COMPUTER SCIENCE TECHNICAL REPORT SERIES

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

TR-1157
DAAG-53-76C-0138

2

April 1982

COMPUTER VISION

Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

Computer vision deals with the analysis and interpretation of images. Over the past 25 years, many ad hoc techniques for analyzing images have been developed, but during the past few years this technology has begun to develop a scientific basis. This paper outlines the major components of a general computer vision system, and summarizes the state of the art in each of these components.

The support the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Janet Salzman in preparing this paper.

1. Introduction

Over the past 25 years, much of the work on applications of pattern recognition, and a significant fraction of the work in artificial intelligence, has dealt with the analysis and interpretation of images. This subject has been variously known as pictorial pattern recognition, image analysis, scene analysis, image understanding, and computer vision. Its applications include document processing (character recognition, etc.), microscopy, radiology, industrial automation (inspection, robot vision), remote sensing, navigation, and reconaissance, to name only the major areas.

Many ad hoc techniques for analyzing images have been developed, so that a large assortment of tools is now available for solving practical problems in this field. Most important, during the past few years the field has begun to develop a scientific basis. This paper outlines the major steps in the computer vision process, and summarizes the state of the art with respect to each of these steps.

2. A computer vision paradigm

The goal of computer vision is the construction of scene descriptions on the basis of information extracted from images or image sequences. With reference to Figure 1, the following are some of the major steps in the computer vision process. We consider here only images obtained by optical sensors, though some of the discussion is also applicable to other types of sensors.

Many types of scenes are essentially two-dimensional; documents are an obvious example, but two-dimensional treatment is often quite adequate in applications such as remote sensing (flat terrain seen from very high attitudes), radiology (where the image is a "shadow" of the object), or microscopy (where the image is a cross-section of the object). In such situations, the image analysis process is basically two-dimensional. We extract "features" such as edges from the image, or segment the image into regions, thus obtaining a map-like representation, which Marr at MIT called the "primal sketch" consisting of image features labelled with their property values. Grouping processes may then be used to obtain improved maps from the initial one. The maps may be represented by abstract relational structures in which, e.g., nodes represent regions, labelled with various property values (color, texture, shape, etc.), and arcs represent relationships among regions. Finally, these structures are matched against stored models, which are generalized relational structures representing classes of maps that correspond to general types of images. Successful matches yield identifications for the image parts, and a structural description of the image, in terms of known entities.

In other situations, notably in robot vision applications, the scenes to be described are fundamentally three-dimensional, involving substantial surface relief and object occlusion. Successful analysis of images of such scenes requires a more elaborate approach in which the three-dimensional nature of the underlying scenes is taken into account. Here the key step in the analysis is to infer the surface orientation at each image point. Clues to surface orientation can be derived directly from shading (i.e., gray level variation) in the image. Alternatively, two-dimensional segmentation and feature extraction techniques can first be applied to the image to extract such features as surface contours and texture primitives, and surface orientation clues can then be derived from contour shapes or from textural variations. Using the surface orientation map, which Marr called the "2½D sketch", feature extraction and segmentation techniques can once again be applied to yield a segmentation into (visible parts of) bodies or objects, and these can in turn be represented by a relational structure. Finally, the structure can be matched against models to yield an interpretation of the scene in terms of known objects. Note that the matching process is more difficult in the three-dimensional case, since the image only shows one side of each object, and objects may partially occlude one another.

The computer vision paradigm described in the last two paragraphs, and illustrated in Figure 1, is highly simplified in several respects. The following are some of the directions in which it needs to be extended or generalized:

a) Ideally, the value (gray level or spectral signature) at each point of an image represents the light received from the scene along a given direction, but these values will not be perfectly accurate because of degradations arising in the process of imaging (for example, blur and noise introduced by the environment or the sensor) or digitization. Image restoration techniques should be used to correct the image values before performing the steps outlined in Figure 1. (Feature extraction may be useful as an aid in estimating the degradations in order to perform effective restoration.)

b) We have assumed in Figure 1 that only a single image of the scene is available as input. If two images taken from different viewpoints are available, stereomapping techniques can be used to construct the surface orientation map by matching either image values or extracted features on the two images and measuring their parallaxes.

If images taken at different times are available, comparing them yields information about the motion of the sensor or of objects in the scene. In this case, the processes of segmentation, model matching, etc. should be performed on the image sequence rather than on the individual images.

c) Figure 1 shows a "one-way" process in which we start with the image and successively construct a 2D map, a 2½D map, etc. More realistically, the arrows in Figure 1 should point both ways. Knowledge about the expected results of a process (segmentation, etc.) should be used to criticize the actual results and modify the process so as to improve them.

d) The model matching process may be hierarchical, with objects composed of subobjects, etc. Hierarchical models are extensively used in syntactic approaches to pattern recognition.

A discussion of image restoration techniques is beyond the score of this article, but stereomapping, time-varying imagery analysis, syntactic methods, and the use of feedback in image analysis will all be briefly discussed in later sections.

This article reviews the basic stages in the computer vision process from a technique-oriented, rather than application-oriented, standpoint. Methods currently used at each stage are

reviewed, their shortcomings are discussed, and approaches that show promise of yielding improved performance are described. The specific areas covered are feature extraction, image matching, segmentation, texture analysis, surface orientation estimation, object representation, and model matching.

3. Feature extraction

The extraction of features such as edges and curves from an image is useful for a variety of purposes. Linear features (curves) are often of importance in their own right, e.g., roads or drainage patterns on low-resolution remote sensor imagery. Edges are useful in image matching for obtaining sharp matches that are insensitive to grayscale distortions (but quite sensitive to geometric distortion); see Section 4. Edges can be used in conjunction with various segmentation techniques to improve the quality of a segmentation (Section 5). Edges and similar locally defined features play important roles in texture analysis; see Section 6. The interpretation of image edges as arising from various types of discontinuities in the scene (occluding edges are discontinuities in range; convex or concave edges are discontinuities in surface slope; shadow edges are discontinuities in illumination) plays an important role in the inference of 3D surface structure from an image (Section 7).

The classical approach to edge detection makes use of digital (finite-difference) versions of standard isotropic derivative operators such as the gradient or Laplacian. A closely related approach is to linearly convolve the image with a set of masks representing ideal step edges in various directions. Lines and curves can be similarly detected by linear convolutions. However, linear operators are not specific to features of a given type; they also respond in other situations involving local intensity changes. An alternative approach, developed at the University of Maryland in 1970, is to use "gated" (nonlinear) operators that respond only when specific relationships hold among the local intensities - e.g., all intensities along the line higher than all the flanking intensities on both sides, and similarly for edges. In all of these approaches, the output is a quantitative edge or curve value; the final detection decision can be made, if desired, by thresholding this value. Similar methods can be used to detect edges defined by discontinuities in color, rather than in intensity.

Several important improvements to the edge detection process have been made over the past decade. New classes of operators have been defined based on fitting polynomial surfaces to the local image intensities, and using the derivatives of these polynomials (which can in turn be expressed in terms of the local intensities) as edge value estimates. This method, which was first proposed by Prewitt (now at NIH) in 1970, allows edges to be located (at maxima of the surface gradient, e.g.) to subpixel accuracy. Another important idea, first proposed by Hueckel at Stanford University in 1970, is to find a best-fitting step edge (or edge-line) to the local intensities.

Classical edge detectors were based on small image neighborhoods, typically 3×3; a more powerful approach, developed at the University of Maryland in 1970, and later refined by Marr and his students at MIT, is to use a set of first- or second-difference operators based on neighborhoods having a range of sizes (e.g., increasing by factors of 2), and combine their outputs, so that discontinuities can be detected at many different scales; here the edges are localized at maxima of first differences, or at zero-crossings of second differences. Operators based on large neighborhoods can also be used to detect texture edges, at which the statistics of various local image properties change abruptly. Cooperation between operators in different positions can be used to enhance the feature values at points lying on smooth edges or curves; this was one of the first applications of "relaxation" methods, developed at the University of Maryland in the mid-1970's, to image analysis at the pixel level.

The standard approaches to edge detection are implicitly based on a very simple model in which the image is regarded as ideally composed of essentially "constant" regions separated by step edges. Recent work by Haralick at Virginia Polytechnic Institute is based on the more general assumption of a piecewise linear, rather than piecewise constant, image, which allows simple shading effects to be taken into account. Research is needed on the development of algorithms designed to detect intensity edges resulting from specific types of scene discontinuities, including shadow edges, slope edges, and range edges. Detection of texture edges too should be based on models for surface texture, rather than for image texture. The operators should incorporate cooperative computation across both positions and sizes.

4. Image matching

Image matching and registration are used for a number of different purposes. By registering two images of a scene obtained from different sensors, one can obtain the multisensor (e.g., multispectral) characteristics of each scene point, which can then be used to classify the points. By comparing images obtained from different locations, one can compute the stereoscopic parallaxes of scene points that are visible on both images, and thus determine their 3D positions. By comparing images taken at different times, one can detect changes that have taken place in the scene, e.g., due to motion of the sensor or motions of objects in the scene. In all of these tasks, registration is carried out by finding pairs of subimages that match one another closely. Subimage matching is also used to detect the occurrence of specific patterns ("templates" or "control points") in an image, for purposes of location (e.g., navigation) or object detection.

Classically, image matching has used match measures derived from cross-correlation computation, or sometimes mismatch measures based on sums of absolute differences; both of these approaches involve point-by-point intensity comparison of the images being matched. Such processes are unsatisfactory for several reasons: they often yield unsharp matches, making it difficult to decide when a match has been detected;

they are sensitive to distortions in both grayscale and geometry; and they are computationally expensive. Match sharpness and grayscale insensitivity can be greatly increased by applying derivative operators, possibly followed by thresholding, to the images before matching - for example, taking first derivatives (e.g., gradient magnitudes) of both images, or the second derivative (e.g., Laplacian magnitude) of one image. Geometric insensitivity can be improved by matching smaller pieces or local features (which are less affected by geometric distortion), and then searching for combinations of such matches in approximately the correct relative positions (this approach was developed by Fischler in the early 1970's), or using relaxation methods to reinforce such combinations (as was done at the University of Maryland in the mid-1970's, and more recently by Faugeras at the University of Southern California); this hierarchical approach also serves to reduce the computational cost of the matching process. An alternative idea, originated by Price and Reddy at Carnegie-Mellon University in the late 1970's, is to segment the image into parts, represent the parts and their relationships by a graph structure, and match these graph structures (see Section 9); here too, relaxation methods are useful.

Another approach to pattern matching makes use of geometric transformations that map instances of a given pattern into peaks in a transform space. This "Hough transform" approach was originally developed to handle simple classes of shapes such as straight lines or circles, but it was recently extended by Ballard at the University of Rochester to arbitrary shapes in both two and three dimensions. The diffculty of matching in the 3D case will be further discussed in Section 9.

5. Segmentation

Descriptions of an image generally refer to significant parts (regions; global features such as contours or curves) of which the image is composed; thus image description requires segmenting it into such parts. A much more challenging task is to segment the image into parts corresponding to the surfaces or bodies of which the underlying scene is composed; this is often very hard to do, since variations in image intensity may not be good indicators of physical variations in the scene, and conversely, physical variations do not always give rise to intensity variations.

The most commonly used approach to image segmentation involves classification of the individual image points (pixels) into subpopulations; the parts obtained in this way are just the subsets of pixels belonging to each class. The classification can be done on the basis of intensity alone ("thresholding"), of color or spectral signature, or of local properties derived from the neighborhood of the given pixel; the last approach is used in feature detection (e.g., classify a pixel as on an edge if the value of some locally computed derivative operator is high in its neighborhood), and it can also be used to segment an image into differently textured regions. Pixels can be classified using a set of properties simultaneously, or the properties can be used one at a time to recursively refine the segmentation, as demonstrated by Ohlander at Carnegie-Mellon University.

Pixels are usually classified independently, which allows fast implementation on parallel hardware; better results can be obtained by classifying sequentially, so that regions composed of pixels belonging to a given class can be "grown" in accordance with given constraints, but such approaches are inherently slow and would not be appropriate for use in real-time systems. Another possibility is to use a relaxation approach in which pixels are classified fuzzily, and the class memberships are then adjusted to favor local consistency; this approach, developed at the University of Maryland, requires a short sequence of iterations each of which can be implemented in parallel. In addition to local consistency, other sources of convergent evidence can be used to improve the quality of segmentation; for example, the classification criteria can be adjusted so as to maximize the edge strengths around the resulting region borders.

Image models should play an important role in image segmentation, but the models used in practice are usually much too simple. In segmenting an image by pixel classification, it is always assumed that the subpopulations are homogeneous, i.e., have essentially constant feature values (intensity, color, etc.). For scenes containing curved surfaces, this assumption is very unrealistic; even if variations in illumination are ignored, changes in surface orientation will give rise to changes in feature values on the image of the surface. Haralick's recent "facet model" allows certain types of variations in feature values (e.g., linear), but the role of surface orientation needs to be made more explicit. By making local orientation estimation an integral part of the segmentation process, and using these estimates to correct the feature values, it should be possible to *cooperatively* compute orientation estimates that optimize the clustering of feature values into subpopulations representing homogeneous surfaces (and not merely homogeneous regions). Spatial consistency constraints, as well as other types of convergent evidence, can also be incorporated into this process.

An alternative approach to segmentation is region-based, rather than pixel-based; an example is the split-and-merge approach advocated by Pavlidis (now at Bell Laboratories), where the goal is to partition an image into homogeneous connected regions by starting with an initial partition and modifying it by splitting regions if they are not sufficiently homogeneous, and merging pairs of adjacent regions if their union is still homogeneous. In this approach, "homogeneous" might mean approximately constant in intensity, or more generally, it might mean a good fit to a polynomial of some degree >0, as in the facet model. Still more generally, the merging and splitting can be controlled by a "semantic" model which estimates probable interpretations of the regions and performs merges or splits so as to increase the likelihood and consistency of the resulting image interpretation. This approach was advocated by Feldman and Yakimovsky at Stanford, and by Tenenbaum and Barrow at SRI, in the mid-1970's. Note, however, that these methods still make no explicit use of surface orientation estimation; they should be based on object semantics rather than region semantics. Grouping locally detected features (edges or lines) into global contours or curves can be done on the basis of global shape (Hough transforms; see Section 4), but if this is not known in advance, one can use methods analogous to split-and-merge - e.g., break a curve at branch points or sharp turns; link curves if they continue one another smoothly. Here again, it would be desirable to modify these criteria to take surface curvature into account.

A major drawback of segmentation based on pixel classification, particularly when it is implemented in parallel, is the difficulty of incorporating geometric knowledge about the desired regions into the segmentation process. The standard approach is to segment, measure geometric properties of the resulting regions, and then attempt to improve the values of these properties by adjusting parameters of the segmentation process; but it would be much preferable to make use of geometric constraints in the segmentation process itself. In region-based segmentation, since the units being manipulated are (pieces of) regions rather than pixels, somewhat greater control over region geometry can be achieved, by biasing the choices of splits and merges to favor the desired geometry. Another possibility, under investigation at the University of Maryland, is to perform segmentation using a multi-resolution ("pyramid") image representation, in which region geometry is coarsely represented by local patterns of "pixels" at the low-resolution levels; here segmentation is based on a cooperative process of pixel linking, which can be designed so that the linking is facilitated if it will give rise to the desired types of low-resolution local patterns. This approach too should be combined with surface orientation estimation, perhaps carried out at multiple resolutions.

6. Texture analysis

Textural properties of image regions are often used for classification (e.g., of terrain types of materials), or for segmentation of the image into differently textured regions. Changes in texture "coarseness" also provide important cues about surface slant and relative range; the direction in which coarseness is changing most rapidly corresponds, for a uniformly textured surface, to the slant direction, while an abrupt change in coarseness indicates the possibility of an occluding edge.

Classically, textural properties have been derived from the autocorrelation or Fourier power spectrum; for example, the coarser the texture (in a given direction), the slower its autocorrelation falls off in that direction from the origin (zero displacement) and the faster its power spectrum falls off in that direction from zero frequency. A related approach, studied extensively by Julesz at Bell Laboratories and by Haralick, characterizes textures by their second-order intensity statistics, i.e., by the frequencies with which given pairs of gray levels occur at given relative displacements. It has long been realized, however, that first order statistics of various local property values (e.g., responses to operators sensitive to local features such as edges, lines, line ends, etc.) are at least equally effective in texture discrimination.

More recent work, by Marr at MIT and others, suggests that local processes of linking between local features, giving rise to "texture elements" or "primitives", also play a significant role in the perception of texture differences. Texture discrimination based on second-order statistics of local features (e.g., occurrences of edge elements in given relative positions and orientations) has begun to be investigated (e.g., by Davis at the University of Texas). Texture analysis based on explicit extraction of primitives has also been explored (e.g., by Maleson and Feldman at the University of Rochester); here statistics derived from properties of the primitives, or of pairs of adjacent primitives, are used as textural properties.

All of this work has dealt with texture as an image property, and has been primarily concerned with uniformly textured regions, such as might arise from non-perspective views of uniformly textured surfaces. Research is needed on the development of texture analysis methods that take surface geometry into account, and that perform cooperative estimation of surface slant and surface texture characteristics, leading to better estimates of both. Similarly, methods of texture-based segmentation or texture edge detection should consider both surface geometry differences and texture differences; and in stereomatching of textured regions, one should use surface slant estimates to correct for the effects of perspective on the quality of the match.

7. Surface orientation estimation

If a high-resolution range sensor is available, the shapes of the visible surfaces in a scene can be obtained directly by constructing a range map. In this section we assume that range information is not directly available. In its absence, range can be inferred from stereopairs, by measuring stereo parallax; or relative range can be inferred from image sequences obtained from a moving sensor, by analyzing the motions of corresponding pixels from frame to frame ("optical flow"). If only a single image is available, one can still make inferences about changes in surface range (i.e., about surface slant) from single images, using clues derived from changes in gray level ("shading"), changes in texture, or from the shapes of edges or curves that lie on or bound a surface.

Stereomapping is based on identifying corresponding points in the two images using image matching techniques. As indicated in Section 4, matching performance is improved if we match features such as edges, rather than intensity values. The MIT approach to stereo, proposed by Marr and implemented by Grimson, is based on applying a set of edge operators, having a range of sizes, to the images; matching the edges produced by the coarsest operator, to yield a rough correspondence between the images; and then refining this correspondence by using successively finer eges. Edge-based approaches may still yield ambiguous results in heavily textured regions where edges are closely spaced. The ambiguity can be reduced by using intensity matching as a check, or by classifying the edges into types (e.g., discontinuities in illumination, range, or orientation) and requiring that corresponding edges be of the same type. In general, matching should be based on feature descriptions, rather than on raw feature response values. Work is needed on the development of matching methods based on other feature types, and particularly on features derived from surface orientation maps - e.g., matching of surface patches. Matching yields a set of range values at the positions of features; Grimson has developed methods of fitting smooth range surfaces to these values. For wide-angle stereo, where there is significant perspective distortion, derivation of a camera model and rectification of the images prior to matching are very desirable; extensive work in this area has been done at Stanford and at SRI.

When a static scene is viewed by a moving sensor, yielding a succession of images, the relative displacements of pixels from one image to the next are known as "optical flow". If these displacements could be computed accurately, it would be possible in principle to infer the motion of the sensor relative to the scene and the relative distances of the scene points from the sensor (but note that there is an inherent speed/range ambiguity). Ideally, the displacements can be estimated by comparing the space and time rates of change of the image intensity; but in practice, these estimates are quite noisy. Horn at MIT has developed an iterative method of estimating a smooth displacement field, but it yields inaccurate results at object boundaries. It should be possible to obtain improved results by combining the rate of change approach with edge detection and matching. For larger displacements, a matching approach can be used to determine corresponding points in successive frames. Ullman at MIT has shown that the motion of a rigid object is completely determined if we know the correspondence betweeen a few points on the object as they appear in two or three successive images. Extensions of this work to jointed objects have also been investigated.

If only a single image of a scene is available, clues about the orientations and relative ranges of the visible surfaces in the scene can still be derived from a number of sources. One source of such information is the shapes of edges in the image, representing occluding contours at the boundaries of surfaces, or contours that lie on the surfaces. The early work of Guzman, Clowes, Huffman, and Waltz, at MIT and elsewhere, as well as the more recent work of Kanade at Carnegie-Mellon, developed methods of inferring the nature of edges in the scene (e.g., convex, concave, or occluding) from the shapes of the junctions at which the edges meet as seen in the image. More recently, researchers at Stanford and Carnegie-Mellon have formulated a variety of constraints on the scene that can be derived from global properties of contours in the image. For example, if an edge is continuous or straight, or two edges are parallel, or two features coincide, in the image, we assume that the same is true in the scene; and if a shape in the image could be the result of perspective distortion of a simpler shape in the scene, we assume that this is actually the case. Other work at SRI at MIT has dealt with the three-dimensional interpretation of occluding and surface contours. For example, given a curve in the image, we might assume that it arises from a space curve of the least possible curvature. Constraints of these types, and others still be to formulated, often yield unambiguous three-dimensional interpretations of the surfaces that appear in images. Most of this work has as yet been applied only to idealized line drawings, but some parts of it have been successfully applied to noisy real-world images.

The inference of information about the surface in a scene from the shapes of edges in an image is known as "shape from contour", or sometimes as "shape from shape" (i.e., 3D shape from 2D shape). A closely related problem is that of inferring surface shape from textural variations in the image. Gibson pointed out over 30 years ago that changes in texture coarseness arise from changes in range; thus it should be possible to infer changes in range from changes in coarseness. Kender at Carnegie-Mellon and Witkin at MIT, among others, have demonstrated that the 3D orientation of a surface can be inferred from the anisotropy in its texture; note that here again, as in the case of shape from contour, we are assuming that if the anisotropy could have arisen from perspective distortion, then it actually did. For a review of texture measures that might be used in inferring "shape from texture", see Section 6. To obtain good results, one should use edge-based or primitive-based, rather than pixel-based, texture descriptors; the richer the descriptors, the more likely are the inferences to be reliable.

In the absence of discriminable features, surface orientation in the scene can also be inferred from intensity variations ("shading") in the image. The pioneering work on the inference of "shape from shading" was done by Horn at MIT, and the work has been continued by a number of other MIT students. Given the position of the (small, distant) light source and the surface reflectance function, surface shape is still not unambiguously determined, but it is strongly constrained, and can be estimated based on additional information, such as the shapes of surface contours, the positions of occluding contours, the restriction that the surface is a surface of revolution, or the requirement that the surface curvature be as uniform as possible. Surface shape becomes unambiguous if we are given several images taken from the same position, but with light sources in different positions; this "photometric stereo" approach has been investigated by Woodham and others. Much of this work has assumed diffuse reflectance, and needs to be extended to reflectance functions that have strong specular components; in such cases the shapes of highlights may provide additional information.

8. Object representation

Digital images are 2D arrays in which each pixel's value gives the intensity (in one or more spectral bands) of the radiation received by the sensor from the scene in a given direction. Other viewer-centered representations of the scene are also conveniently represented in array form, with the value of a "pixel" representing illumination, reflectivity, range, or components of surface slant at the scene point located along a given direction. Various types of image transforms, as well as symbolic "overlay" images defining the locations of features (contours, curves, etc.) or regions, are other examples of 2D arrays that are often used in image processing.

Features and regions in an image can also be represented in other ways which are usually more compact than the overlay array representation and which also may make it easier to extract various types of information about their shapes. The following representations are all two-dimensional, and are appropriate only if 3D shape information is not known. One classical approach is to represent regions by border codes, defining the sequence of moves from neighbor to neighbor that must be made in order to circumnavigate the border; curves can also be represented by such move sequences ("chain codes"), which were introduced in 1960 by Freeman (now at Rensselaer

Polytechnic Institute). Another standard way of representing regions, studied at about the same time by Blum (now at NIH), is as unions of maximal "blocks" contained in them - e.g., maximal "runs" of region points on each row of the image, or maximal upright squares contained in the region; the set of run lengths on each row, or the set of centers and radii of the squares (known as the "medial axis") completely determines the region. The square centers tend to lie on a set of arcs or curves that constitute the "skeleton" of the region; if we specify each such arc by a chain code, and also specify a radius function along the arc, we have a representation of the region as a union of "generalized ribbons".

There has been recent interest in the use of hierarchically structured representations that incorporate both coarse and fine information about a region or feature. One often used hierarchical maximal-block representation is based on recursive subdivision into quadrants, where the blocks can be represented by the nodes of a degree-4 tree (a "quadtree"). This representation has been extensively studied by Samet at the University of Maryland. A hierarchical border or curve representation based on recursive polygonal approximation, with the segments represented by the nodes of a "strip tree", was recently introduced by Ballard at the University of Rochester, while a border or curve representation based on quadrant subdivision has been studied by Shneier at the University of Maryland.

At a higher level of abstraction, a segmented image is often represented by a graph in which the nodes correspond to regions (or parts of surfaces, if 3D information is available) or features, labelled with property names or values, and the arcs are labelled with relation values or names. A problem with this type of representation is that it does not preserve the details of region geometry, and so can only provide simplified information about geometrical properties and relations, many of which have no simple characterizations. An ideal representation should provide information at multiple resolution, so that both gross geometry and important local features are easily available, together with the topological and locational constraints on the features' positions, where these constraints may have varying degrees of fuzziness. It should also be easy to modify the representation to reflect the effects of 3D geometrical transformations, so that representations of objects viewed from different positions can be easily compared.

Representations of surfaces and objects, i.e., "2½-dimensional" and "3-dimensional" scene representations, are also an important area of study. The visible surfaces in a scene can be represented by an array of slope vectors; the histogram of these vectors is known as the "gradient space" map. The range to each point in the scene is another important type of viewer-centered array representation.

In order to identify the objects in a scene, it is desirable to relate the viewer-centerd representations of the visible surfaces to object-centered representations that describe the objects on a three-dimensional level. A variety of object representations can be defined, generalizing the representations of two-dimensional regions described above. An object can be represented by a series of slices, and a 2D representation can be used for each slice. Alternatively, an object can be represented as a union of maximal blocks - e.g., by an "octree" (based on recursive subdivision of space into octants) or by a 3D "medial axis". If this axis is approximated by a set of space curves, each represented by a 3D chain code, and we also specify a radius function along each curve, we have a representation of the object as a union of "generalized cylinders" or "generalized cones"; this representation has been extensively studied by Binford and his students at Stanford, as well as by Marr at MIT.

9. Model matching

The image analysis processes described up to now give rise to a decomposition of the image into regions, or of the scene into objects. A "literal" description of the image or scene can thus be given in the form of a relational structure in which the nodes correspond to features, regions, or objects, labelled by lists of their property values (shape, texture, color, etc.) and the arcs correspond to relations (adjacency, relative position, etc.). However, this type of "semantics-free" description is usually not what is wanted; rather, one wants a description in terms of a known configuration of known objects. This requires "recognizing" the objects by comparing their descriptions to stored "models", which are generalized descriptions defining object classes.

Even in two dimensions, such models are often very difficult to formulate, since the constraints on the allowable property values and relationships are hard to define. In three dimensions, the problem is rendered even more difficult by the fact that only one side of an object can be visible in an image; the image description is two-dimensional, while the stored object models are presumably three-dimensional, object-centered representations.

The most extensive work on recognition of three-dimensional objects from images is embodied in the ACRONYM system, developed by Binford and his students at Stanford. This system incorporates methods of predicting the two-dimensional appearance (shape, shading, etc.) of a given object in an image taken from a given point of view. Conversely, it provides means of defining constraints on the three-dimensional properties of the object that could give rise to a given image, and for manipulating sets of such constraints. These capabilities are incorporated in a prediction/verification process which uses the image to make predictions about the object, and verifies that the image could in fact have arisen from an object that satisfies the resulting set of constraints. Thus far, ACRONYM has been implemented only in restricted domains, but it is based on very general principles, and should be widely extendable.

It is often appropriate to model regions or objects hierarchically, i.e., as composed of parts arranged in particular ways, where the parts themselves are arrangements of subparts, and so on. There is an analogy between this type of hierarchical representation and the use of grammars to define languages; here a sentence is composed of phrases which are in turn composed of clauses, etc. Based on this observation, the process of recognizing an object as belonging to a given hierarchically defined class of objects is analogous to the process of recognizing a well-formed sentence as belonging to a given language, by parsing it with respect to a grammar for that language. This "syntactic" approach to object (or pattern) recognition has been extensively studied by Fu and his students at Purdue University. It has been used successfully for recognition of two-dimensional shapes, patterns, and textures; but it is less appropriate for three-dimensional object recognition, since it is not obvious how to incorporate in it mechanisms for relating 2D images to 3D objects.

Many difficult problems are associated with the model matching task. It is not trivial to define models for given classes of patterns or objects. (In the case of syntactic models, the problem of inferring them from sets of examples is known as grammatical inference. The pioneering work on the inference of relational structure models from examples was done by Winston at MIT in 1970.) Given a large set of models, it is not obvious how to determine the right one(s) with which to compare a given object; this is known as the indexing problem. Even if the correct model is known, comparing it with the descriptions of a given object may involve combinatorial search. (Here, however, relaxation or constraint satisfaction methods can often be used to reduce the search space.) The best approach is to use the model(s) to control the image analysis process, and to design this process in such a way that most of the possible models are eliminated at early stages of the analysis. Unfortunately, there exists as yet no general theory of how to design image analysis processes based on given sets of models; the control structures used in computer vision systems have been designed largely on heuristic grounds.

10. Some general issues

We have seen that the computer vision process involves, in general, many different processes that incorporate many different types of information about the class of images being analyzed. There is no general theory of control in computer vision; in other words, there are no general principles that specify how these processes should interact in carrying out a given task. In particular, when a number of methods exist for performing a given task, e.g., feature detection or inference of surface orientation, it would usually be desirable to implement several of the methods in order to obtain a consensus; but there is no general theory of how to combine evidence from multiple sources.

Most of the successful applications of computer vision have involved relatively simple domains, and have been primarily two-dimensional. For example, in robot vision, systems that recognize parts on a belt (well illuminated, non-overlapping, in specific 3D orientations) are not hard to build, but systems that recognize parts in a bin (shadowed, overlapping, arbitrarily oriented) are still a research issue. Techniques exist that will in principle handle such complex situations, but they need to be refined and extensively tested before they can be used in practice.

The discussion of the computer vision process in this article has been quite general-purpose, without reference to particular domains of application. It is also possible to build "specialist" or "expert" vision systems tailored to a specific domain, which make use of methods especially designed for that domain. From a practical standpoint, successful applications of computer vision are likely to be of this specialized nature. It is the general approach, however, that makes computer vision at least potentially a science, and that will continue to provide a theoretical background for the design of application-oriented systems.

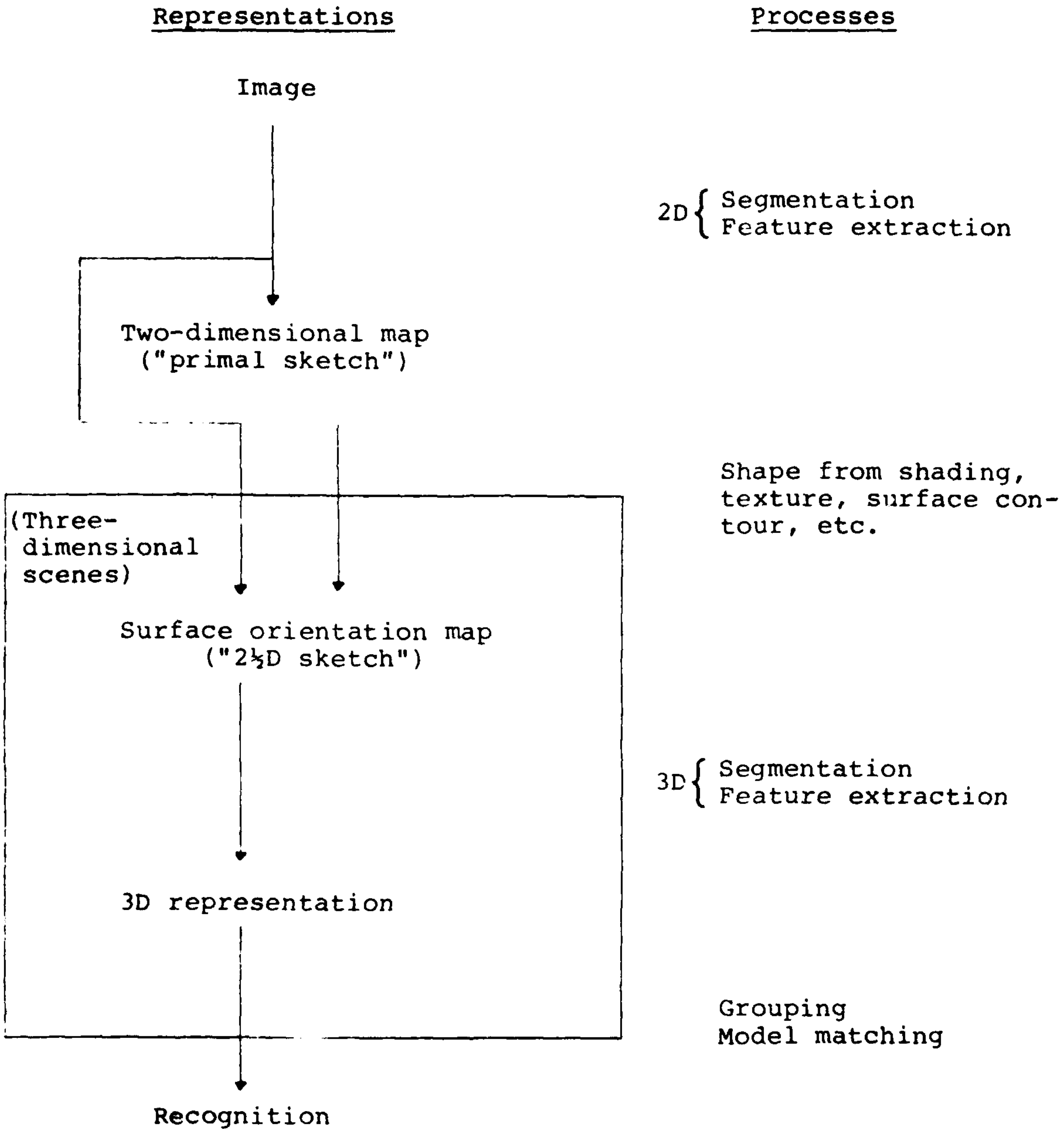

Figure 1. Simplified diagram of a computer vision system.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. A128 570

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
COMPUTER VISION

5. TYPE OF REPORT & PERIOD COVERED
Technical

6. PERFORMING ORG. REPORT NUMBER
TR-1157

7. AUTHOR(s)
Azriel Rosenfeld

8. CONTRACT OR GRANT NUMBER(s)
DAAG-53-76C-0138

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Army Night Vision Lab.
Ft. Belvoir, VA 22060

12. REPORT DATE
April 1982

13. NUMBER OF PAGES
38

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer vision
Image analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer vision deals with the analysis and interpretation of images. Over the past 25 years, many ad hoc techniques for analyzing images have been developed, but during the past few years this technology has begun to develop a scientific basis. This paper outlines the major components of a general computer vision system, and summarizes the state of the art in each of these components.