optimization. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI 9*, 2, 220-228.

NAGAO, M. 1984. Control strategies in pattern analysis. *Pattern Recogn. 17*, 1, 45-56.

NAGAO, M. AND MATSUYAMA, T. 1980. *A Structural Analysis of Complex Aerial Photographs*, Plenum Press, New York.

NANDHAKUMAR, N. AND AGGARWAL, J. K. 1985. The artificial intelligence approach to pattern recognition: A perspective and an overview. *Pattern Recogn. 18*, 6, 383-389.

NAZIF, A. M. AND LEVINE, M. D. 1984. Low level image segmentation: An expert system. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI 6*, 5, 555-577.

NIBLACK, W. AND PETKOVIC, D. 1988. On improving the accuracy of the Hough transform. In *Proceedings of Computer Vision and Pattern Recognition* (Ann Arbor, Mich.), IEEE Computer Society Press, Washington, pp. 574-579.

NUYTS, J. MORTELMANS, L., SUETENS, P., OOSTERLINCK, A. AND DE ROO, M. 1989.

Model-based quantification of myocardial perfusion images from SPECT. *J. Nuclear Med. 30*, 1992–2001.

OHLANDER, R., PRICE, K., AND REDDY, D. R. 1978. Picture segmentation using a recursive region splitting method. *Comput. Graph. Image Process. 8*, 3, 313–333.

OHTA, Y. 1985. *Knowledge-Based Interpretation of Outdoor Natural Color Scenes.* Pitman Advanced Publishing Program, Boston.

PAVLIDIS, T. 1986. A critical survey of image analysis methods. In *Proceedings of the 8th International Conference on Pattern Recognition.* (Paris, France), IEEE Press, NY, pp. 502–511.

PEARL, J. P. 1985. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, Reading, Mass.

PENTLAND, A. P. 1990. Automatic extraction of deformable part models. *Int. J. Comput. Vision 4*, 107–126.

PREMOLI, A., GRATTONI, P., AND POLLASTRI, F. 1989. A non-sequential contour detection with a prior knowledge. *Pattern Recogn. Lett. 9*, 45–51.

QUAM, L. H. 1978. Road tracking and anomaly detection in aerial imagery. SRI International AI Center Tech. Note 158. In *Proceedings of the ARPA Image Understanding Workshop*, (May) pp. 51–55.

RAO, A. R. AND JAIN, R. 1988. Knowledge representation and control in computer vision systems. *IEEE Expert*, 64–79.

REYNOLDS, G. O. DEVELIS, J. B., PARRENT, G. B., JR. AND THOMPSON, B. J. 1989. The new physical optics notebook: Tutorials in fourier optics. In *Detection of Objects by Complex Inverse Filtering*, SPIE and AIP, New York, pp. 417–420.

RICHARDS, J. A 1986. *Remote Sensing Digital Image Analysis*, Springer-Verlag, Berlin.

ROSENFELD, A. 1969. *Picture Processing by Computer*, Academic Press, New York.

ROSENFELD, A. 1987. Image analysis: Problems, progress and prospects. In *Readings in Computer Vision*, M. A. Fischler and O. Firschein, Eds. Morgan Kaufmann Publishers, Inc., pp. 3–12.

ROSENFELD, A., HUMMEL, R. A., AND ZUCKER, S. W. 1976. Scene labeling by relaxation operations. *IEEE Trans. Syst. Man Cybern. SMC 6*, 6, 420–433.

SHANNON, C. E. 1948. A mathematical theory of communication. *Bells Syst. Tech. J. 27*, 623–656.

SHNEIER, M. O., LUMIA, R., AND KENT, E. W. 1986. Model-based strategies for high-level robot vision. *Comput. Vision Graph. Image Process. 33*, 293–306.

SUETENS, P., SMETS, C., VAN DE WERF, F., AND OOSTERLINCK, A. 1989. Recognition of the coronary blood vessels on angiograms using hierarchical model-based iconic search. In *Proceedings of Computer Vision and Pattern Recognition* (San Diego, Calif.), IEEE Computer Society Press, Washington, pp. 576–581.

TENENBAUM, J. M. AND BARROW, H. G. 1977 Experiments in interpretation-guided segmentation. *Artif. Intell. 8*, 241–274.

TENENBAUM, J. M., BARROW, H. G., BOLLES, R. C., FISCHLER, M. A., AND WOLF, H. C. 1979. Map-guided interpretation of remotely-sensed imagery. *IEEE Pattern Recogn. Image Process.*, 610–617.

TERZPOULOS, D., WITKIN, A., AND KASS, M. 1988. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artif. Intell. 36*, 91–123.

TOU, J. T. AND GONZALES, R. C. 1974. *Pattern Recognition Principles*, Addison-Wesley, Reading, Mass.

WALLACE, A. M. 1988. A comparison of approaches to high-level image interpretation. *Pattern Recogn. 21*, 3, 241–259.

WANG, C.-H. AND SRIHARI, S. N. 1988. A framework for object recognition in a visually complex environment and its application to locating address blocks on mail pieces. *Int. J. Comput. Vision 2*, 125–151.

WHEELER, S. G. AND MISRA, P. N. 1980. Crop classification with landsat multispectral scanner data II. *Pattern Recogn. 12*, 219–228.

WITKIN, A., FLEISHER, K., AND BARR, A. 1987a. Energy constraints on parameterized models. *Comput. Graph. 21*, 4, 225–232.

WITKIN, A., TERZOPOULOS, D., AND KASS, M. 1987b. Signal matching through scale space. *Int. J. Comput. Vision 1*, 2, 133–144.

WU, Q., SUETENS, P., AND OOSTERLINCK, A. 1987. Toward an expert system for chromosome analysis. *Knowledge-Based Syst. 1*, 1, 43–52.

WU, Q., SUETENS, P., AND OOSTERLINCK, A. 1989 On knowledge-based improvement of biomedical pattern recognition: A case study. In *Proceedings of the 5th IEEE Conference on Artificial Intelligence Applications* (Miami, Fla.), IEEE, pp. 239–244.

YAMADA, H., MERRITT, C., AND KASVAND, T. 1988 Recognition of kidney glomerulus by dynamic programming matching method. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 5, 731–737.

ZHANG, Z. AND SIMAAN, M. 1987. A rule-based interpretation system for segmentation of seismic images. *Pattern Recogn. 20*, 1, 45–53.

Received September 1989; final revision accepted October 1991.