Image Segmentation and Feature Extraction

JACK SKLANSKY, SENIOR MEMBER, IEEE

Abstract—**An overview of present computer techniques of partitioning continuous-tone images into meaningful segments and of characterizing these segments by sets of "features" is presented. Segmentation often consists of two methods: boundary detection and texture analysis. Both of these are discussed. The design of the segmenter and feature extractor are intimately related to the design of the rest of the image analysis system—particularly the preprocessor and the classifier. Toward aiding this design, a few guidelines and illustrative examples are included.**

I. Introduction

IMAGE segmentation and feature extraction are two major components of modern computerized image analysis or "machine vision." This paper provides an overview of these two components.

A computerized image analyzer usually consists of many—sometimes all—of the following major components: 1) source of radiant energy or illumination, 2) scene, 3) sensor, 4) scanner, 5) digitizer, 6) preprocessor, 7) boundary detector, 8) texture analyzer, 9) feature extractor, 10) classifier, 11) knowledge bank, 12) summarizer, 13) adviser, 14) questioner, 15) display, 16) interactive controller, 17) human user (last in list, but first in importance). These components are usually organized approximately as indicated in Fig. 1. Among these components the texture analyzer and boundary detector together form the segmenter, which identifies meaningful connected components ("segments") of the picture. The feature extractor computes a set of descriptors which facilitate classifying and labeling the segments into categories. These categories are used subsequently in the generation of a summary or description of the scene.

In addition to discussing the segmenter we shall describe several of the types of features that have proved successful for labeling the segments in various applications. (Sometimes—as in the case of Fourier harmonics of intrinsic equations [58] of boundaries—the features are also used for improving the segmentation process.) We shall omit discussion of the selection of a best subset of features from a larger set—a subject for which there is a substantial literature and which would take us beyond the intended scope of this paper.

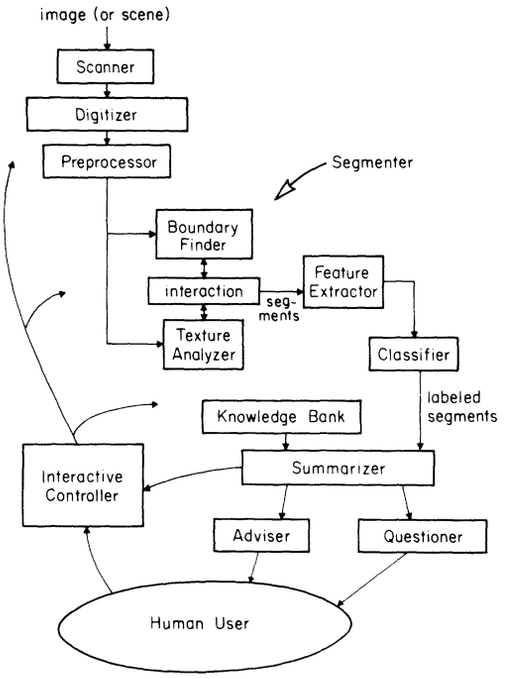

Fig. 1. Organization of computerized image analyzer.

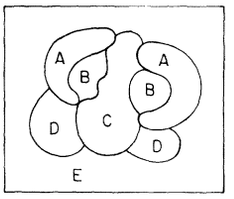

Fig. 2. Segmentation of picture.

Fig. 2 illustrates a possible segmentation of a scene. Each segment of the scene consists of a boundary, the connected region enclosed by the boundary, and a distinct label.

The process of feature extraction associates with each segment a vector of primitive descriptors or "features" (or "characteristics" or "measurements" or "properties"). For example, the segment A might be associated with a vector of the form

$$a = (a_1, \cdots, a_m).$$

Part of the field of computerized image analysis is concerned with the modeling or reconstruction of three-dimensional scenes, and hence, with the segmentation of three-dimensional space. Examples of this activity are the analysis of the "blocks world" [9] and computerized tomo-

Manuscript received February 21, 1977; revised September 9, 1977. This work was supported by the National Institute of General Medical Sciences of the U.S. Public Health Service under Grant No. GM-17632. This paper is a revision of a seminar presented at the NATO Advanced Study Institute on Digital Image Processing and Analysis, Bonas, France, June 22, 1976.

The author is with the School of Engineering, University of California, Irvine, CA 92717.

0018-9472/78/0400-0237$00.75

graphy in medical radiography [15]. The continued growth of these activities most likely will lead to increased interest in the segmentation of 3-space. Even higher dimensional spaces may subsequently be viewed in this manner, in order to account for gray level and time in the observed images. One also can view finding clusters of feature vectors in d-space [13] as a segmentation of a d-dimensional "image."

The design of the segmenter and feature extractor are intimately related to the design of the rest of the image analysis system—particularly the preprocessor and the classifier. Effective, precisely described general routines for tying these components together have not yet been achieved. In this paper, steps toward this end are attempted through the use of a few guidelines and illustrative examples.

II. Guidelines

We list here a few guidelines which we have followed in our own work, and which may be of use to others. We are receptive to suggestions for modifications and additions to this list.

Know Your User!

1. It is sometimes noted that happiness consists of a reduction of frustrations. Hence spend enough time with your user to discover his frustrations, and search for ways that your system may reduce these frustrations.
2. Assimilate into your image analysis system as much as possible of the user's present knowledge base. You may also wish to computerize a selected subset of his present techniques.

Know How Your Image Was Formed!

The variation in gray level across the boundary of a segment depends very strongly on the original object, the optical system, the film, the scanner, and the digitization process. These components will affect the design of your boundary detector as well as your texture analyzer.

Pay Careful Attention to Preprocessing!

Preprocessing, such as linear filtering or histogram transformation, has significant effects on the performance of segmenters and feature extractors.

Don't Give Up; Hardware May Save You!

Digital image analysis is a hungry devourer of data—enough to make extreme demands on the image acquisition, the storage, and the display capabilities of most digital image-processing facilities. Many such facilities are built around one or two dedicated minicomputers. Under these conditions the computation time and the cost of your system may be excessive. Because of the exciting new developments in photosensitive devices, large-scale integrated digital electronics, and various forms of digital computer memories and display devices, it is wise to maintain close contact with these developments. In time, they may help you build a cost-effective system.

III. The Image

We define an *image* as a nonnegative scalar function $i(\boldsymbol{x})$, where $\boldsymbol{x}$ is a two-dimensional vector. A frequent physical embodiment of $i(\boldsymbol{x})$ is a light intensity in an "image plane" $\mathcal{X}$. A *digital image* is an image that is defined only at a specified set of points in $\mathcal{X}$—usually integer-valued points forming a rectangular array.

In many practical situations one must account for two, three, or more spectral components or colors of the image. In such cases, the scalar $i(\boldsymbol{x})$ must be replaced by a vector $\boldsymbol{i}(\boldsymbol{x})$, where each component of $\boldsymbol{i}(\boldsymbol{x})$ is either a spectral component (a color) or a function of these components. For ordinary full-color visual scenes, $\boldsymbol{i}(\boldsymbol{x})$ often consists of the following three components: hue, saturation, and brightness. These three components are linear functions of the three primary colors: red, green, and blue [7], [59]. Other primary "colors" are used in special applications, such as satellite imagery, blood cell microscopy, and nuclear medicine.

In the remainder of this paper, we assume that the digitized image is a scalar function of $\boldsymbol{x}$.

IV. Preprocessing

There are four major types of preprocessing operations: gray-scale conversion, linear filtering, geometric equalization, and histogram transformations. These operations, when properly designed, can contribute significantly to the effectiveness of segmentation and feature extraction.

Gray-Scale Conversion

Suppose we are given an image $i(x)$. Gray-scale conversion usually converts $i(\boldsymbol{x})$ into $p(\boldsymbol{x})$, where

$$p(\boldsymbol{x}) = \log i(\boldsymbol{x}).$$

This conversion has the following benefit: differences of adjacent pairs of values of $p(\boldsymbol{x})$ are approximately proportional to subjective contrast. Thus a uniformly increasing step wedge in $p(\boldsymbol{x})$ corresponds to an exponentially increasing step wedge in $i(\boldsymbol{x})$, which in turn is observed subjectively as a uniformly increasing step wedge.

If the recorded image is $q(\boldsymbol{x})$, and $q(\boldsymbol{x})$ is a monotonically increasing function of $i(\boldsymbol{x})$, let us say

$$q(\boldsymbol{x}) = \Phi[i(\boldsymbol{x})],$$

then one would usually transform $\Phi[i(\boldsymbol{x})]$ to $p(\boldsymbol{x}) = \log i(\boldsymbol{x})$ by the transformation

$$p(\boldsymbol{x}) = \log \Phi^{-1}[q(\boldsymbol{x})].$$

In many digital image processing systems, the gray-scale conversion is carried out in an analog device such as a logarithmic amplifier.

Linear Digital Filtering

Linear digital filtering usually consists of sampling and quantizing, followed by digital convolutions. Digital convolutions are sometimes implemented by digital Fourier transforms. In many applications this filtering must take place at adjustable spatial and/or gray-level resolutions as well as

adjustable positions and sizes of the portion of the picture under analysis.

Linear filtering can accomplish the following [45]: a) suppress low-frequency noise, such as slowly varying illumination; b) suppress narrow-band noise, such as is sometimes produced by aberrations in electronic communications equipment; c) suppress high-frequency noise, such as dust on film and/or electronic shot noise; d) normalize size of picture by a local averaging operation or by a uniform sampling of a fraction of the pixels; e) compensate for linear aberrations in the film exposure and development; f) adjust the dynamic range of the picture in the display; and g) enhance the visibility of details for human viewing of the displayed picture.

Geometric Equalization

The relation among film, optics, and scene often introduces so-called photogrammetric distortions that can be corrected by well-understood mathematical formulas. These corrections usually produce an orthogonal projection of the scene onto the image plane.

Electronic television circuits often introduce uneven gray-level or spatial distortions. The gray-level distortions are forms of "shading." The spatial distortion is an uneven stretching or "barreling."

It is possible, and often desirable, to remove photogrammetric distortions, shading and barreling—and other significant geometric distortions—by algorithms constructed especially for these purposes [3]. We refer to these algorithms as "geometric equalizers."

Histogram Transformations

The histogram of the gray levels of a picture or of a section of a picture is often used a) for edge or boundary detection, b) for extraction of textural features, and c) as a guide to transforming the gray levels to new levels to facilitate the display of the image. The latter transformation results in a new histogram—a histogram of the transformed gray levels—that may improve the computer's ability to carry out a) and b).

In several reports and papers discussing histogram transformations it is suggested that flat or constant histograms are best for viewing displayed images and for subsequent feature extraction and segmentation. We have found that this concept may be misused if one is not aware of a certain pitfall. Before indicating what this pitfall is, we present our view of why histogram flattening or "equalization" often yields visually pleasing results in the displayed image.

Let g denote an arbitrary gray level, and let b denote an arbitrary interval width. Usually b is restricted to small values with respect to g. Let N denote the number of pixels whose gray values lie in the interval $(g - (b/2), g + (b/2))$. Since most medical images have large low-frequency Fourier harmonics, the gray level at any pixel is in most instances close to the gray levels at neighboring pixels. Consequently, when b is small, the N pixels whose gray levels lie in $(g - (b/2), g + (b/2))$ form just a few connected sets of points or "components" in the x-plane.

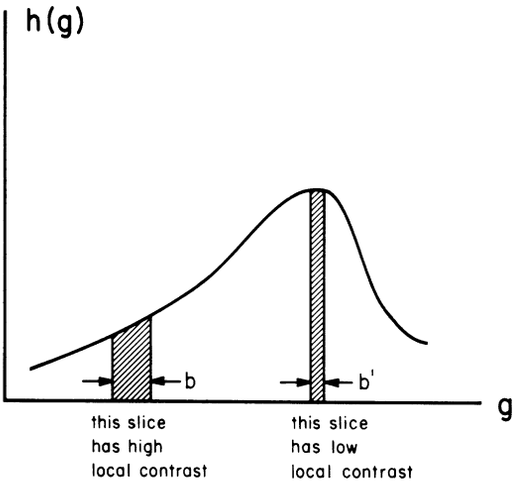

Fig. 3. Relation of local contrast to $h(g)$.

The segments of $p(x)$ determined by these components constitute an N-pixel *slice* of $p(x)$. Consider a set of N-pixel slices of $p(x)$. In some textural pictures (i.e., pictures consisting of large regions of nearly uniform textures) the sizes, shapes, and distributions of the components among these slices are similar. Under this condition a measure of the visibility of details among these slices is the variation of gray level—or local contrast—within each slice. This variation is approximately proportional to the interval width b. Let $h(g)$ denote the histogram of g, i.e., the frequency of occurrence of g in the picture. When b is small, and the histogram $h(g)$ can be approximated as a continuous function; b and $h(g)$ are related as follows:

$$N = \int_{g-(b/2)}^{g+(b/2)} h(y)\, dy \cong bh(g).$$

Hence

$$b \cong \frac{N}{h(g)}. \tag{1}$$

Thus, under the assumption of low-frequency dominance, the relative visibility of details at gray level g is approximately inversely proportional to $h(g)$. This relation of the local contrast to $h(g)$ is illustrated in Fig. 3.

If all portions of a picture function $p(x)$ are equally important, it would be desirable to achieve equal visibility—i.e., equal local contrast, in all N-pixel slices of $p(x)$. This leads to a histogram transformation $\hat{g} = K(g)$, $K(g)$ monotonically increasing, such that the histogram of $\hat{g}$ is

$$l(\hat{g}) = \begin{cases} \text{constant}, & \text{for } a \le g \le c \\ 0, & \text{elsewhere} \end{cases}$$

where $[a,c]$ is chosen to span a typical dynamic range of the human visual system, taking into account the light-emitting or light-reflecting properties of the display device. A transformation $K(g)$ that yields a rectangular or "flat" histogram, as indicated in the above equation, is referred to as a *histogram equalization.*

Now we indicate the pitfall mentioned earlier. In several

papers on digital image processing, it is suggested that flat or rectangular histograms are best for viewing the images and/or for subsequent computerized analysis of the image [18], [19], [57]. We have found that this concept may be misused. For example, by this concept, small amplitude short edges at infrequently occurring gray levels would be given increased contrast. However, the increased local contrast may yield only an apparent increase in the visibility of these edges because it may be achieved by a merging of gray-level quantization cells.

Most images are dominated by low frequencies that carry little information about the scene. These low frequencies consume a large range of gray-level quantization cells with little benefit to the viewer. Hence *before* any histogram transformations are carried out it is useful to suppress (but not eliminate) the low spatial harmonics by linear filtering on high-resolution data (high resolution in both image space and gray level). Subsequent histogram transformation can achieve: a) a reduction in the computational effort expended on extracting textural features, b) a normalization of some textural features, and c) a near-optimal range of gray levels for visual display. (The final transformed data may be reduced in both gray-level resolution and spatial resolution with little harm to subsequent image analysis.)

V. Global Analysis

In some applications the first step of segmentation is a rough analysis of the entire picture. This analysis is designed to answer global questions such as: a) Is this picture a microphotograph of a blood smear, is it an aerial view of the earth, is it a chest radiograph, or is it none of the preceding? b) Is the spatial and/or gray-level resolution of this picture adequate for a first segmentation or for a preliminary analysis of texture?

Two interesting examples of global analysis are a) split-and-merge on the thresholded gray levels for preliminary categorization of the picture [27], and b) variable-span edge detection for preliminary determination of the resolution needed for computing textural coarseness and/or merging strokes into streaks [51], [56].

Another use of global analysis occurs in the detection of landmarks to guide the process of boundary detection. An example of this is Chien and Fu's use of a spatially dependent arrangement of templates for detecting six "key points" on the boundary of the radiographic image of a lung [5]. These six points provide an effective means of imposing global constraints on the subsequently detected lung boundary.

VI. Boundary Detection

Boundary detection usually consists of a sequence of steps, starting from the detection of edge elements or "strokes," eliminating false strokes, merging the strokes into "streaks," eliminating false streaks, combining the streaks into boundaries, and eliminating false boundaries. (Alternative methods of boundary detection are based significantly on texture analysis, especially gray-level statistics. We defer discussion of these techniques to Section VIII.)

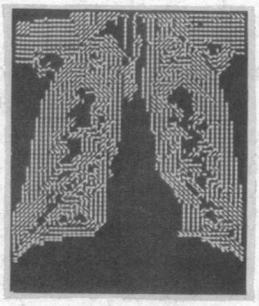

Fig. 4. Vector edge elements of chest radiograph.

Edge Element Finder

There are two major types of edge elements in use today:

a) nondirectional,
b) directional.

Examples of nondirectional edge elements are the magnitude of a digital approximation of the gradient, and the magnitude of a digital approximation of the Laplacian. (Often the term "gradient" or "digital gradient" is used when "digital approximation of the gradient" is understood. Similarly, for "Laplacian.")

Directional edge elements are often referred to as "vector edge elements." In one convention the vector edge element points in the direction of the gradient, while in another convention it points in a direction 90 degrees clockwise from the direction of the gradient [2], [24], [43], [71]. We prefer the latter convention, because it leads to a better visualization of the edges of the object when the vector edge elements of the object are displayed. This visualization is illustrated in Fig. 4, which displays the array of vector edge elements of a chest radiograph. Here the outlines of the lungs are clearly visible.

Successful uses of many forms of digital gradients and digital Laplacians have been reported [2], [49], [58], [71]. Among them we have found the following particularly effective.

1) Digital Gradient: The horizontal component of the digital gradient is

$$\nabla_1 p(x) = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \circledast\ p(x),$$

where $p(x)$ denotes the digitized image received by the digital gradient operator, and $\circledast$ denotes digital convolution. Digital convolution is defined as follows:

$$q(x) \circledast p(x) = \sum_y p(y)q(x-y),$$

where the summation is over all admissible position vectors in the image plane.

The vertical component of the digital gradient is

$$\nabla_2 p(x) = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} \circledast\ p(x).$$

2) *Laplacian:*

$$\nabla^2 p(x) = \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} \circledast\ p(x).$$

In both the digital gradient and the digital Laplacian, one must often adjust the span to fit the shape of the gray-level cross section of the edge. For example, for a span of $s = 2$, $\nabla_1 p(x)$ becomes

$$\nabla_1 p(x) = \begin{array}{|c|c|c|c|c|c|}\hline -1 & -1 & 0 & 0 & 1 & 1 \\ \hline -1 & -1 & 0 & 0 & 1 & 1 \\ \hline -2 & -2 & 0 & 0 & 2 & 2 \\ \hline -2 & -2 & 0 & 0 & 2 & 2 \\ \hline -1 & -1 & 0 & 0 & 1 & 1 \\ \hline -1 & -1 & 0 & 0 & 1 & 1 \\ \hline \end{array} \circledast\ p(x).$$

(*Note:* Normalizing multiplying factors are omitted in all of the above formulas for the digital gradient and digital Laplacian.)

To find the proper span, we suggest starting with a maximum span that is expected to be possibly effective (corresponding to the smallest expected useful spatial resolution) and check whether the edge elements obtained are sufficient in number and in agreement (or coherence) to yield streaks. An example of an application of this type of analysis appears in [56], where the distribution of edge elements is used for estimating the coarseness of a pictorial texture.

Under some conditions—particularly when the gray-level cross section across a boundary so suggests—the Laplacian and gradient may be combined to yield a more accurate computation of edge elements or vector edge elements [71].

Let

$$\nabla p(x) = [\nabla_1 p(x), \nabla_2 p(x)]$$

$$= \text{vector gradient}$$

$$M(x) = \begin{cases} 1, & \text{if } |\nabla p(x)| > \theta_G \text{ and } |\nabla^2 p(x)| > \theta_L \\ 0, & \text{otherwise,} \end{cases}$$

where $|\nabla p(x)|$ = length of $\nabla p(x)$, and θ_G and θ_L are chosen empirically.

Another successful approach to finding edge elements is Hueckel's algorithm [28], which finds the least square error approximation of an ideal edge (defined as a step) inside a circular neighborhood of each pixel. Thus Hueckel's algorithm computes both the size and direction of the edge element, under the assumption that the ideal edge's gray-level cross section is a step.

Streak Finder

A streak finder (or "edge binder") combines the strokes or edge elements into "streaks," i.e., relatively long sections of boundaries. These streaks are formed by combining strokes that are approximately collinear or slowly curving and which lie on curves that satisfy *a priori* constraints dictated by the class of pictures under analysis. The use of variable resolution analysis is often effective in streak finding.

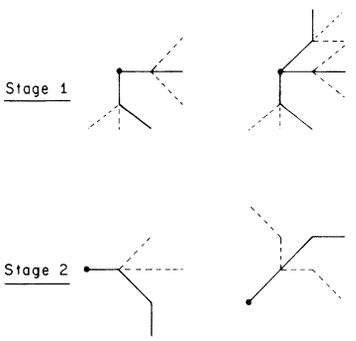

Fig. 5. Finding streaks by dynamic programming.

Three basic techniques have been used effectively in streak finding: heuristic search, relaxation, and template matching.

Heuristic search [38], [39] is obtained by pruning a search for an edge by a) using a figure of merit or "score" at each node of the search tree, and b) a guided "plan" from earlier phases of boundary detection. The use of an *exhaustive* search in conjunction with an additive score results in a form of dynamic programming [2], [42].

Fig. 5 illustrates how streaks are formed at successive stages of a dynamic programming search. At Stage 1, optimum edge elements of length 2 are formed by finding the best annexation of a length-1 edge element to each length-1 edge element. At Stage 2, the optimum edge elements of length 3 are formed by finding the best annexation of a length-2 edge element to each edge element of length 1. In Fig. 5, the rejected edge elements are shown dotted, and the accepted elements are shown as solid lines.

The criterion of "best" or "optimum" is established by a recursive equation of the form

$$f(x, l_i + l_j) = \max_k [f(x,l_i) + f(x_k,l_j) - q(x,x_k)]$$

where

x	initial point of arc,
x_k	terminal point of arc,
$q(x,x_k)$	monotonically increasing function of digital estimate of integrated curvature of optimum arc from x to x_k,
l_i	length of arcs at Stage i,
$f(x,l_i)$	figure of merit of arcs of length l_i starting at point x.

In some cases one can use the above equation to find the best edge sequence of length 4 by annexing the best length-2 edge elements to the length-2 edge elements found in Stage 1. In this way the lengths of the building blocks for constructing streaks can grow exponentially [2].

The iteration procedures in heuristic search are computationally similar to "relaxation" methods. In typical relaxation methods, the probability that a candidate edge element is a true edge element is reestimated at each iteration [54], [36], [53].

A third effective approach to streak finding is guided template matching, matched filtering, and/or Hough detection of collinearity, all of which are closely related [31], [41], [51], [60]. This procedure fills each element of an array of accumulators in digitized parameter space with increments of 1 according to whether a curve with the parameters specified by the accumulator is likely to pass through a given pixel. This algorithm, in contrast to the heuristic search technique, is parallel in nature and leads to fast algorithms.

Once a streak is found by one of the above procedures, the streak may act as the medial axis of a ribbon that functions as a plan for high-resolution curve following [30]. Applications of this concept have been applied to computer-aided radiography [2], [35], [71].

Once the streaks have been found the streaks may be combined into final boundaries of objects at high resolution. An example of this procedure appears in the combining of dorsal and ventral portions of a rib contour to form a full rib contour [71]. Another example of this procedure is the combining of streaks to form the outlines of blocks [61].

Once the boundaries of the objects have been found, a detailed analysis of the texture within these objects may serve to provide further corrections of the computed boundaries. Such an approach was developed recently for terrain analysis [26].

VII. Variable Resolution Analysis

Effective boundary detection and texture analysis (the latter is discussed in Section VIII) often involve hierarchical multistage procedures in which the gray-level and spatial resolutions are optimized at each stage. We refer to this as a "variable resolution analysis."

Another example of variable-resolution analysis occurs in the use of polar coordinates. For example, to normalize an object with respect to rotation and magnification, it may be convenient to use the polar-coordinate autocorrelation [8]

$$\gamma(s,\phi) = \iint r\, dr\, d\theta p(r,\Theta)p(sr, \Theta + \Phi).$$

Nonuniform resolution analysis may also occur in three-dimensional space, as in computerized tomography [15].

The effect of a hierarchical variable resolution analysis of an image is equivalent to an analysis on a variable resolution retina, with finer resolution occurring where the image has greater importance or interest. (The effect of variable resolution gray levels may be accounted for by adding a third dimension and using three-dimensional mosaics.) This view of scene analysis led us to the development of a theory of variable-resolution digital vision, based on the use of nonuniform mosaics as artificial retinas [64], [65]. For these nonuniform mosaics, we showed that *minimum-perimeter polygons* can be effective, economical representations of line drawings, silhouettes, or "blobs." A minimum-perimeter polygon (MPP) of a given digitized image is defined as a shortest perimeter polygonal blob whose digitization yields the given image.

An example of the use of MPP's for segmentation is the problem of separating overlapping digitized images of disjoint blobs on a nonuniform mosaic. More specifically, suppose the mosaic consists of nonuniformly spaced dots, as illustrated in Fig. 6(a). First the dots are used as nuclei to generate an array of nonoverlapping convex cells that cover the plane. For this purpose, Fam suggests finding the mosaic produced by the skeleton of the dots [11], [65]. He refers to this as the *growth mosaic* of the dots. The MPP of the growth mosaic, constrained to include the dots as "sensors" in the digitization process, is illustrated in Fig. 6(b). Next, the neck of the MPP is found—for example, by a guided sequence of measurements of distances between pairs of vertices (lengths of chords). If the ratio of the neck to the largest diameter of the MPP is sufficiently small, the neck is used as a cut set to separate the digitized image into two disjoint images.

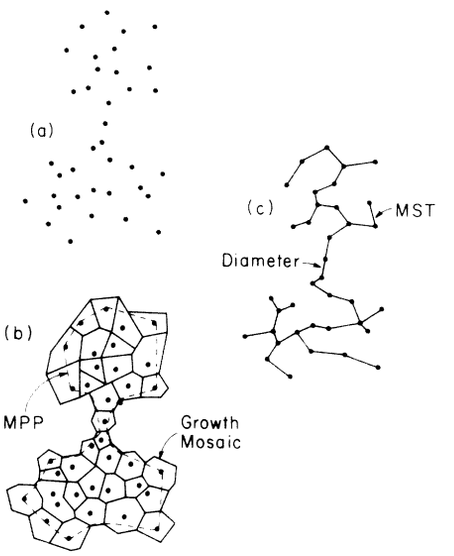

Fig. 6. Separating overlapping blobs on point mosaic.

In an interesting alternative to this approach, one finds the minimum spanning tree (MST) of the given image, as in Fig. 6(c). Then one finds the diameter (a longest path) along the MST and the density of branches along the diameter. (This is illustrated in Fig. 6(c), where the diameter is indicated by the heavy line.) A minimum of this density along the diameter indicates a likely neck [74].

VIII. Texture Analysis

We suggest the following operational definition of "texture." A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic. One may view a textured object as a d-dimensional structure (in image analysis, d is usually 2 or 3) constructed by a machine, or an algorithm whose major parameters remain approximately constant or vary slowly throughout the process of construction.

Among the frequently used texturally based approaches to image segmentation are region growing, region merging, and region splitting [53], [73], [25], [27], [77]. Region growing and region merging are processes in which adjacent regions having similar textures are combined or merged. Region splitting is a process in which the detection of

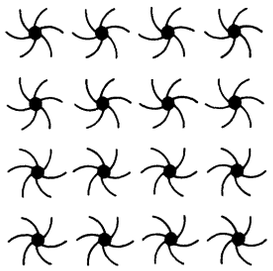

Fig. 7. Two textures.

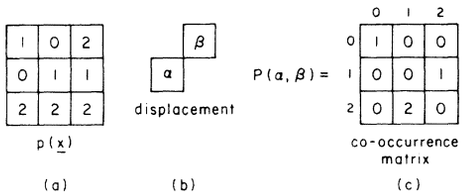

Fig. 8. Co-occurrence matrix.

dissimilarity of texture among two or more neighborhoods within a region results in splitting the region into regions of greater homogeneity. Merging and splitting are often carried out alternately, yielding a so-called "split-and-merge" process [27].

The use of split-and-merge and region growing techniques can yield an effective preliminary segmentation of part, or even all of an image. This procedure may be followed by boundary detection for the final segmentation.

We conjecture that in certain types of medical images, such as radiographs, photographs of the ocular fundus, and blood samples, a disease state may be exhibited at an early date by textural changes that are difficult to detect even by a trained viewer, but which are detectable by a computer. Preliminary evidence supporting this hypothesis has been provided by Bela Julesz. He showed that two-dimensional textures differing only in third and/or higher order statistics are, for practical purposes, indistinguishable to a human viewer (except by careful scrutiny) but can be discriminated by a computer [29]. An example of an enlarged section of a pair of such textures, made up of clockwise and counterclockwise simplified pin wheels, is shown in Fig. 7.

Texture analysis results in a summary, a representation, or a model of a textured region. The simplest form of summary is merely a list of textural features. More complex summaries or models describe relations among features as well as the features themselves. Some models permit filling of gaps in textured regions. Other models provide a means of measuring similarity between textures as well as filling gaps or reproducing textures from a compressed model. All of these models provide a means for compressing the amount of information required to represent the texture.

Computerized extraction of textural features falls into two major categories: structural and statistical. The structural approach falls into two subcategories: a) the use of placement rules or generative grammars (which are permitted to be ambiguous) [4], [76], and b) the use of Fourier harmonics or transform codes to represent the texture [19], [48]. The statistical approach falls into three categories: a) the use of power spectrum or autocorrelation function [23], [34]; b) the use of gray-level statistics [19], [21], [32], [72], e.g., co-occurrence matrices, run-length matrices, and histograms; and c) the use of local feature statistics, e.g., the statistical distribution of corners, the variance of the magnitude of the gradient, etc. [52]. (Category b) may be viewed as a special case of category c).)

Histogram analysis deserves special mention as a form of gray-level statistics. The valley in a bimodal histogram sometimes is an indicator of a gray level that separates two meaningful segments of a picture. Under these circumstances, the computation of this valley may be viewed as a form of edge detector [1], [6], [50].

We have found that certain diagnostically significant textural features in xeromammograms are strongly related to infrequently occurring gray levels in the tails of certain shapes of histograms. Because these gray levels occur infrequently, histogram equalization inhibits rather than enhances the extraction of these features. This is an example of how preprocessing can have a serious effect on the effectiveness of segmentation. (For further discussion, see Section IV.)

The *co-occurrence matrix* (or, equivalently, the *spatial dependence matrix*) is an extension of histogram analysis to higher-order gray-level distributions. A histogram is an estimate of the distribution of gray levels in an image. A co-occurrence matrix is an estimate of the distribution of pairs of gray levels separated spatially by a specified displacement vector.

The co-occurrence matrix is illustrated in Fig. 8. Part (a) of this figure shows a 3×3 digital image $p(\boldsymbol{x})$, with the gray levels distributed over the values 0,1,2. Part (b) shows the relative displacement for the co-occurrence matrix, and part (c) shows the co-occurrence matrix. For example, the value of the co-occurrence matrix at $(\alpha,\beta) = (2,1)$ is 2. This represents the two occurrences of the pair of gray levels (2,1) in $p(\boldsymbol{x})$, with the second member of the pair displaced by one unit rightward and upward with respect to the first member.

The set of co-occurrence matrices for every possible displacement within a given digitized image is a complete representation of the second-order statistics of that image, provided the statistics of the image are spatially stationary. From these matrices one may extract features that summarize the textures of images in a compact manner, while retaining the information needed to discriminate between textures. Examples of these features are the sum of the normalized squares of the elements of the matrix ("angular second moment") and the normalized variance of $\alpha\beta$ ("correlation"). Haralick and his co-workers have applied such features to the segmentation of aerial terrestial scenes [19] and the classification of samples of rock [19], [21].

Co-occurrence matrices may be extended to the representation of third and higher order statistics—as would be needed for discriminating between the two textures in Fig. 7. To represent third-order statistics, one needs a set of three-dimensional co-occurrence arrays for gray levels at the vertices of a specified triangle T. Each element of the array is an estimate of the frequency of occurrence of a triad of gray

levels, with relative spatial displacements specified by T. Representations of nth-order statistics, $n > 3$, require co-occurrence arrays at specified n-gons (n-vertex polygons).

Clearly, for most second- and higher order statistics, the computation of *all* possible co-occurrence arrays—for every displacement, every triangle, etc.—is usually too expensive and time consuming. One must therefore rely on insight and intuitive guessing in order to discover the proper displacements, triangles, and n-gons.

Models of texture that can reproduce the observed textures with some degree of fidelity may be particularly useful in medical radiography, where textured tissue such as bone or breast may be partially or fully obscured by other tissue. A two-dimensional model of such texture may facilitate the recovery of the obscured tissue. Such modeling would be useful for visual displays. It also may help in the subtraction of one texture from another, e.g., the subtraction of the ribs from lung tissue in a chest radiograph, so as to enhance the detectability of objects in the remaining image.

One such approach to modeling texture has been described by McCormick and Jayaramamurthy [40]. In this technique the textured scene is scanned left to right, top to bottom, along a sequence of horizontal parallel lines. The resultant time series is represented by a linear prediction equation. (A similar approach has been used effectively in the modeling of speech [37].)

Another approach to modeling texture is based on the theory of Markov random fields [22]. It is assumed that the gray level of every pixel depends stochastically on the array of gray levels in the neighboring pixels, and on nothing else. We refer to this as a *Markov texture*. One can show that a Markov texture can be fully represented by the distribution of geometrically defined classes of connected components in the contours of constant gray levels of this texture. Fig. 9 illustrates five components of these gray-level contours on a square mosaic. Preliminary information-theoretic evidence indicates that some of these components, e.g., components having small total arc lengths, may be discarded, and still the remaining components will yield an approximate reconstruction of the original texture. (In a sense this approach is a generalization and extended formalization of the use of run-length matrices [72], [14] as descriptors of textures.)

In every analysis of texture one faces the problem of choosing the size of the window or neighborhood within which the textural features are extracted, and choosing the aperture or size of resolution element within each window [2], [53], [65]. Hierarchic approaches to this analysis have achieved some success [2], [20]. In these approaches, both global and textural features are made available simultaneously, thereby permitting both the local and the global features to guide the choice of neighborhoods where additional local features are to be extracted.

IX. Feature Extraction

Feature extraction is a process usually associated with the analysis of segments of a picture. These features are then used in a classifier to label the segments. Features may be used to describe an object in a picture, a collection of objects, or an entire picture. For example, the number of cells in a microscope slide of a blood sample may be the single relevant feature describing that slide.

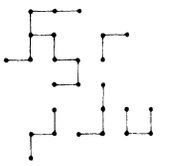

Fig. 9. Gray-level contours on square mosaic.

As we mentioned earlier, features are an elementary form of summary. More advanced and powerful summaries show the relations among features and in some cases may permit approximate reconstructions of the segments under analysis. For summaries consisting only of lists of features, there is a substantial literature on "feature selection" [13]. Feature selection is a process of pruning ineffective subsets of features from an initial set of features to yield a smaller effective set. Most of the techniques for feature selection are based on statistical distance measures [13].

Feature extractors may be categorized as follows:

1) descriptors of boundaries (shape descriptors),
2) descriptors of texture,
3) descriptors of spatial relations (syntactic descriptors).

Shape Descriptors

Shape descriptors are dependent on economical representations of the boundaries. A precise and general purpose representation is provided by the chain code [12]. For smooth low-curvature boundaries, minimum-perimeter polygons or ε-approximation polygons are often more economical than chain codes [69]. It has been shown, for example, that the ratio of the number of vertices of the MPP of a digitized circle of radius r to the number of elements in the chain code of that circle varies inversely as the square root of r [63]. Still further compressions of the boundaries depend on the particular shapes of the segments of the picture. It is convenient to divide these segments into two classes: line-like segments (curves) and blobs.

Line-like segments may be represented by their end points, lengths, angles, intersections, and measures of tortuosity. Relationships among these lines may be represented (probably uneconomically) by an incidence matrix, where the rows and columns are labeled by nodes. Each element of the matrix is 1 or 0 depending on whether or not the corresponding pair of nodes is connected by an arc. The matrix may provide additional information by entering a feature (e.g., the length or tortuosity) of each arc in the appropriate element of the matrix.

A wide range of descriptors have been used for blobs. Among these are the Fourier components of the functions specified by an "intrinsic equation" [58] (slope versus arc length, curvature versus arc length, etc.), the distance skeleton, run tracking, moments, autocorrelation, chord statistics (equivalent to integral geometry) [9], [46], and the distribution of sizes and shapes of the convex hulls and concavities in the MPP [68], [69]. In all of these cases the effectiveness of the features is measured by a) intraclass

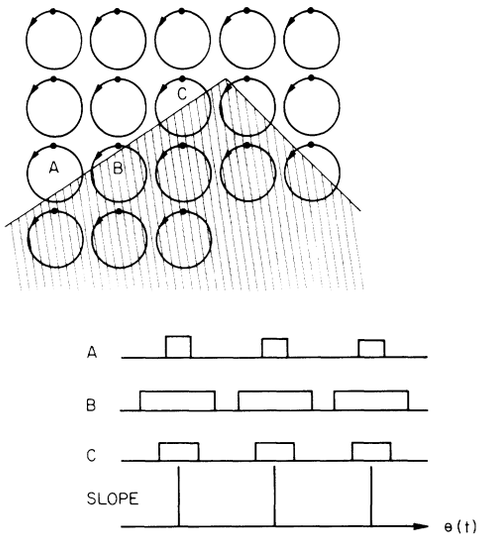

Fig. 10. Mechanism for computing slope density.

invariance, b) interclass sensitivity, c) number of bits required to store the features, d) the cost of extracting the features—in computing time and hardware, and e) the reliability and/or accuracy of the feature extraction process.

Among the above methods of feature extraction the methods of chord statistics is perhaps least used. Yet in one application, the recognition of aircraft silhouettes, a chord statistic technique was modified to a form that made it suitable for an economical hardware implementation [66].

This implementation was based on the use of an optical mechanism that translates the image of the silhouette along a small circular path—a "nutation"—on a fixed array of photosensors, as illustrated in Fig. 10 [44], [66], [67]. Each rectangular pulse formed by these detectors is converted into a narrow impulse of fixed amplitude positioned at the centroid of the original pulse. One can show that the time of occurrence of each impulse, measured from the origin of the nutation period, is approximately proportional to the slope of the silhouette where the circular orbit meets the silhouette.

The impulses are added, yielding an approximate histogram $f(\Theta)$ of the slopes of the silhouette, where the time axis relative to the origin of the nutation period is proportional to the slope Θ. The function $f(\Theta)$ is entered into a bank of filters to compute the Fourier harmonics of the slope density. These harmonics are an economical representation of the blob, and can provide a full reconstruction of the blob when the blob is convex. Eliminating all the harmonics above a specified frequency yields a least mean-square-error approximation of the true slope density while retaining closure of the resultant silhouette. The slope density can be normalized with respect to both rotation and magnification of the silhouette (it is inherently independent of translation), by dividing $f(\Theta)$ by the area under $f(\Theta)$ and by setting the phase of the second harmonic equal to zero. (For closure, it is necessary that the first harmonic is zero.) With this approach it was possible to obtain effective discrimination among silhouettes of five classes of aircraft at all possible roll angles [66].

Texture Descriptors

Texture descriptors in the analysis of segments may be drawn from the same categories as those used in detecting the segments. There is a difference, however, in that the textural features in this phase of analysis must be more accurate and more sensitive to class differences than in the earlier phase. An example of this situation occurs in the analysis of mammograms. During the segmentation phase, primitive textures at a relative coarse resolution (approximately 500 μm), yield "suspicious regions." These "suspicious regions" (our segments) may then be analyzed at high resolution (25 μm), to classify the region among several classes, e.g., normal, benign cyst, benign increase in vascularity, carcinoma.

Syntactic Description

Syntactic descriptors provide descriptions of spatial relations among objects in an image and descriptions of relations among features. Such descriptors can vary from very simple ones, e.g., the number of blobs in a segment, to quite complex, e.g., a hierarchy of blobs within fibers within tissues within an organ (here the organ is the segment under analysis). The more complex descriptors are particularly appropriate for interaction with a human user, since unrelated features are likely to be insufficiently responsive to questions from an intelligent, inquisitive user.

If the segment is a silhouette, a concavity tree [68], [69] or a simplified form of distance skeleton [58] may be an appropriate descriptor. If the segment is a lung in a chest radiograph, a graph of boundaries and enclosed texture descriptors denoting and describing the ribs, tumors, mediastinum, pulmonary vascularities, and their spatial relationships may be an appropriate descriptor. If the segment is a stack of blocks, the links between faces of the blocks, as suggested by Guzman [17], and the relations between adjacent blocks (e.g., A is on B, A is on C, C is on D, etc.) may be appropriate. If the segment is a cell nucleus containing several blobs, the minimum spanning tree joining the centroids of the blobs may be an appropriate descriptor.

X. Concluding Remark

We have found that an understanding of a) the scene, b) the formation of the image, c) the acquisition and display hardware, d) the preprocessing, e) the difficulties the viewer may have in interpreting the image, and f) the knowledge base for the class of scenes and the overall task all can contribute effectively to segmentation and feature extraction. The best computerized image analyzers will most likely incorporate a tight interplay of all of these factors.

Acknowledgment

The author is indebted to many colleagues and students for much of the research described here. Among them are D. Ballard, G. Davison, A. Fam, M. Hassner, D. Kibler, C. Kimme-Smith, B. Lyon, M. Nadler, P. Nahin, and H. Wechsler.

REFERENCES

[1] J. W. Bacus, "A whitening transformation for two-color blood cell images," *Pattern Recognition*, vol. 8, no. 1, pp. 53–60, Jan. 1976.

[2] D. H. Ballard and J. Sklansky, "A ladder structured decision tree for recognizing tumors in chest radiographs," *IEEE Trans. Comput.*, vol. C-25, no. 5, pp. 503–513, May 1976.

[3] F. C. Billingsley, "Applications of digital image processing," *Applied Optics*, vol. 9, pp. 289–299, Feb. 1970.

[4] L. Carlucci, "A formal system for texture languages," *Pattern Recognition*, vol. 4, pp. 53–72, 1972.

[5] Y. P. Chien and K. S. Fu, "Recognition of X-ray picture patterns," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, no. 2, pp. 145–156, Mar. 1974.

[6] C. K. Chow and T. Kaneko, "Boundary detection of radiographic images by a threshold method," *Proc. IFIP Congress 1971*, North-Holland Co., Amsterdam, The Netherlands, Section TA-7, pp. 130–134, 1972.

[7] T. N. Cornsweet, *Visual Perception*. New York: Academic, 1970.

[8] W. Doyle, "Operations useful for similarity invariant pattern recognition," *J. ACM*, vol. 9, pp. 259–267, 1962.

[9] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973, ch. 12.

[10] R. W. Ehrich and J. P. Foith, "Representation of random waveforms by relational trees," *IEEE Trans. Comput.*, vol. C-25, no. 7, pp. 725–736.

[11] A. Fam, "On the representation of digitized images," Tech. Rep. TP-75-4, Pattern Recognition Project, School of Engineering, Univ. Calif., Irvine, June 1975.

[12] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electronic Comput.*, vol. EC-10, no. 2, pp. 260–268, June 1961.

[13] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972, chs. 8 and 9.

[14] M. M. Galloway, "Texture analysis using gray level run lengths," *Computer Graphics and Image Processing*, vol. 4, no. 2, pp. 172–179, June 1975.

[15] R. Gordon and G. T. Herman, "Three dimensional reconstruction from projections," *International Review of Cytology*, 1973.

[16] C. H. Granlund, "Fourier preprocessing for hand print character recognition," *IEEE Trans. Comput.*, vol. C-21, no. 2, pp. 195–201, Feb. 1972.

[17] A. Guzman, "Decomposition of a visual scene into three dimensional bodies," *Automatic Interpretation and Classification of Images*. New York: Academic, 1969, pp. 243–276.

[18] E. L. Hall, R. P. Kruger, S. J. Dwyer, III, D. L. Hall, R. W. McLaren, and G. S. Lodwick, "A survey of preprocessing and feature extraction techniques for radiographic images," *IEEE Trans. Comput.*, vol. C-20, no. 9, pp. 1032–1044, Sept. 1971.

[19] R. M. Haralick, K. Shanmugam, and L. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[20] A. Henson and E. Riseman, "Preprocessing cones: A computational structure for scene analysis," COINS Tech. Rep. 74C-4, Dep. Computer and Information Science, Univ. Mass., Amherst, 1974.

[21] R. M. Haralick and K. Shanmugan, "Computer classification of reservoir sandstones," *IEEE Trans. Geoscience Electronics*, pp. 171–177, Oct. 1973.

[22] M. Hassner, "Markov models of digitized images—A data compression analysis," presented at the *1976 International Symposium on Information Theory*, Ronneby, Sweden.

[23] R. M. Haralick and K. Shanmugam, "Comparative study of a discrete linear basis for image data compression," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, no. 1, pp. 16–27, Jan. 1974.

[24] F. Holdermann and H. Kazmierczak, "Generation of line drawings from gray-scale pictures," *Artificial Intelligence, AGARD Conference Proc.* No. 94, NATO, 7 Rue Ancelle, 97 Neuilly-sur-Seine, France, 1971, 31-1.

[25] W. S. Holmes, "Automatic photointerpretation and target location," *Proc. IEEE*, vol. 54, no. 12, pp. 1679–1686, Dec. 1966.

[26] R. M. Hord and N. Gramenopoulos, "Edge detection and regionalized terrain classification from satellite photography," *Computer Graphics and Image Processing*, vol. 4, no. 2, pp. 184–199, June 1975.

[27] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," *Proc. Second Int. Joint Conf. Pattern Recognition*, pp. 424–433, 1974.

[28] M. H. Hueckel, "An operator which locates edges in digitized pictures," *J. ACM*, vol. 18, pp. 113–125, Jan. 1971.

[29] B. Julesz, "Visual pattern discrimination," *IRE Trans. Informat. Theory*, vol. IT-8, no. 2, pp. 84–92, Feb. 1962.

[30] M. D. Kelly, "Edge detection by a computer using planning," in *Machine Intelligence 6*, B. Meltzer and D. Michie, Eds. New York: Wiley, pp. 397–411.

[31] C. Kimme, D. H. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," *Communications ACM*, vol. 18, no. 2, pp. 120–122, Feb. 1975.

[32] C. Kimme, B. J. O'Loughlin, and J. Sklansky, "Automatic detection of suspicious abnormalities in breast radiographs," *Proc. Conf. Computer Graphics, Pattern Recognition and Data Structure*, pp. 84–88, May 1975.

[33] A. Martelli, "An application of heuristic search methods to edge and contour detection," *Communications ACM*, vol. 19, no. 2, pp. 73–83, Feb. 1976.

[34] G. O. Lendaris and C. L. Stanley, "Diffraction pattern sampling for automatic pattern recognition," *Proc. IEEE*, vol. 58, no. 2, pp. 198–216, 1970.

[35] M. D. Levine and J. Leemet, "Computer recognition of the human spinal outline using radiographic image processing," *Pattern Recognition*, vol. 7, no. 4, pp. 177–185, Dec. 1975.

[36] S. Y. Lu and K. S. Fu, "Stochastic error-correcting syntax analysis for recognition of noisy patterns," *IEEE Trans. Comput.*, 1977.

[37] J. E. Markel and A. H. Gray, *Linear Prediction of Speech*. New York: Springer Verlag, 1976.

[38] A. Martelli, "An application of heuristic search methods to edge and contour detection," *Communications ACM*, vol. 19, no. 2, pp. 73–83, Feb. 1976.

[39] ——, "Edge detection using heuristic search methods," *Computer Graphics and Image Processing*, vol. 1, no. 2, pp. 169–182, Aug. 1972.

[40] B. H. McCormick and S. N. Jayaramamurthy, "Time series model for texture synthesis," *Int. J. Computer and Information Sciences*, vol. 3, no. 4, pp. 329–343, Dec. 1974.

[41] D. Merlin and D. J. Farber, "A parallel mechanism for detecting curves in pictures," *IEEE Trans. Comput.*, vol. C-24, no. 1, pp. 96–98, Jan. 1975.

[42] U. Montanari, "On the optimal detection of curves in noisy pictures," *Communications ACM*, vol. 14, pp. 335–345, May 1971.

[43] M. Nadler, "An analog-digital character recognition system," *IEEE Trans. Electronic Comput.*, vol. EC-12, pp. 814–821, Dec. 1963.

[44] P. J. Nahin, "The theory and measurement of a silhouette descriptor for image pre-processing and recognition," *Pattern Recognition*, vol. 6, pp. 85–95, 1974.

[45] R. Nathan, "Spatial frequency filtering," *Picture Processing and Psychopictorics*, Lipkin and Rosenfeld, Eds. New York: Academic, 1970, pp. 151–163.

[46] A. B. J. Novikoff, "Integral geometry as a tool in pattern perception," *Principles of Self-Organization*, Von Foerster and Zopf, Eds. New York: Pergamon, 1962.

[47] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 3, pp. 170–179, Mar. 1977.

[48] W. K. Pratt, W. H. Chen, and L. R. Welch, "Slant transform image coding," *IEEE Trans. Commun.*, vol. 22, pp. 1075–1093, 1974.

[49] J. M. S. Prewitt, "Object enhancement and extraction," *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970, pp. 75–149.

[50] ——, "The analysis of cell images," *Annals of the New York Academy of Sciences*, vol. 128, no. 3, pp. 1035–1053, Jan. 31, 1966.

[51] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing*, vol. 1, no. 3, pp. 244–256, Nov. 1972.

[52] J. S. Read and S. N. Jayaramamurthy, "Automatic generation of texture feature detectors," *IEEE Trans. Comput.*, vol. C-21, pp. 803–812, 1972.

[53] E. M. Riseman and M. A. Arbib, "Computational techniques in the visual segmentation of static scenes," *Computer Graphics and Image Processing*, vol. 6, pp. 221–276, June 1977.

[54] A. Rosenfeld, "Iterative methods in image analysis," *Proc. IEEE Computer Society Conf. Pattern Recognition and Image Processing*, Long Beach, CA, pp. 14–20, June 1977.

[55] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976.

[56] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, no. 5, pp. 562–569, May 1971.

[57] A. Rosenfeld and E. B. Troy, "Visual texture analysis," *Conference*

Record of the Symposium on Feature Extraction and Selection in Pattern Recognition, IEEE Publication No. 70-C, 51-C, pp. 115–124, Oct. 1970.
[58] A. Rosenfeld, *Picture Processing by Computer*. New York: Academic, 1969.
[59] B. Schacter, L. S. Davis, and A. Rosenfeld, "Scene segmentation by cluster detection in color space," Tech. Rep. 424, Computer Science Center, University of Maryland, 1975.
[60] S. D. Shapiro, "Transformations for the computer detection of curves in noisy pictures," *Computer Graphics and Image Processing*, vol. 4, no. 4, pp. 328–338, Dec. 1975.
[61] Y. Shirai, "A heterarchical program for recognition of polyhedra," *Artificial Intelligence*, Memo 263, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1972.
[62] J. Sklansky, "On the Hough technique for curve detection," Tech. Rep. TP-77-3, Pattern Recognition Project, School of Engineering, Univ. Calif., Irvine, Aug. 1977, to be published.
[63] ——, "An upper bound on the number of sides of a minimum-perimeter polygon," Interim Rep. IP-47-2, Pattern Recognition Project, School of Engineering, Univ. Calif., Irvine, July 1974.
[64] J. Sklansky and D. F. Kibler, "A theory of nonuniformly digitized binary pictures," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 9, pp. 637–647, Sept. 1976.
[65] J. Sklansky and A. Fam, "Variable resolution digital vision," *J. Information Science and Cybernetics*, vol. 1, no. 6, 1976. Also, Tech. Rep. TP-75-5 Pattern Recognition Project, School of Engineering, Univ. Calif., Irvine, July 1975.
[66] J. Sklansky and G. A. Davison, "Recognizing three-dimensional objects by their silhouettes," *J. Soc. Photo-opt. Instrum. Eng.*, vol. 10, pp. 10–17, Oct./Nov./Dec. 1971.
[67] J. Sklansky and P. J. Nahin, "A parallel mechanism for describing silhouettes," *IEEE Trans. Comput.*, vol. C-21, no. 11, pp. 1233–1237, Nov. 1972.
[68] J. Sklansky, "Measuring concavity on a rectangular mosaic," *IEEE Trans. Comput.*, vol. C-21, pp. 1355–1364, Dec. 1972.
[69] J. Sklansky, R. L. Chazin, and B. J. Hansen, "Minimum-perimeter polygons of digitized silhouettes," *IEEE Trans. Comput.*, vol. C-21, pp. 260–268, 1972.
[70] J. R. Ullman, "An algorithm for subgraph isomorphism," *J. Assoc. for Computing Machinery*, vol. 23, no. 1, pp. 31–42, Jan. 1976.
[71] H. Wechsler and J. Sklansky, "Finding the rib cage in chest radiographs," *Pattern Recognition*, vol. 9, no. 1, pp. 21–30, Jan. 1976.
[72] J. S. Weszka, C. S. Dyer, and A. Rosenfeld, "Comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, Apr. 1976.
[73] Y. Yakimovsky and J. A. Feldman, "A semantics-based decision theory region analyzer," *Proc. Third Int. Joint Conf. Artificial Intelligence*, pp. 580–588, 1973.
[74] C. T. Zahn, "Graph-theoretical methods for detailing and describing Gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, no. 1, pp. 68–86, Jan. 1971.
[75] C. J. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Comput.*, vol. C-21, no. 3, pp. 269–281, Mar. 1972.
[76] S. W. Zucker, "Toward a model of texture," in *Computer Graphics and Image Processing*, vol. 5, no. 2, pp. 190–202, June 1976.
[77] ——, "Region growing: Childhood and adolescence," *Computer Graphics and Image Processing*, pp. 382–399.

Enhancement of Noisy Images Using an Interpolative Model in Two Dimensions

H. R. KESHAVAN AND MANDYAM D. SRINATH, SENIOR MEMBER, IEEE

Abstract—**The use of two-dimensional interpolative models for recursive enhancement of noisy images is considered. The two-dimensional data is decorrelated either row-wise or column-wise to obtain an equivalent one-dimensional interpolative model. An adaptive identification scheme to identify the parameters of the model is presented which are then used in a simple estimation scheme for recovering the image data. A modification of the FFT algorithm as well as the DCT algorithm are used in decorrelating the data. The performance of these transforms for the present application is compared in terms of the number of computations required and the mean-square error between the enhanced image and the original noise-free image. Several examples are presented to illustrate the applicability of the algorithm.**

Manuscript received May 2, 1977; revised July 14, 1977 and October 28, 1977.

H. R. Keshavan was with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX 75275. He is now with the Microwave Division, Collins Radio Group, Rockwell International, Dallas, TX 75272.

M. D. Srinath is with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX 75275.

I. INTRODUCTION

THE TERM enhancement is generally defined as the process of improving the image appearance in such a way that either a human observer or a machine can extract certain desired information more readily from the enhanced image. In general, image enhancement involves the operation of eliminating noise disturbances contained in the image. The noise disturbances may be due to spurious object reflections, inaccuracies in the sensing mechanism, or disturbances introduced during transmission [1]. While both recursive and nonrecursive techniques have been used in the digital enhancement of images, the former are very attractive because of their on-line implementation capability, simplicity in implementation, and lower demand on storage requirements. While nonrecursive techniques employ the use of low-pass filtering, inverse filtering, use of orthogonal transforms, histogram manipulation, etc. [1]–[6], recursive techniques include the use of the Kalman filter and other sequential estimation techniques [7]–[13].

0018-9472/78/0400-0247$00.75