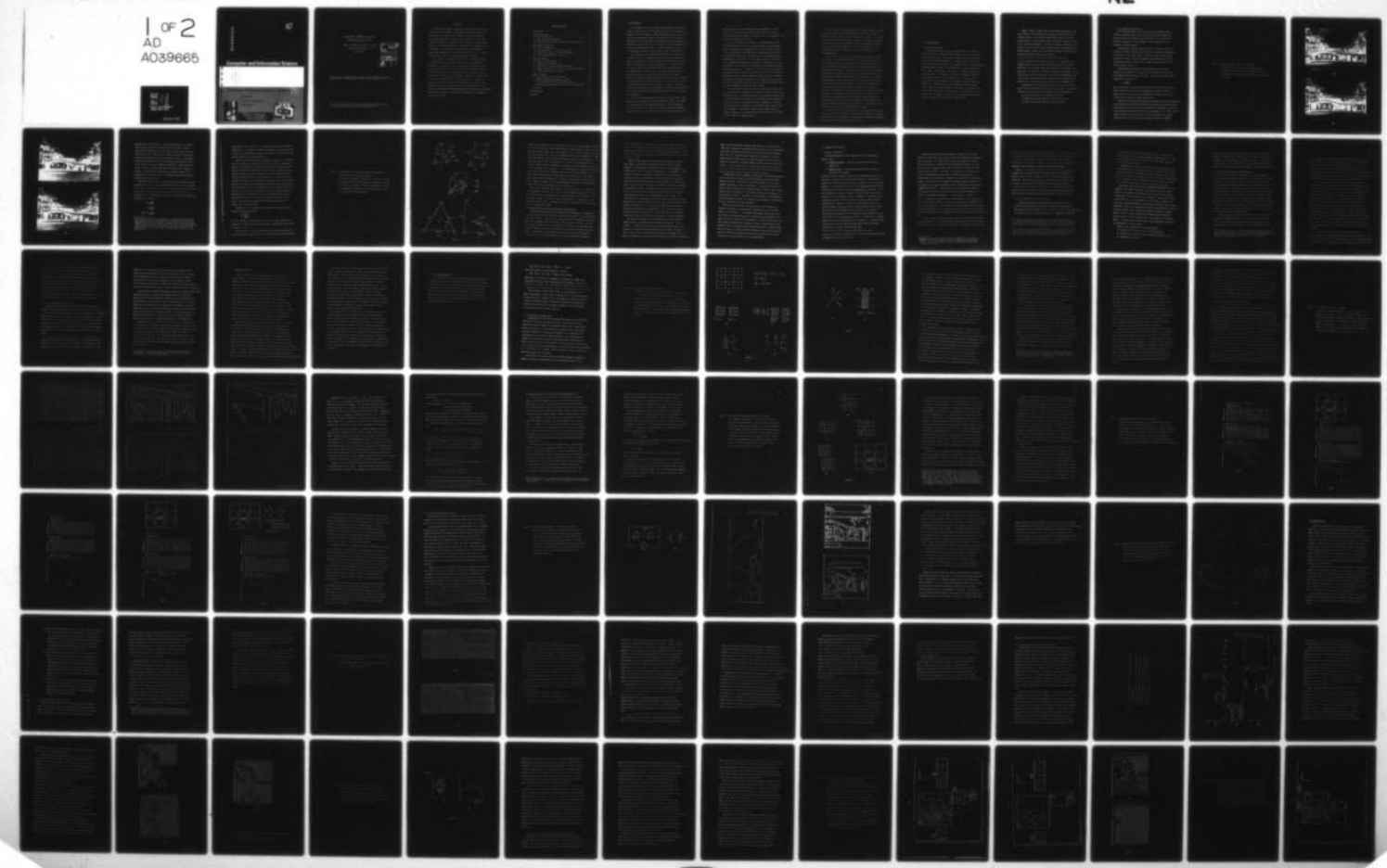
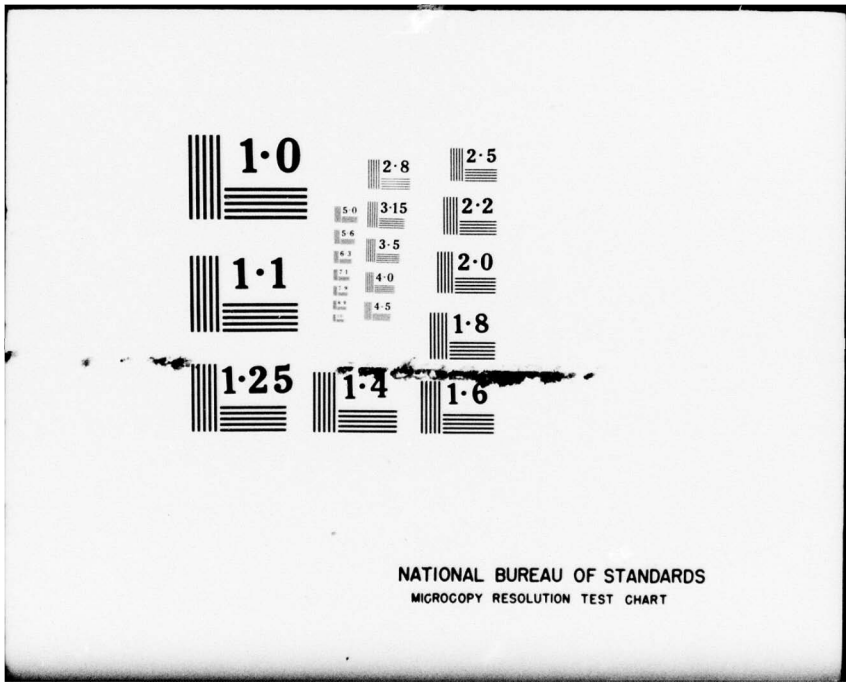
AD-A039 665 MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/G 9/2
COMPUTATIONAL TECHNIQUES IN THE VISUAL SEGMENTATION OF STATIC S--ETC(U)
MAR 77 E M RISEMAN, M A ARBIB N00014-75-C-0459
UNCLASSIFIED COINS-TR-77-4 NL

1 OF 2
AD
A039665

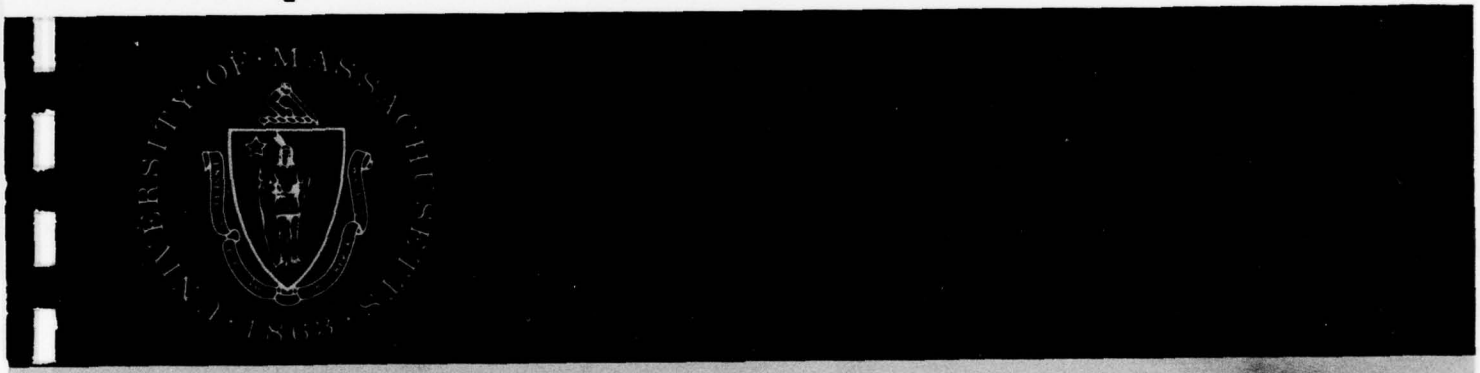



1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
6·3
7·1
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AD A039665

(8) DW

Computer and Information Science



University of Massachusetts at Amherst

Computers

Theory of Computation

Cybernetics

DDC FILE COPY

DDC
MAY 18 197
B

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

COMPUTATIONAL TECHNIQUES IN THE VISUAL SEGMENTATION OF STATIC SCENES[1]

Edward M. Riseman and Michael A. Arbib

COINS Technical Report 77-4

March 1977

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. Avail. and/or SPECIAL
A

This paper is a revised version of "Computational Techniques in Visual Systems. Part II: Segmenting Static Scenes", COINS Technical Report 76-11.

AD68330

[1]This work was supported in part by ONR grant N00014-75-C-0459 and NSF grant DCR75-16098, and by NIH grant 5R01 NS09755-06 COM.

Abstract

A wide range of segmentation techniques continues to evolve in the literature on scene analysis. Many of these approaches have been constrained to limited applications or goals. This survey analyzes the complexities encountered in applying these techniques to color images of natural scenes involving complex textured objects. It also explores new ways of using the techniques to overcome some of the problems which are described. An outline of considerations in the development of a general image segmentation system which can provide input to a semantic interpretation process is distributed throughout the paper.

In particular, the problems of feature selection and extraction in images with textural variations are discussed. The approaches to segmentation are divided into two broad categories, boundary formation and region formation. The tools for extraction of boundaries involve spatial differentiation, non-maxima suppression, relaxation processes, and grouping of local edges into segments. Approaches to region formation include region growing under local spatial guidance, histograms for analysis of global feature activity, and finally an integration of the strengths of each by a spatial analysis of feature activity. A brief discussion of attempts by others to integrate the segmentation and interpretation phases is also provided. The discussion is supported by a variety of experimental results.

TABLE OF CONTENTS

1. Introduction 1

2. Feature Extraction 3

2.1 Raw Input and Color 3

2.2 Hue, Saturation and Intensity 5

2.3 Extracting Other Features Over Windows of Variable Size . . 8

3. Segmentation and Texture 11

3.1 Goals of Segmentation 11

3.2 Problems and Goals in Processing Texture 13

3.3 Hierarchical Approaches to Texture Analysis 15

3.4 First- and Higher-Order Statistics 17

4. Boundary Formation 19

4.1 Spatial Differentiation 20a

4.2 Suppression of Redundant Data 21

4.3 Relaxation Processes for Boundary Formation 25

4.4 Applying Relaxation to an Inter-Pixel Edge Representation . 28

4.5 Grouping Edges into Line Segments 32

5. Region Formation 34

5.1 Region Growing via Local Analysis 35

5.2 Merging Regions Under Semantic Guidance 37a

5.3 Region Formation via Global Feature Analysis 42

5.4 Integrating Spatial Analysis with Global Feature Analysis . 45

6. Conclusion 52

Acknowledgements 55

References 56

I. Introduction

In the design of a general computer vision system for interpreting images, one must face many of the issues confronting the development of complex AI systems in general. Image understanding requires the processing of vast quantities of sensory data, with noise from the sensing mechanisms as well as non-semantic information obscuring the semantically significant entities that are to be perceived. One must organize both processes and knowledge structures in a modular fashion to interact in a flexible manner (Hanson & Riseman [1976], Arbib & Riseman [1976]).

The complexities in the design and implementation of such systems typically has led to a decomposition of the problem into distinct subsystems for segmentation and interpretation, often referred to as 'low-level' and 'high-level' processing, respectively. We view the goal of the initial stages of processing in visual systems as segmentation, a transformation of the data into a partitioned image with parts in a representation which is more amenable to the semantic processing. The general problems of segmentation involve processing arrays of numeric values representing brightness (and color) in order to extract features of boundaries and regions over local areas or 'windows'. By a variety of means this information can be aggregated, labelled with symbolic names and attributes, and then interfaced to knowledge structures by interpretation processes.

There has been some debate over the degree to which semantic information should be employed in the partitioning of an image. The problem of segmenting scenes with textural variations is rather challenging, and it is clear that the context of local data in a picture influences our interpretation of that data. Then it is reasonable to ask why the

image should not be processed immediately with knowledge of 'chairs', 'tables', or any other objects expected in the image. This will be discussed in more detail later in this paper, but it is worthwhile to make our views on this matter clear now.

A vision system which is to operate in a constrained domain with constrained goals will be able to use such knowledge to its advantage. However, this means that the segmentation operations cannot be applied to a new domain without providing the new knowledge for that domain; in each case the content, form, and manner of use of the domain-dependent knowledge must be specified. This also might involve serious computational considerations depending on the amount of knowledge and its use. This implies a reconstruction and evaluation of the segmentation system in each new application. It seems to us that there is a large degree of non-semantic patterns of sensory visual data which can allow effective, although not perfect, initial segmentation without recourse to semantics. A similar view has been expressed by Zucker, Rosenfeld and Davis [1975]. The human visual system can do quite well in partitioning nonsense images, even when neighboring regions are highly textured.

For these reasons we view the problem of image understanding as one of performing initial segmentation via general procedures, feeding this low-level output to a high-level system, and then allowing feedback loops so that the interpretation processes can influence refined segmentation. This allows semantic information to influence segmentation in a goal oriented way without coupling all such knowledge directly into the low-level processes. In this paper, however, we will look primarily at computer techniques for a one-way transformation from 'raw' visual input of static images to a segmented array.

From this point of view, the segmentation processes provide a compact description of the location and characteristics of visually distinct areas of the image. However, the local analyses may generate a great deal of spurious activity because objects in images do not appear as uniformly colored areas (as in cartoon drawings) but rather have natural textural variations, reflectance, shadows, etc. Thus, the integration of local processing into globally consistent boundaries and regions is not at all straightforward.

From a classic AI point of view, this analysis involves an enormous search space. If one adopts the ideal goal of bringing together these local representations of data into an optimal global representation, one must immediately face the combinatorics of the problem and the question of computational efficiency. Global brute force search is quite impossible, and of course one would not even recognize acceptable solutions without the application of higher-level processes to each alternative. Humans can understand images of natural scenes even in the presence of a high degree of noise and local textural variations. Clearly, the different phases of processing that are employed must be integrated and techniques to constrain the alternatives within each are necessary. Interaction between the analyses of local visual areas can be employed, but there must be provision for global guidance; not all possible global boundaries can be considered, but local noise in the formation of a long straight line should be handled by the global view of the line. In this paper we will examine some of the ways of dealing with these problems.

In the next two sections of the paper, we examine feature extraction, color, and texture. The main focus of this paper, techniques for boundary formation and for region formation, are presented in the next two sections, with a concluding discussion in the last section.

2. Feature Extraction

2.1 Raw Input and Color

Firstly, then, what is the 'raw' visual input? In an animal, it is simply the pattern of light (distributed across the spectrum) falling on the animal's retinas. This pattern changes over time as the animal moves and the environment changes. In a computer visual system, the input may be far more restricted. The simplest input is a black-and-white photograph which provides a two-dimensional map of light intensity in a static scene. Such an input can be subject to boundary formation and texture analysis. In this paper, we shall provide computer techniques for analyzing a static scene enriched by color. The usual way of representing a color photograph is by coding it as three arrrays, each sampling the brightness of the pattern through a different standard filter. Usually, the peak frequencies of the filters correspond to the three primary colors of red, green, and blue. This is true of the eye as well as of the computer: each rod in the retina has peak receptivity near the frequency of one of the three primary colors.

Figure 1 depicts a simple house scene viewed through each of the three filters and also averaged into a black and white (B & W) monochromatic image. If one views the blue component (Fig. 1c) of the colored image as a black-and-white photo, then bright regions are those with a strong blue component. Since white light has all spectral components, both blue sky and white clouds may appear indistinguishable in the blue image. However, the red and green components of the image will portray the boundary between sky and clouds. The color of the sky is actually cyan (greenish blue) which has a much larger green contribution than red. Consequently the red component (Fig. 1a) of the image would show the sky area to be much darker than the cloud area, while the contrast is not as great in the green (Fig. 1b) component. By properly viewing the three images one can estimate the colors of other areas, e.g., the roof and unshadowed side of the house is reddish, the grass is yellowish green (high in green, moderate red), the house trim is white (high in all components), etc.

Consequently, even the roughest sense of the color of an object cannot be determined without looking at all three values. On a dark to light gray scale from 0 to 63, a red value of 40 could represent:

1) a pure red (if the other components are 0); or

2) yellow (if the green value is 40, and blue is 0); or

3) white (if both green and blue are also 40); etc.

2.2 Hue, Saturation and Intensity

Due to these problems, the raw data is often transformed into a different coordinate system which is more intuitive to the human user: hue, saturation and intensity, often referred to in this paper as HSI features or parameters. The following is a brief discussion of the definition of these features.

The information associated with each point (i,j) can be viewed as a vector in 3 space, [R(i,j), G(i,j), B(i,j)]. We restrict each element of the vector to the range [0,63]. To help the reader visualize this, we adapt the clear and simple presentation provided by Schacter, Davis and Rosenfeld [1975], and view this as a vector within the 63 × 63 × 63 cube depicted in Figure 2a.

By viewing the brightness of a point as an average of the three primary color components, it is clear that the origin [0,0,0] is black and maximum brightness [63,63,63] is white. We may define a gray scale of brightness or 'intensity' by

$$I \triangleq \frac{R+B+G}{3} .$$

This is equivalent to the length of the projection of the vector [R,G,B] associated with any point upon the diagonal vector shown in Figure 2a. Thus, points in the color cube get progressively brighter as one moves from the bottom right to the upper left corners.

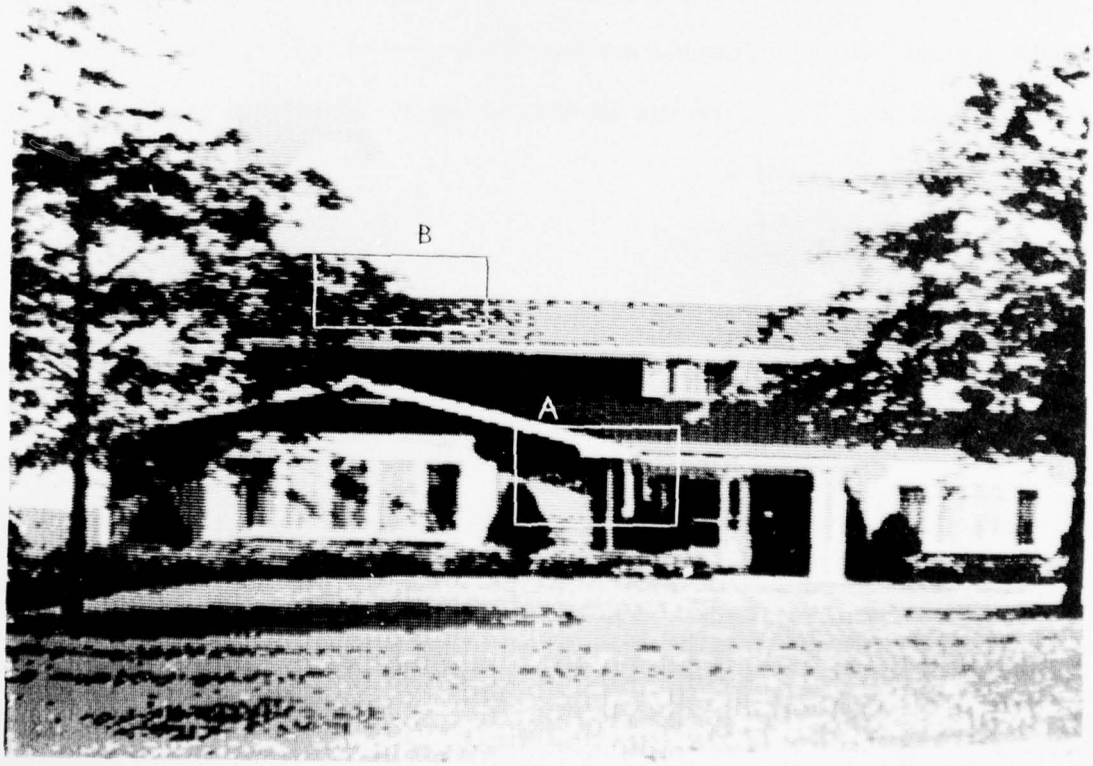
Other colors are obtained as one combines the primary colors in different amounts. The corners of the color cube are labelled in Figure 2b with the names of perceived colors which are formed from the three primary colors. For example, red and green in equal amounts produce yellow, when the blue component is 0. Thus, one can imagine the right face of the cube in Figure 2b varying across green, yellowish-green, yellow, yellowish-red (orange), and red. A diagonal

Figure 1: Digitized images of a natural color scene.
(a) Red, (b) Green, and (c) Blue components are shown.
(d) Intensity (or brightness) is an average of the first three images; subareas A and B will be used in examples later in the paper.

(a)

(b)

(c)



(d)

line from black to yellow (i.e., red and green components are equal) will represent yellow at different levels of intensity[1]. Now we need a way of describing the points inside the cube as well as on the surface of the cube. The points in a plane perpendicular to the gray-scale vector from black to white are of equal intensity. The largest such plane within the unit cube is the plane passing through the cube at the corners R, G and B, forming the equilateral triangle depicted in Fig. 2c and 2d. At any other level of intensity this triangle is smaller. The implication is that there is a smaller range of color combinations that can be formed as one approaches minimum and maximum intensity (black and white).

The color triangle of Fig. 2c can now be used to describe two other characteristics of color space, hue and saturation, which are independent of intensity. The intersection of the color triangle with the line between the origin and any point P in the color cube defines the projection of P onto the color triangle at 'P'. The placement of this point P' is defined by normalizing the values of R, G and B:

$$r = \frac{R}{R+G+B}$$

$$g = \frac{G}{R+G+B}$$

and $$b = \frac{B}{R+G+B}$$

---

[1] The problem is much more complicated from a psychological view because our perception of the color yellow is also a function of intensity and below some threshold, we might call it another color such as tan, brown, blackish-brown, or black. Human perception of color is a very complicated process and we will not be able to treat this problem in detail. The reader is referred to Evans [1948], Cornsweet [1970], Bouma [1971] and Beck [1975].

which implies that $r + g + b = 1$. Since there are only two independent variables, it is convenient to convert the equilateral color triangle into a right color triangle with the point P' defined by r and g (on the R and G axes, respectively) as shown in Fig. 2e.

Now one can specify the hue and saturation of point P'. Intuitively, hue can be thought of as representing the type of color. Saturation is a measure of the richness or purity of the color and is inversely proportional to the amount of white light diluting the hue. Both of the colors pink and scarlet may have the same hue, but pink is unsaturated while scarlet is highly saturated. If one represents the center of the color triangle as W (where W is the neutral point representing the projection of white and all gray levels between white and black), then the extension of the line between W and P' to the perimeter of the triangle describes the hue of P'; it is denoted by H in Fig. 2e. There is a one-one mapping between points on the perimeter of the color traingle and the angular orientation θ with respect to an arbitrary reference point, in this case R. Thus, hue can be represented as an angle θ with red as 0°, green as 120°, and blue as 240°.

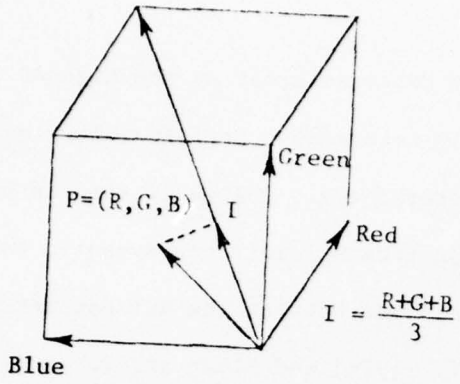
Saturation of P' is computed as a percentage of the distance of P' from W to the perimeter point H:

$$S = \frac{|\bar{P}'-\bar{W}|}{|\bar{H}-\bar{W}|} .$$

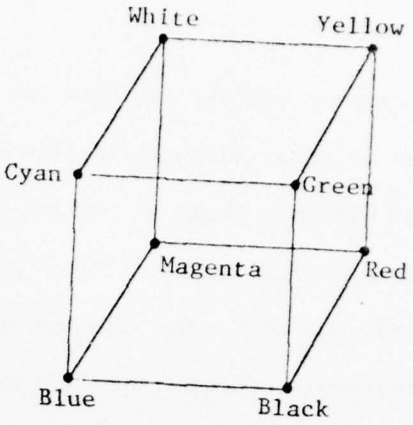
If P' is anywhere on the perimeter of the color triangle, then it has a saturation of 100% while the point W (white) is completely diluted and has a saturation of 0%.

The HSI features that we have defined are not entirely independent. If one examines the diagrams in Fig. 2, one can see that totally saturated

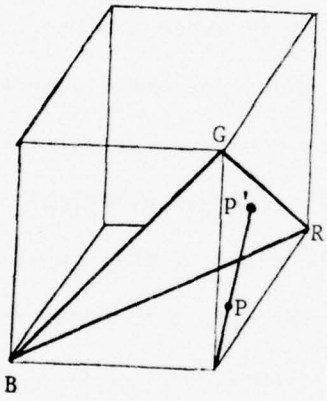
Figure 2: Transformation of the raw data (R,G,B) into parameters of hue, saturation, and intensity (H,S,I).
(a) The color cube and (b) the names of colors at the corners. (c) Formation of the color triangle. (d) Projection of a point p' on the color triangle. (e) Only two parameters of (r,g,b) are independent, producing the right color triangle; H and S are shown in this representation.



(a)



(b)



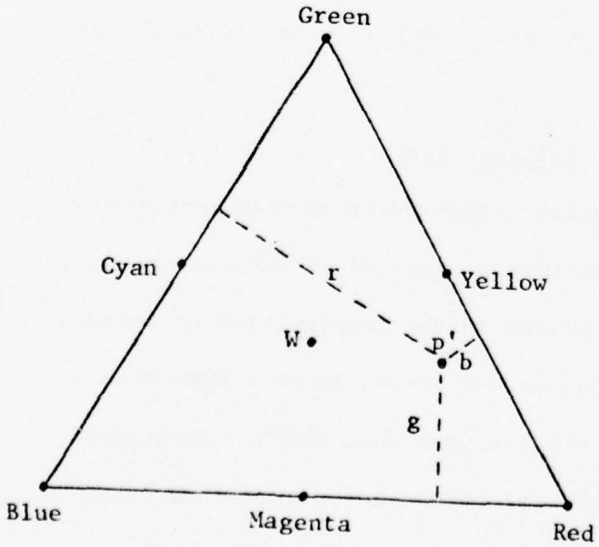
(c)

$$r = \frac{R}{R+G+B}$$

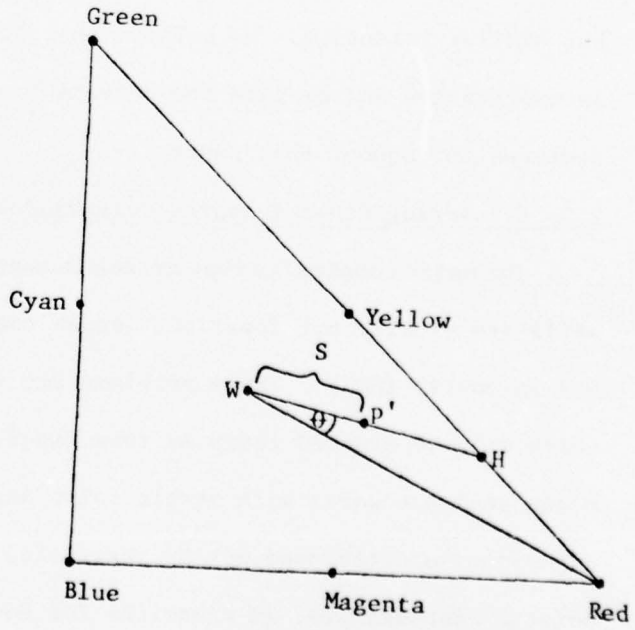
$$g = \frac{G}{R+G+B}$$

$$b = \frac{B}{R+G+B}$$

$$r + g + b = 1$$



(d)



(e)

Figure 2

yellow, cyan or magenta can have an intensity twice as great as the highest intensity red, blue or green which still remains totally saturated. Certain colors may only be perceived within a range of one of the parameters; e.g., yellow is seen as brown only when the intensity is low. Thus, if a mapping from HSI into the symbolic color names is desired, one must take into account dependencies between the HSI parameters. For other treatments of color see Tenenbaum et al. [1974] and Sloan and Bajcsy [1975].

Recently Kender [1976] has addressed a problem that some have known about, but has not been discussed in the literature. In the transformation to normalized components or HSI, there are points of instability where arbitrarily small changes in R, G, B will produce large differences in the transformed components; e.g., near point W, small changes in the raw components can cause very large changes in hue and saturation. Kender's treatment is a very thorough numerical analysis of the computation and use of color, but is beyond the scope of this paper.

Most of the information with respect to boundaries seems to be visible in the B & W intensity array of Fig. 1d. Thus, one can avoid the problems of color if one is willing to risk the disappearance of boundaries between areas of distinct color but similar intensity. We believe that color information is extremely useful for interpretation and despite the potential problems will continue to refer to the HSI features throughout this paper.

2.3 Extracting Other Features Over Windows of Variable Size

The major complexity that arises in segmentation is that the areas to be partitioned still are usually not invariant across the primitive parameters of hue, saturation, and intensity (HSI). These problems are intertwined in the complexities of texture which will be treated later in this paper. What we now stress is that even when scene analysis works with static color input, the features upon which segmentation algorithms operate need not be restricted to the HSI values associated with image points. For example, an algorithm for boundary detection may only produce the correct results if it operates on some of the average HSI parameters computed

across a local window of the right size; the proper boundary may only be obvious to local operators after some degree of blurring (which provides a more global viewpoint). Thus, one is faced with analyzing features of local windows of differing sizes as well as of individual points at the resolution level of the image.

Once the constraints on what constitutes a feature are relaxed in this manner, a huge class of possibly important properties becomes available. The meaningful feature might actually be the variance of a property over a local area, not just the average of that property. This provides a measure of invariance or homogeneity of a given property. If texture elements are extracted as atomic areas which are homogeneous in one or more of the HSI parameters, then the shape, size, and orientation of these areas might be the crucial property forming the cohesiveness of the perceived region. Although we shall not attempt to discuss the extent of the many efforts at feature extraction, properties for which computational procedures have been developed include: average of an area (blurring), average edge per unit area (spatial differentiation and then blurring), average orientation of local edges and average spot size of uniform contiguous area. All of these techniques have been carefully explored by Rosenfeld's group (Rosenfeld et al. [1970,71,72]) and are treated by Rosenfeld and Kak [1976]. Bajcsy [1973] has used frequency distribution in the Fourier domain in the analysis of texture gradients.

The computational games that are available for constructing more complex features by combining these techniques seem endless. Let us consider a sequence of operators to determine the orientation of line elements as the textural property characterizing a region. One might first compute a series of directional derivatives of the image in color space to determine the strength of color differences at various orientations;

then, in a blurring process, average these values over a local window of some size to delimit the areas which contain the lines (i.e., average edge/unit area); and finally differentiate these values for each orientation. The result at a particular point represents the strength of a boundary between areas on either side of it, where the values of these areas are based on the property of average strength of edges in the particular orientation. If the first two steps are replaced by a function which computes the size of atomic areas and then averages these sizes, the differentiation might discriminate between textures of different coarseness.

The important point to remember is that many of the algorithms discussed can work upon any array of extracted features, not just the simple examples presented. The problem is further complicated by the choice of applying algorithms to vectors of parameters. A spatial differentiation operator might be applied to the intensity array of a static scene to find large changes in brightness or to all three of the RGB or HSI parameters as a three-dimensional vector. The metric is often defined in one-dimensional and three-dimensional space, but in general can be applied in n-dimensional space (if n features have been extracted).

Given the state-of-the-art in scene analysis, one is faced with a combinatoric explosion of alternatives--experience has not yet provided answers to this problem. It is probable that working systems will require the ability to determine dynamically the proper size and membership of the subset of features employed by the algorithms. It must be stressed that many scene analysis systems will be tailored for specific applications--be they assembly lines, cardiovascular data, chromosome analysis or satelite imagery. Much of the success of such a system will depend on the judicious choice of those local features most likely to speed up the segmentation of the restricted class of images presented by the problem domain.

3. Segmentation and Texture

3.1 Goals of Segmentation

We shall distinguish two main approaches to the segmentation of natural scenes:

a) Boundary Formation - finding the boundaries which delimit a region; and

b) Region Formation - analyzing properties of areas to merge or split them into regions.

The goals of these two types of analysis are equivalent--they both form a partition of the scene into regions and boundaries. They both must employ some type of grouping, clustering, or binding of local areas/edges together. But the focus of the first is upon differences (discontinuity) in properties while the second is upon similarities of properties. It is quite possible that specific examples of these approaches could produce consistent or even isomorphic results. Placements of boundaries in one representation might be exactly between the regions formed in another representation. However, in practice algorithms which operate upon arrays of numbers representing complex visual information end up taking many different forms in dealing with the problems to be described. The data are often manipulated differently depending on whether one tries to form lines or extract properties of areas. A scheme which is tracking edges would be able to use the expected straightness of a boundary during the processing, while the region approach might collect distributed characteristics of widely separated local areas.

A powerful scene analysis system will make cooperative use of several such processes in handling all but the most sharply differentiated of regions (Arbib and Riseman [1976]).

Before we discuss particular segmentation techniques, let us look again at the scene depicted in Figure 1 and note distinguishing characteristics of the parts of the image that we would hope to extract as regions. The sky and clouds are relatively distinct homogeneous regions. It turns out to be easy to segment the main area of sky from the rest of the scene on the basis of intensity. The grass, only slightly more difficult since it has a rather homogeneous fine texture, becomes distinct from the surrounding areas on the basis of 'average' hue or intensity in a blurring process. Of course the area in shadow is separated sharply from the rest of the grass on the basis of intensity, but it turns out that there is only a slight shift in hue[1]. This means that there is information available during segmentation either to form the shadowed grass area separately or to bind it to the unshaded grass area. In these examples, it appears that a conservative strategy which forms separate regions might be better since there is information available to merge the regions with more confidence later under semantic guidance. If these regions are merged immediately, then problems of backtracking must be faced. The primitive regions which have been formed will need to be examined later to see whether they should be partitioned in an alternate way.

The more difficult areas in the scene of Figure 1 are the maze of textural variations in the tree, the smaller areas of detail which are not clearly defined in the house and shrubs, and the areas running off into shadows. The left window

---

[1] In general, one cannot expect the hue of a shadowed area to remain unchanged. In fact if the light is reduced significantly, the hue will be quite prone to error (Kender [1976]) as it approaches black through the lower intensities.

area is partly occluded by the leaves and branches, so that the distinct portions of the window trim and panes do not form areas which are easy to join together and interpret in the absence of context. This problem in the window area is compounded by reflections and shadows, e.g., the light areas of the window pane in the intensity image of Figure 1d are blue reflections from the sky.

One can now appreciate the difficulty of purely low-level formation of a single region covering the whole tree--both the area of tree with sky showing through, as well as the area of tree with obscured house in the background. The background textural elements are quite different in these two areas, yet there are common textural qualities which form one part of this macrotexture (the leaves and branches) in each case. This, and the fact that texture elements in the two areas are connected, are crucial clues which can be used to hypothesize the joining of these two regions.[1]

3.2 Problems and Goals in Processing Texture

The major problem for all segmentation techniques is texture.[2] We use the term texture rather loosely to encompass the variations in the visual properties of objects/surfaces/regions, including the texture induced by reflections from an irregular surface (e.g., highlights in the crown

---

[1] Once again, cooperation of high-level processes which know something about background areas showing through objects can be used to remove any remaining ambiguity. This reiterates the importance of our observation that high-level systems ought to affect segmentation at some point in the processing.

[2] Here we are referring to the primary difficulty in partitioning a scene into distinct visual components, not the major goal of determining the semantic relationships between the pictorial components. Later, we briefly discuss some attempts to integrate semantics with the segmentation process.

of a tree) and by occlusion of the light source (e.g., shadows in the crown of the tree due to branches blocking the light source). The areas to be partitioned are rarely uniform in any of the simple parameters of hue, saturation and intensity.

Segmentation processes are always faced with the difficulty of distinguishing between a region covered by texture elements and the texture element itself; the system might be mistakenly focussing upon the internal structure of a region. The proper area varies as a function of resolution, focus of attention, and goal orientation--is one attempting to bound the leaf, the branch, the clump of leaves, the tree from other trees?

Many studies have been conducted on images containing at most two textures or the simpler problem of classifying an image of a single texture. The problems that appear when one requires a single process to deal with arbitrary texture types in regions of varying size, quality, and placement have not been explored in the literature. Textures can occur as a recursive embedding of texture types to make the task even more difficult. Faced with a combinatoric explosion of possibilities, researchers have correctly chosen to deal with restricted classes of textures. However, the set of tools that have been developed might become more effective when a system can employ them in some general but structured manner. It appears that the time is ripe for an integrative attack upon the complexities of visual texture.

There are three common goals in texture analysis:

a) classification of texture into a set of categories;

b) description of texture in terms of primitive properties; and

c) segmentation of texture.

In the first two cases, one usually assumes that the given sample is an example of a single texture. Because this is not true in the third case, techniques for categorization of a single given texture or formation of its description are not sufficient for the determination of a boundary between two areas (of unknown sizes and shapes) but with distinct textures.

3.3 Hierarchical Approaches to Texture Analysis

One of the main problems in segmentation of textured regions is that the textural feature whose difference is to characterize the boundary may need to be extracted over a local area of unknown size and shape. If the information is sampled over areas that are not large with respect to texture elements or variations, then one cannot expect these local analyses to provide feature values that are invariant across the textured region. Consequently, it is desirable to extract the textural information over as large an area as possible. However, this leads to the 'window problem'--one cannot be sure of when the window area over which the feature is extracted is entirely placed inside a region or when it is extracting a 'mutant' value (i.e., confusing a mixture of two textures as a single new texture) because it overlaps regions.[1]

A general segmentation system will need the ability to extract such information over varying window sizes. The selection of the proper size for the 'receptive field' must surely be a dynamic decision (and sometimes could be provided by feedback from the interpretive process).

---

[1] This problem is related to the 'mixed pixel' problem. When an image is first scanned, the pixel could be on the boundary between distinct visual areas. This would produce a value between the values which would be produced for pixels entirely to the two sides of the boundary.

One can structure the system to analyze sets of increasing window sizes (e.g., $2^n \times 2^n$, n = 1,2,...) in some hierarchical manner (Rosenfeld and Thurston [1971], Marr [1975]) so that the correct size is sure to be included. One then must deal with the problem of automatically selecting the relevant data or maintaining all of it in some multi-level data structure. Although the problems become quite tricky, they do not seem insurmountable; however, such systems structures are still not yet understood very well.

The hierarchical processing cone structure (Hanson and Riseman [1974]) might allow an integrated attack upon these problems. Extraction of features over varying size windows is implicit in the design of the system. The processing cone is a simulation of a parallel array of micro-computers that is hierarchically organized into layers of decreasing resolution (256^2, 128^2, 64^2,..., 1^2). Sequences of operations allow full resolution image data to be transformed, compressed in amount, and stored at higher levels of the cone as coarse resolution features of subareas below. This allows both local and global features to be available simultaneously. Coarse descriptions of major areas might be utilized to guide the formation of more refined representations by merging atomic areas at lower levels. In this way the cone allows the system to work at both levels of description, either independently or dependently, but finally with the goal of bringing the local and global descriptions together.

An interesting approach to the recursive embedding of texture characteristics has recently been suggested by Ehrich and Foith [1975, 1976]. A versatile data structure for extracting the relationships between intensity peaks and valleys of a one-dimensional scan line, called a 'relational tree',

has been developed. The description of peaks and subpeaks (which could represent microtextures within a macrotexture) in terms of their width and relative heights can be extracted from the waveform rather simply. The relational tree captures structural information in a hierarchical fashion; fine texture appears as 'frontier peaks' embedded in coarse (wide) peaks representing more global textural characteristics. The two-dimensional case becomes somewhat more complicated since the structure of distinct scan lines in the same or different orientations must be correlated. The approach appears to be quite powerful for classification and description, and bears promise for applications to segmentation.

3.4 First- and Higher-Order Statistics

One approach to texture description uses first- and higher-order statistics of (monochromatic) scene elements (Julesz [1975]). The first-order statistic is simply the average gray level of an area; and differences in this parameter have been widely employed in previous work. The computation of the second-order statistic for an area requires the determination of the likelihood of finding gray levels i and j for pairs of points as a function both of the length and orientation of a line between them. Third-order statistics are extracted as a function of the relationships between three-tuples of points.

The information in second-order statistics is precisely the data contained in the 'gray level adjacency' matrices which have been studied by Haralick [1973] and Rosenfeld & Troy [1970]. For a given length and orientation, a square matrix of the co-occurrences of gray level i with

gray level j in the defined relationships must be constructed. This technique has been used effectively for classification of texture samples by transforming each matrix of values into a scalar value by computing features such as the angular second moment about the diagonal (ASMD). It is interesting to note that the ASMD can be computed locally in parallel using little intermediate storage in the processing cones (Hanson & Riseman [1974]).

In a series of interesting experiments, Julesz et al. [1973] and Julesz [1975] demonstrated that two textures with identical first- and second-order statistics but different third-order statistics cannot be spontaneously discriminated by a human observer, while differences in first- or second-order statistics generally allow spontaneous discrimination. They showed that textures could be constructed with these characteristics by performing simple transformations of the texture element[1]. This would imply that if the first- and second-order statistics were extracted from a texture, these often could be used to determine boundaries of the textured area.

Unfortunately, use of second-order statistics is not a computationally viable approach. For purposes of segmentation, the amount of data that would have to be collected to determine similarity or differences of general second-order statistics of unknown areas of arbitrary size, shape, and placement is an enormous data overload. Thus, segmentation based on extracting a full range of second-order statistics seems doomed to failure. However, the use of selected features dependent only on second-order statistics could prove quite fruitful.

---

[1] It should be noted that this study was constrained to black and white binary images. Although the extrapolation to more general scenes is reasonable, it should be done with caution.

4. Boundary Formation

There are many ways to form a description of a scene in terms of a line drawing. There are several intermediate representations of boundaries that often are formed prior to obtaining the final representation. Computation of the strength (and sometimes orientation) of the gradient of intensity can be obtained via the application of a spatial differentiation operator. The transformed image is composed of independent edges whose spatial relationships, among other things, can be used to infer more global entities. Optionally, these edges might be filtered to remove redundant and/or less important edges. Then, a subset of edges might be linked into line segments; in some domains these segments might be restricted to linking edges that either form a straight line or observe certain constraints on edge orientation. Finally, the line segments might be grouped together in terms of the standard ways lines may come together at vertices or in terms of more complete boundaries.

Much of the early research in scene analysis was based on techniques for tracking straight lines (Roberts [1965], Binford & Horn [1971]). If the objects under consideration were polyhedra, then knowledge about their vertices could also be employed during or after the formation of straight line segments (Roberts [1965], Clowes [1971], Huffman [1971], Shirai [1972], Duda & Hart [1973], Waltz [1975]). It should be evident that in natural scenes these techniques will be quite limited if utilized alone. More general nonsemantic procedures for binding local edges into longer segments are needed. An interesting approach that the reader should be aware of, but that we will not examine here, involves an understanding of surfaces, their orientation, and the light reflected from them (Mackworth [1973], Horn [1975]).

There are a variety of techniques and control strategies that can be used to form edge representations. Edges can be seqeuentially tracked along points of roughly uniform gradient strength. On the other hand, edge information can be extracted prior to sequential or parallel binding of edges into line segments. Colinear edges can produce clusters in feature space via Hough-like transforms (Duda and Hart [1973]). This can allow groups of edges with similar properties to be globally analyzed and provide local direction to the control of boundary formation (O'Gorman and Clowes [1973], Nevatia [1975], Shapiro [1975], Wechsler and Sklansky [1975]). The latter approaches bear similarity to techniques for region analysis presented later in this paper, and we hope the reader can extrapolate their potential by considering the general utility of global feature analysis in forming regions.

One cannot expect a low-level system to directly provide all final boundary representations which might be meaningfully interpreted by a semantic processor. This search space is enormous and constraints upon the final representation are almost always embedded in the techniques and control strategies. Sometimes in cases of uncertainty the goal of forming a single final representation can be relaxed, and the determination of a consistent representation can be delayed for other processes which utilize different types of knowledge. Thus, we will limit ourselves to examining variations on two approaches to the early stages of finding lines. A survey on edge detection techniques by Davis [1974] focusses upon other issues and approaches, providing a nice complement to the treatment in this paper.

4.1 Spatial Differentiation

As we have stated, the usual first step in computing boundaries is the application of a spatial differentiation operator (often defined as an edge mask or template) to transform the original image into one with edges highlighted. Although many such operators have been suggested (Hueckel [1973], Bullock [1974], Fram and Deutsch [1975], McKee and Aggarwal [1975], Marr [1975]), one that combines low complexity with high reliability is an operator (Kirsch [1971]) computed on the local window shown in Figure 3a as follows. Let

$$M_i = |5(a_i + a_{i+1} + a_{i+2}) - 3(a_{i+3} + \ldots + a_{i+7})|$$

where the indices are computed modulo 8; then let

$$S(X) = \text{Max } M_i \quad \text{and} \quad D(X) = \{i \mid \text{such that } M_i \text{ is max}\}.$$

This gives, at every point X, estimates of the gradient strength $S(X)$ and gradient direction $D(X)$ (quantized to 45° intervals). Later we will show that it is useful to save the sign of the gradient; this will tell us the sides that are light and dark as we move across an edge of a given contrast.

If X is within a uniform area, $S(X) = 0$ and orientation of an edge is meaningless, whereas $D(X)$ is defined, but not necessarily unique, in all other cases. Actually $D(X)$ only encodes four unique orientations. For each orientation, though, information is available as to which side of point X is the best fit of the edge; an example of the two placements of a vertical boundary is shown in Fig. 3b.

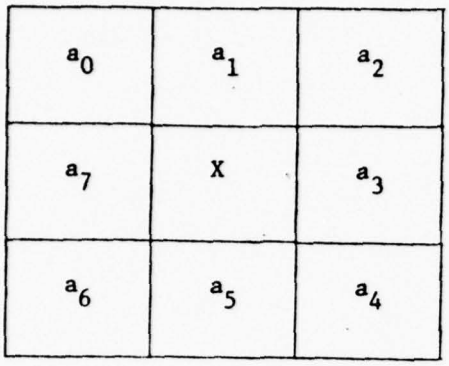
4.2 Suppression of Redundant Data

A disadvantage of most spatial differencing operators is that multiple indications of the same line can be produced. The raw digitized data sometimes introduces a gradient of brightness which is not a step function when one is expected. In the house scene of Figure 1d, the sharp boundary between sky (intensity ≈ 52) and roof (intensity ≈ 33) actually has one intermediate row of transition values (intensity ≈ 46). This problem is related to the placement and size of the scanning point which might overlap the areas (the problem of "mixed pixels"). In many cases there is a ramp function in the data because of shadowing and highlights. Thus, many different window placements will redundantly detect a boundary, whereas the goal is to find a single line which best separates the two areas.

In the case of the specific operator we have introduced, an additional problem of multiple representations of the same boundary occurs. If an

Figure 3: A spatial differentiation operator.

(a) The strength S(X) and the orientation D(X) of the gradient at X. (b) Placement of edge with respect to X. (c) Edges which are logically equivalent can be formed at adjacent points. (d) Non-maxima suppression could cause fragmentation. (e) Shifting edges can standardize their location. (f) Directions for non-maxima suppression of edges. (g) Suppression operates only for edges with the same sign of the gradient so that one pixel wide regions can be detected.

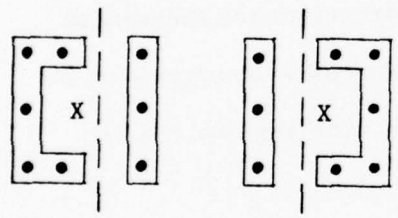


(a)

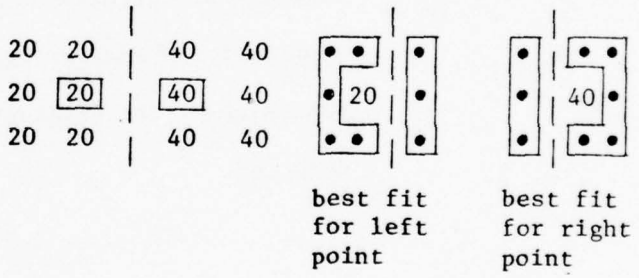
$$M_i = |5(a_i + a_{i+2}) - 3(a_{i+3} + \ldots + a_{i+7})|$$

$$S(X) = \text{Max } M_i$$

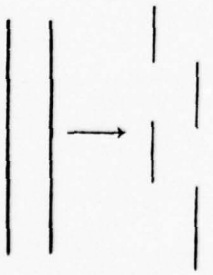
$$D(X) = \{i | M_i \text{ is Max}\}$$



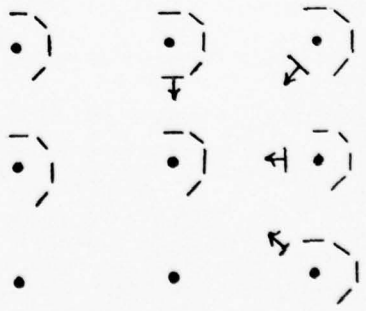
(b)



(c)

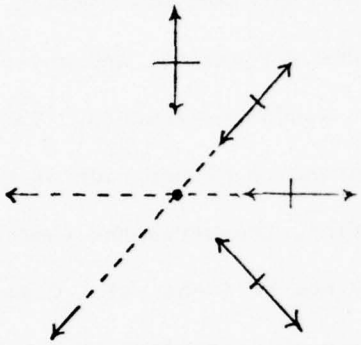


(d)

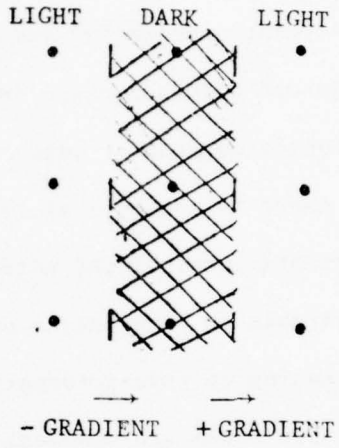


(e)

Figure 3



(f)



(g)

Figure 3

edge occurs on one side of point X, the edge will be detected when the 3 × 3 window is centered on some of the points adjacent to X. A simple case is depicted in Figure 3c where a vertical line is detected as equivalent adjacent boundaries with equal strength and orientation. Any single point appears in nine different window placements, and any pair of adjacent points appears in six different window placements. There is a confusing overlap of edge analysis. In areas to either side of an edge where there is some (possibly minor) variation, the strengths (and even the orientations) of the adjacent edges may not be identical. Clearly it is desirable to have only a single indication of a boundary, and techniques for cleaning up this information are called for. However, as redundant and weaker edges are removed, the condition of Figure 3(d) must be avoided; the suppression of local edges should not lead to global fragmentation of a line. Two operations will be employed to enhance the meaningful information: removal of logically equivalent edges and suppression of non-maximum strength edges.

The representation can be simplified by adopting a standard position for edges at a given orientation, thereby eliminating separate indications of logically equivalent edge positions. Currently, a pair of parallel edges at adjacent pixels can represent a variety of situations; edges which are two pixels apart (and probably distinct), edges one pixel apart, or edges actually in the same position. By adopting a general convention of shifting edges, pairs of adjacent boundaries can be collapsed into a single representative in a consistent manner. The standard positions that we have selected for the four orientations of our operator are shown in Figure 3(e). Each of the four orientations associated with a pixel now has a fixed position relative to the pixel; an edge can only appear in the north to southeast semicircle about a pixel.

The edges in non-standard position can be uniquely shifted to the pixel which has that edge in a logically equivalent, but standard, position. The neighboring edges which could shift into a single pixel are also shown in Figure 3(e). In the end there are still 8 edge values competing at four unique locations with the maximum surviving. This could be computed directly at the first application of the edge masks and allow the 8 possibilities to compete directly rather than be distributed among 5 pixels (and of course competing with many edges in other positions) before their results are collected into the single pixel.

For many edge operators suggested in the literature, suppression techniques are limited (or noisy) because information which encodes the placement of the edge with respect to the pixel is not available. The suppression schemes must focus upon the strength and orientation of these boundaries in order to clean up the edge image. Various thinning and smoothing techniques have been suggested. Rather than review this body of literature, we will examine only the techniques for suppressing non-maxima (Rosenfeld & Thurston [1971], Hayes et al. [1974]) of edges and spots. They seem to be directed towards the heart of the problem--a local analysis which retains a good fit of an edge and suppresses redundant data.[1]

Suppression can take place by having each local edge examine the strength and orientation of the edges in its neighborhood. It will be suppressed by indications of parallel (or possibly, near parallel)

---

[1] This process can actually proceed prior to grouping local edges into a more global line. However, a single straight line which is globally the best fit might be useful in directing the local analysis and is an argument for delaying such local suppression.

lines of greater strength nearby. The simplest heuristic scheme given our edge representation is portrayed in Figure 3(f) where an edge at some orientation can be suppressed by a stronger parallel edge which is adjacent in a direction perpendicular to the orientation of the first; this means that for each edge, we must examine the values of exactly two of its neighbors. There are many other heuristics that can be employed as a function of strength and orientation; e.g., edges at 45° angles to each other might also activate suppression, or can require a different threshold factor of relative strength before suppression will succeed.

Finally, there is a problem with this suppression scheme in that one pixel wide regions will produce parallel edges which would suppress each other. Here we can employ the sign of the gradient to discriminate between distinct boundaries as depicted in the example of Figure 3(g). Only parallel edges having the same gradient sign can be multiple instances of the same edge; suppression will not take place otherwise.

Figure 4 shows a differentiated portion of subimage A in Figure 1d (the diagonal roof and door area in the house); Figure 4a and 4b represent S(x) and D(x) before suppression. Note that in Figure 4a S(x) has been scaled by a constant factor, and the sign of the gradient has been included to guide the later stages of suppression. For simplicity the four orientations of edges are represented graphically in Figure 4b, even though this leaves ambiguous the exact position of each edge in the diagram. In Figures 4c and 4d, the edges have been moved to standard positions so that their position relative to the pixel is between north and (moving to the right) southeast. Now adjacency and contact of boundaries is clearer.

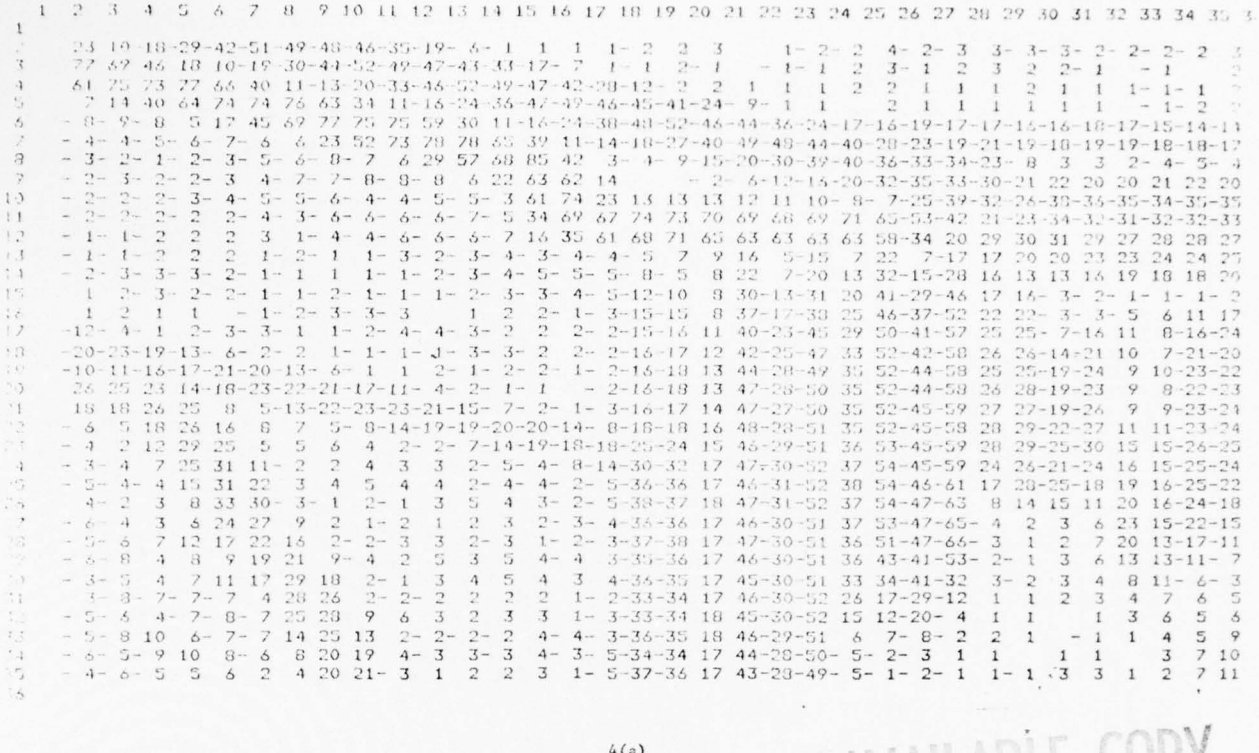
Figures 4e and 4f show the results of first suppressing non-maxima and then thresholding out weak edges; the thresholding is carried out by computing mean and variance of non-zero S(x) and removing edges whose strength is below $\mu + k\sigma$ (here $k = -.25$). Note the presence of the one-pixel-wide light vertical region in the bottom center of the image. The important boundaries stand out clearly but there are still some spurious and redundant edges along some boundaries or at vertices. These edges are a result of the complexities introduced by many of the edge formation windows overlapping a boundary in various ways. Many of them can be removed by using a suppression pass that is slightly more sophisticated.

4.3 Relaxation Processes for Boundary Formation

All of the preceding considerations might be generalized and embodied in a process of competition and cooperation within the parallel 'relaxation' procedures formulated by Rosenfeld, Hummel & Zucker [1976]; using this approach there has been an exciting range of applications from boundary analysis (Zucker, Hummel & Rosenfeld [1975], Vanderbrug [1975]) to template matching (Davis & Rosenfeld [1976]). This approach of distributed computation overlaps earlier ideas including the spring-loaded templates to flexibly map parts into a whole (Fischler and Elschlager [1973]), "constraint satisfaction" applied to labelling vertices of polyhedra with shadows (Waltz [1975]), and to the formation of a consistent set of labels for the identities of regions by Tenenbaum and Barrow [1976].

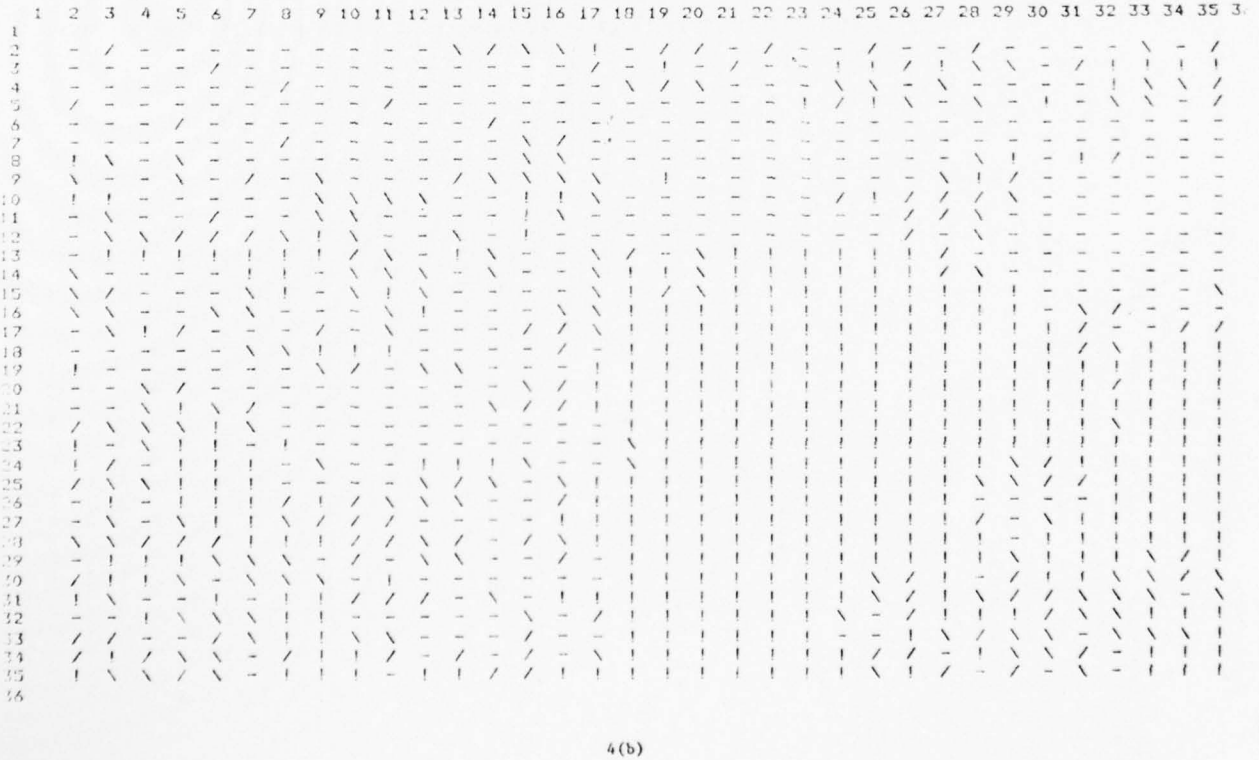
Here we will briefly review the general idea while applying these ideas of distributed computation to boundary formation. This approach can embody not only nonmaxima edge suppression but also edge fitting and binding. The advantage of the relaxation techniques is that likelihoods of all orientations of each adjacent point can contribute to the label assigned to a given point, not just the 'best' choice for adjacent elements. Whereas in the previous algorithms the strengths of edges at nonoptimal orientations are thrown away, they now prove very useful. Thus, a break in a long horizontal line might be repaired automatically by the context. We present our own variant to the approach of Zucker et al. [1975].

Figure 4: An example of processing local edges.
(a)-(b) The strength and orientation of edges produced by applying the operator of Figure 3 to subimage A in Figure 1(d). The sign of the gradient is retained to show the relative brightness on each side of an edge. (c)-(d) Removal of logically equivalent edges by standardization of the position of edges with respect to the pixels they separate. (e)-(f) Suppression of non-maxima edges and thresholding edges whose strength is below $\theta = \mu - .25\sigma$.

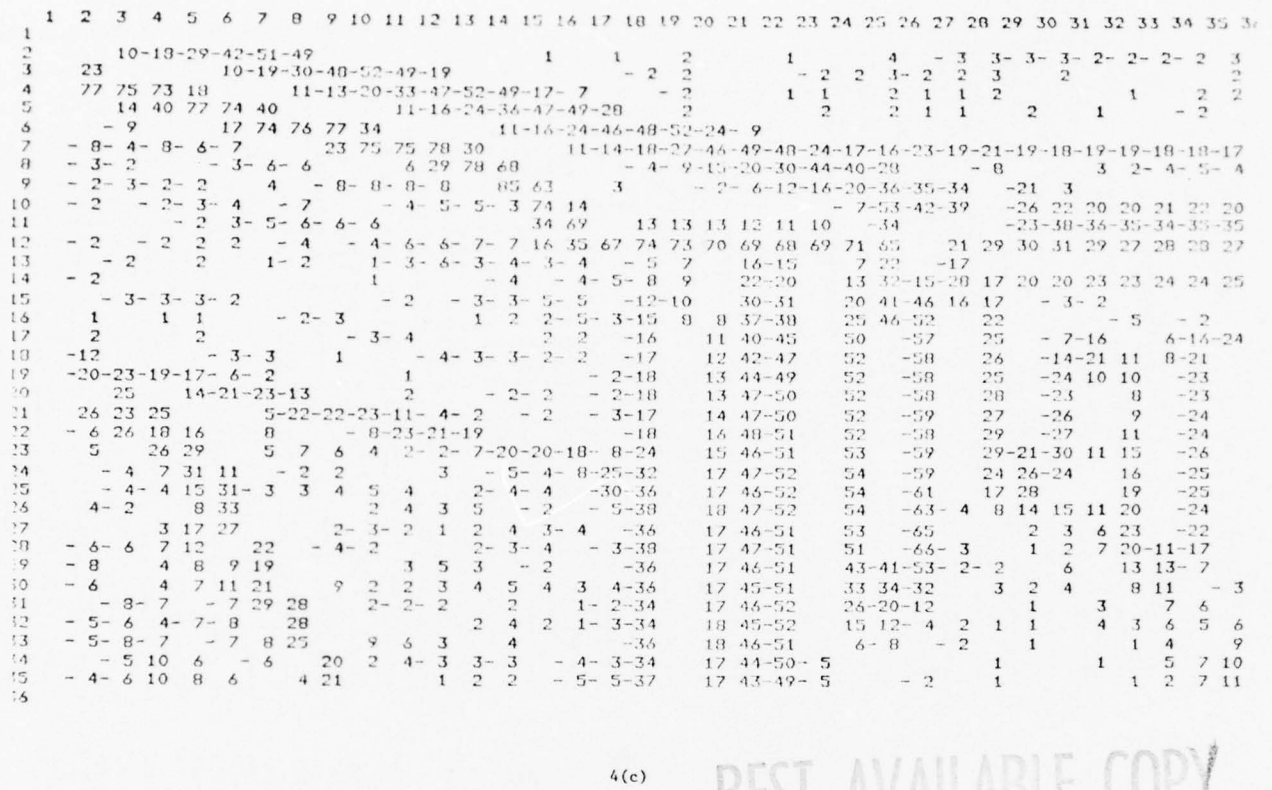


4(a)

BEST AVAILABLE COPY

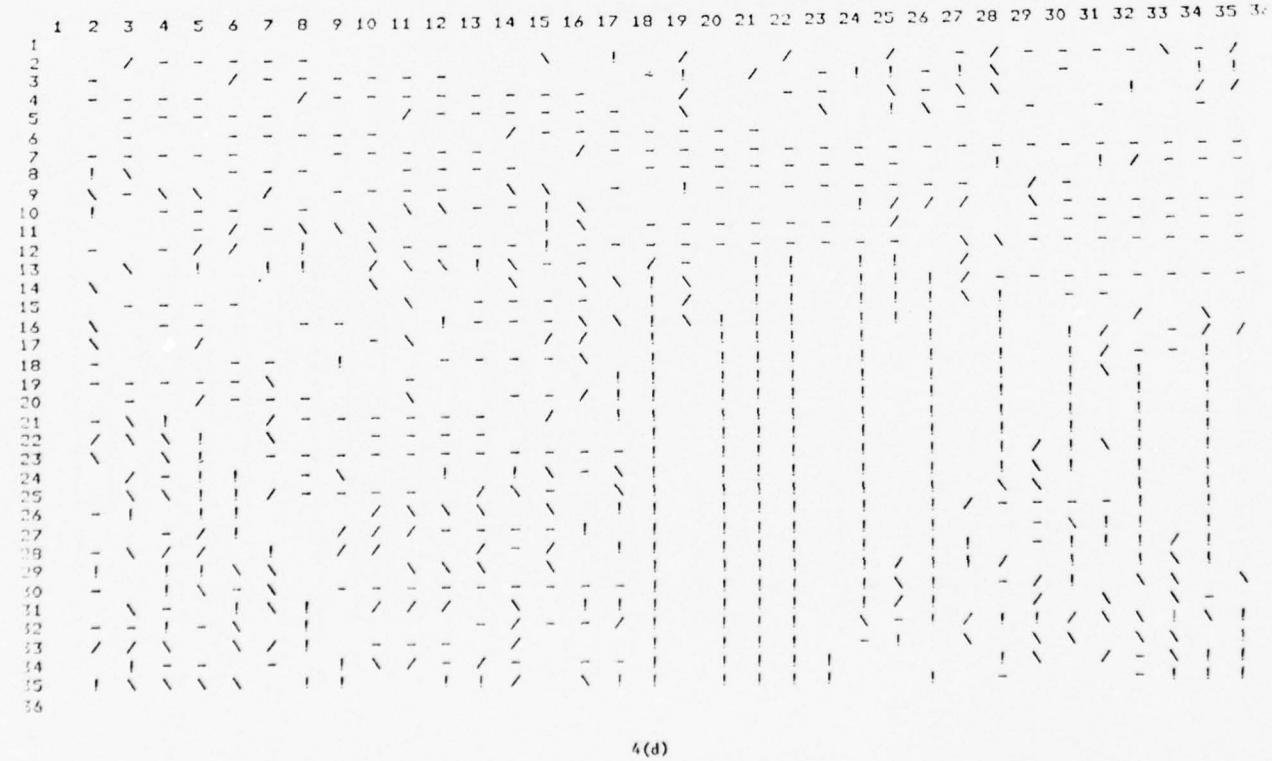


4(b)

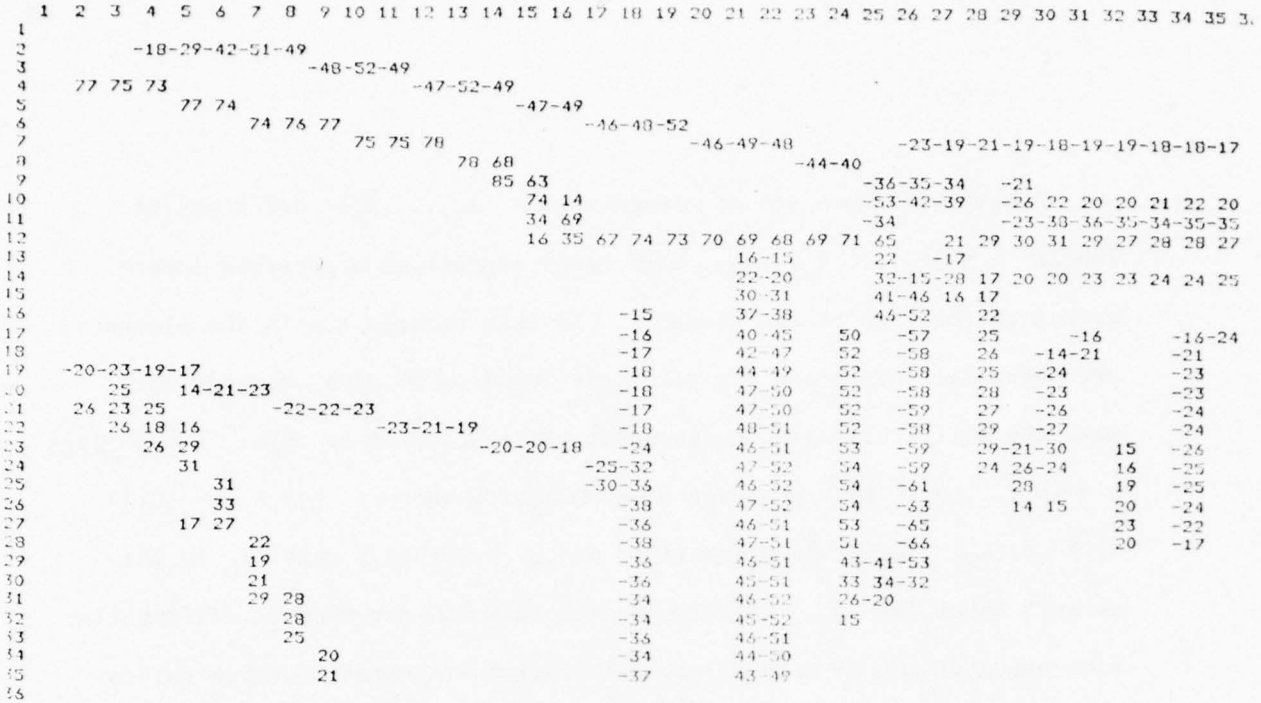


4(c)

BEST AVAILABLE COPY

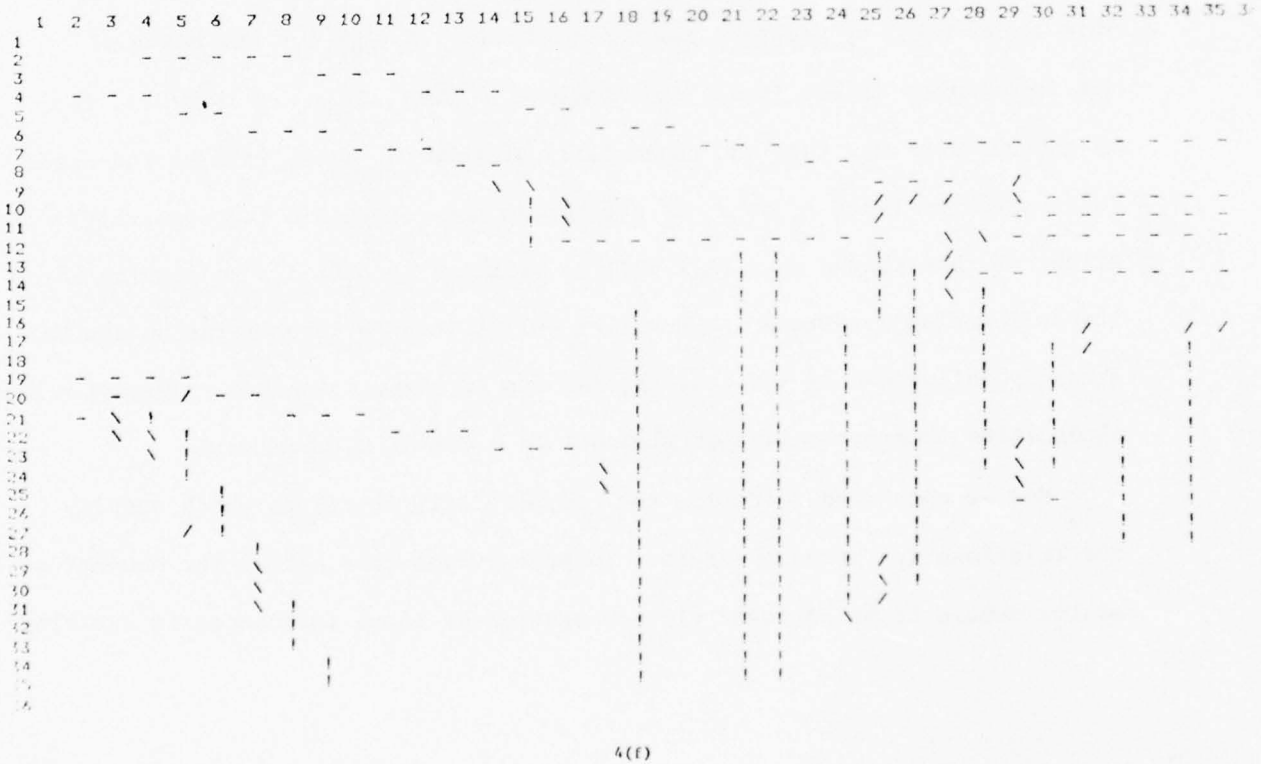


4(d)



4(e)

BEST AVAILABLE COPY



4(f)

Assume we have a set of elements $A = \{a_1,\ldots,a_n\}$ and a set of labels $\Lambda = \{\lambda_1,\ldots,\lambda_m\}$, where each label represents a possible interpretation for each of the elements. In this example domain the elements are the image points and the label set consists of edge orientations, with the null label used to represent the absence of an edge. A labelling $p = (p_1,\ldots,p_n)$ is a sequence of probability vectors $p_i: \Lambda \longrightarrow [0,1]$ with $p_i(\lambda_k)$ being the probability of the hypothesis that λ_k is the correct label for a_i. Shortly we will show how our spatial differentiation operator can be used to provide initial estimates of these values.

The relaxation process involves an iterated updating of these probabilities in an attempt to move P towards a globally correct labelling. This is achieved by updating the value of each $p_i(\lambda_k)$ on the basis of the information in its local "neighborhood". Thus, if a_j is in $N(a_i)$, the neighborhood of a_i, then the probability of label λ_k at a_i will be increased (decreased) by label λ_ℓ at a_j if the labels are compatible (incompatible). The effect of this change on $p_i(\lambda_k)$ will be weighted by $p_j(\lambda_\ell)$, the likelihood of the influencing hypothesis. Thus, the belief in each interpretation can be strongly influenced by its context, leading to competition and cooperation between alternative interpretations of elements in a common neighborhood.

Now we only need to define the compatibility functions which specify the relationships between labels. To some extent this allows the semantics of the domain to be employed via propagation of local influences in arriving

at a global interpretation. We define the compatibility function between a_i and a_j as

$$r_{ij}: \Lambda \times \Lambda \longrightarrow [-1,1]$$

such that $r_{ij}(\lambda_k,\lambda_\ell) > 0$ if λ_k and λ_ℓ are compatible;

< 0 if λ_k and λ_ℓ are incompatible;

$= 0$ if λ_k and λ_ℓ are independent.

Here we use the term compatible in the sense of the phrase "lends support to". For edge labelling the compatibility of edge orientations must capture both the types (orientation) of edges as well as their spatial relationship.

Finally, we have the basic idea of updating the change in $p_i(\lambda_k)$ as

$$\Delta p_i(\lambda_k) = \sum_{j \in N(a_i)} d_{ij} \sum_{\ell=1}^{m} r_{ij}(\lambda_k,\lambda_\ell) p_j(\lambda_\ell) \quad \text{for } i = 1,\ldots,n \text{ and } k = 1,\ldots,m$$

where d_{ij} is a weighting of the influence of the various a_j upon a_i. Let us denote the probability of a label λ_k after the t^{th} iteration as $p_i^{(t)}(\lambda_k)$. Since $p_i + \Delta p_i$ can become negative for a label with strong negative evidence from its context, the updating will be nonlinear as follows

$$p_i^{(t+1)}(\lambda_k) = p_i^{(t)}(\lambda_k)[1 + \Delta p_i^{(t)}(\lambda_k)]$$

with Δp_i remaining in the interval from -1 to +1.

We now modify the equation to normalize the updated values across $k = 1,\ldots,m$ in order to maintain a probability vector

$$p_i^{(t+1)}(\lambda_k) = \frac{p_i^{(t)}(\lambda_k)(1 + \Delta p_i^{(t)}(k))}{\sum_k [p_i^{(t)}(\lambda_k)(1 + \Delta p_i^{(t)}(k))]} .$$

This updating process can be iterated some number of times, converging upon a locally consistent interpretation which hopefully is a globally acceptable interpretation. Some results on the convergence of this process are provided by Zucker, Krishnamurty and Hoar [1976].

4.4 Applying Relaxation to an Inter-Pixel Edge Representation

The ideas of the last section will be illustrated in a specific example. For this treatment we choose a different representation of edges among the pixels. Using the differentiation operator presented in the previous sections, the set of possible edges that can be associated with a pixel are shown in Figure 5(a); this would leave us with 5 possible labels at each point, LINE0°, LINE45°, LINE90°, LINE135°, NULL (no line).

Here we simplify this representation by only allowing horizontal and vertical edges to be placed in the image between pairs of adjacent points as in Figure 5(b). Rather than associating edges with pixels, we have an inter-pixel edge representation with the location and orientation of edges represented at a local level more naturally. This type of representation has some desirable characteristics and has been used elsewhere (Brice and Fenema [1970], Yakimovsky [1976], Prager, Hanson and Riseman [1976]).

There are now only twice as many possible edges as pixels (Figure 5b) compared to four times as many before (Figure 5a). However, we will view these edges quite differently. The results presented in earlier sections allowed the four types of edges about a pixel to compete, with only the strongest surviving. Figure 5(c) demonstrates why we do not wish to allow the horizontal and vertical edge around a point to be mutually exclusive--they both should be present for diagonal boundaries.[1]

This leads us to viewing each horizontal and each vertical edge in our current representation as a distinct element a_i in the set of

---

[1] Note that now higher level processes will be required to detect the global characteristics of a straight line at some orientation other than horizontal or vertical.

elements A. For each element there are only two labels to be associated with it, EDGE and NO-EDGE. Since the relaxation scheme that we have described demands that the probabilities of the labels sum to one, we have a situation which has simplified nicely, where only one probability, P(EDGE), need be stored to represent the likelihood of the two labels.

Before discussing the compatability coefficients and the manner in which the labels will be updated, we will adapt our differentiation operator of Figure 3 to the new situation. Figure 5(d) demonstrates the computation of the strength $S(E_i)$ of an edge E_i (in this case vertical) as the max of the output of the two masks which were associated with putting an edge in the given position. Now let us utilize the strength of the globally strongest edge in the image

$$SMAX = \max_{E_i \in \text{image}} S(E_i)$$

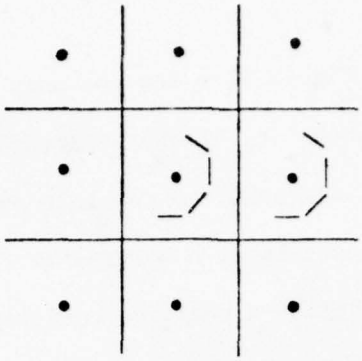
to convert each $S(E_i)$ into the probability of EDGE (and consequently determining the probability of NOEDGE) at location i by

$$P(E_i) = \frac{E_i}{MAX} \quad .$$

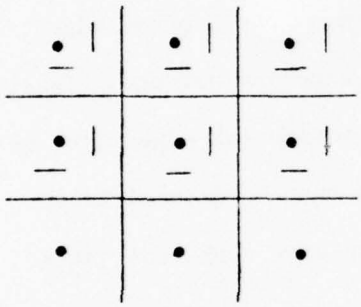
Thus, the probability of an edge will approach 1 only at the strongest edges in the image.

Only the specification of the compatability functions remain. We must define $r_{ij}: \Lambda \times \Lambda \longrightarrow [-1,1]$ to cause suppression of redundant lines and strengthening of weak or incorrect lines. Generally these are intuitively specified as heuristic weights. Let us consider the types of weights on the neighborhood of surrounding labels which should influence the likelihood of a horizontal label.

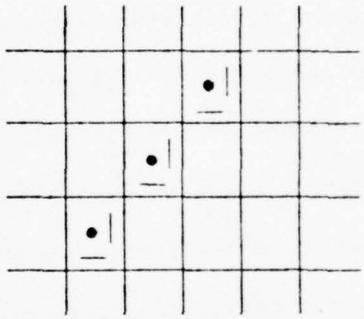
Figure 5: An inter-pixel edge representation for relaxation.
(a) Competing orientations of an edge for each pixel in the previous representation. (b) Both horizontal and vertical edges about a pixel will be allowed in the new representation. (c) For complete diagonal boundaries both horizontal and vertical edges at a pixel are required. (d) Modified edge operator is maximum strength of the two placements of masks. (e) The labels in the neighborhood of a horizontal edge which might be used to update the probability of a horizontal edge; note that the null label is depicted by □.



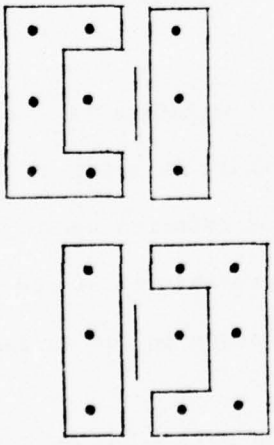
(a)



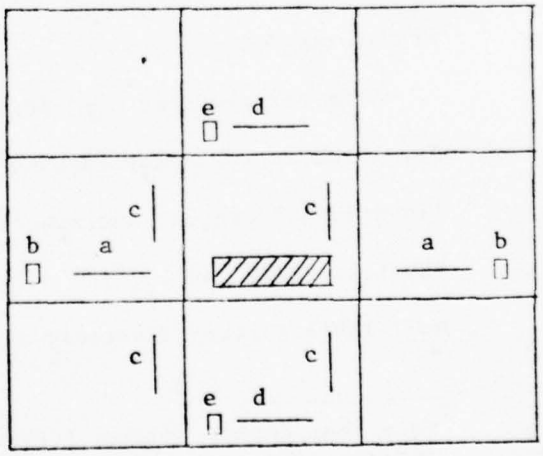
(b)



(c)



(d)



(e)

Figure 5

The labels shown in Figure 5(e) are the only labels that will be allowed to affect the probability of the horizontal label in the center of a 3×3 window. Horizontal edges "a" to the left and right of a horizontal label represent the continuation of a horizontal line and should support the likelihood of that label by a positive coefficient; the null label "b" left and right should have a negative weight because the edge doesn't continue. Vertical edges "c" should have positive weights since they represent a consistent extension of a horizontal edge. Horizontal edges above and below "d" call for suppression--hence a negative weight. Finally, the presence of a null label "e" above or below a horizontal edge might be considered as supporting evidence ($r_{ij} = +.3$) of that horizontal label and would be positive. The size of the weights employed represent one's heuristic estimate of the relative compatability of the label of point j on the horizontal label of point i. Specification of the vertical label can be derived by symmetry (a 90° rotation followed by a mirror image transformation).

The correlations for updating the null label can be heuristically specified in a similar fashion but it is difficult to specify as a set of linearly independent contributions.[1] We will address this question again shortly. Here, an $r_{ij} = 0$ on all points will cause the probability of the null label to vary inversely with positive or negative changes in the evidence

[1] Note that Zucker, Hummel & Rosenfeld [1975] deal with the null label by setting up a competing null label process. However, the desirable weights are only clear in areas where there is no evidence of strong edges anywhere in the local context. Thus, if multiple edges for a single boundary are allowed, the null label probabilities only need to grow in areas without edges. If one is trying to carefully refine the presence of a boundary to a single thinned edge representation, the features for increasing the probability of the null label cannot easily be expressed as a set of weights for a linear function.

of an edge. Thus, we have the means of computing a change in the probabilities of the horizontal or vertical labels based on the surrounding context, and by renormalizing obtain new probabilities of these edges.

Figure 6 are results of examples with different compatability coefficients and show various problems with the process as it has been formulated in this paper. In order to avoid a strong edge from being overwhelmed by the combined effect of a pair of weaker parallel edges to either side, reduction of strength of non-maxima edges by some factor k (in our example k = 2) will be applied prior to beginning the relaxation process; i.e., all edges parallel and adjacent to a stronger edge will be reduced. Figure 6(a) shows the resultant vertical and horizontal probabilities and an edge image with all edges with probability lower than .2 removed. It is clear that there are incorrect edges whose probability must be lowered while many vertical edges in the diagonal boundary should be increased.

Figure 6(b) shows an example set of coefficients and the results after 1 iteration, while 6(c) shows results after 6 iterations. The information cleans up with most of the vertical spurs hanging off the diagonal boundary in 6(a) being rapidly reduced. However, the major diagonal boundaries are missing key vertical edges whose probability has also been reduced. In addition the upper right diagonal boundary started with lower probability edges and they are in the process of disintegrating. In order to combat these effects, the size of some of the positive weights are increased in the example of Figure 6(d). The probabilities of edges after 6 iterations show many spurious edges growing stronger while parts of the weaker boundary still disappear.

Figure 6: The relaxation process for boundary formation.
(a) The initial probabilities of vertical and horizontal edges, and the location of edges with probability $\geq .2$. (b) An example set of weights and the probabilities after one iteration. (c) Shows probabilities after 6 iterations. (d) A set of coefficients which tend to grow more lines, and the results after 6 iterations.
(e) The addition of a feature which is a non-linear function of a set of points in the neighborhood, and the results after 6 iterations.

PROB. ARRAYS ON ITER. 0 - *INITIAL PROB.*

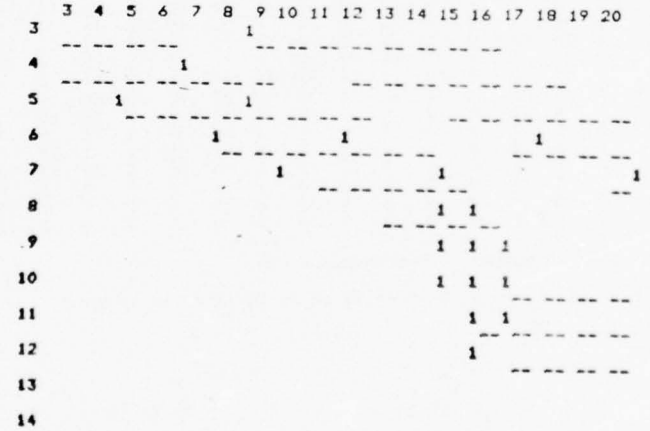
VERTICAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	13	10	5	3	7	21	10	9	8	17	7	17	5	1	0	0	2	0
4	10	10	12	26	11	5	4	8	19	9	19	8	7	7	18	4	0	0
5	11	26	12	11	12	26	13	9	3	5	8	19	9	9	9	8	7	7
6	1	0	6	13	28	12	10	12	26	13	9	2	6	9	21	9	8	8
7	0	1	3	1	1	8	26	12	12	11	14	37	13	2	3	7	9	20
8	0	0	3	0	1	3	1	2	8	14	18	26	64	15	1	5	2	4
9	0	1	0	0	0	1	5	2	5	1	9	35	80	30	5	0	0	2
10	0	0	0	0	2	6	3	1	5	1	1	24	94	35	9	0	0	3
11	0	1	0	1	1	5	2	2	1	0	2	13	54	26	9	5	2	3
12	1	0	0	0	1	5	1	3	1	1	2	6	28	7	4	2	2	18
13	2	0	3	0	0	2	0	1	3	0	4	0	1	3	1	0	1	8
14	1	0	0	0	1	0	1	0	0	0	1	0	1	1	4	10	2	10

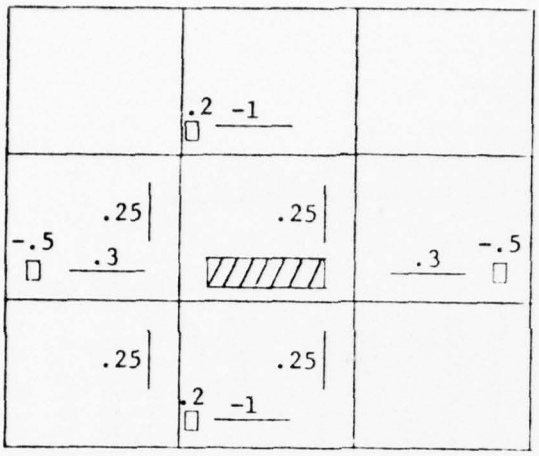
HORIZONTAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	95	93	40	28	13	15	22	27	29	59	66	63	26	20	11	4	0	1
4	36	46	98	94	45	38	25	11	17	24	28	30	60	63	28	23	16	9
5	5	13	25	38	94	97	98	45	36	23	12	19	24	27	59	61	66	28
6	4	5	3	4	15	29	41	96	96	99	44	31	14	12	20	25	29	58
7	1	2	3	4	4	4	7	19	33	44	100	86	67	16	5	8	14	21
8	3	0	0	1	2	3	5	5	5	8	22	36	28	38	5	1	3	7
9	0	2	3	5	6	8	3	4	4	4	3	6	16	10	10	4	4	4
10	3	1	1	1	3	2	3	2	2	3	2	3	5	18	31	33	33	32
11	1	3	1	1	1	2	2	5	8	8	8	8	19	49	85	94	93	89
12	0	0	0	1	0	0	0	1	2	3	3	3	2	11	22	28	28	29
13	1	0	0	0	0	0	0	0	0	0	0	1	2	2	2	1	1	4
14	3	3	3	2	0	0	0	0	0	0	3	4	7	6	7	1	1	3

GRAPHIC OUTPUT THRESHHOLD = .20



6(a)



PROB. ARRAYS ON ITER. 1

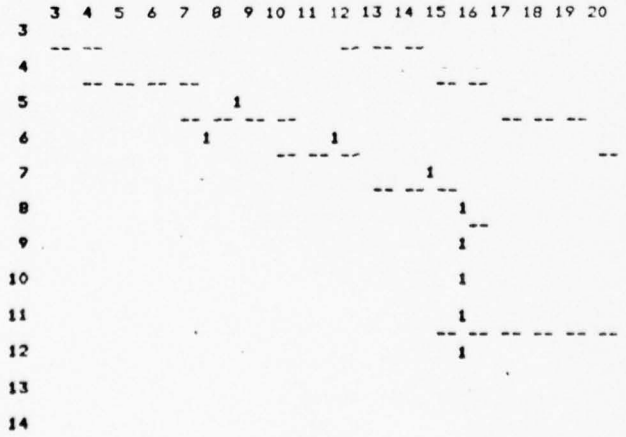
VERTICAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	8	8	4	3	5	15	6	7	6	13	4	10	2	0	0	0	1	0
4	9	11	9	19	5	4	3	4	12	5	15	7	5	3	10	1	0	0
5	4	18	9	12	12	24	7	6	2	3	5	12	6	7	8	6	4	3
6	0	0	3	6	23	11	11	10	23	7	6	2	3	4	15	6	6	7
7	0	0	1	0	0	3	18	8	12	11	10	32	10	1	2	4	4	15
8	0	0	1	0	0	1	1	1	5	9	13	16	64	0	0	2	1	2
9	0	0	0	0	0	0	2	0	2	0	2	0	81	0	0	0	0	1
10	0	0	0	0	0	2	1	0	2	0	0	0	94	0	2	0	0	1
11	0	0	0	0	0	2	1	0	0	0	0	0	57	13	7	4	2	3
12	0	0	0	0	0	2	0	1	0	0	1	1	26	5	3	2	2	18
13	1	0	1	0	0	1	0	0	1	0	2	0	0	1	0	0	0	4
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	0	2

HORIZONTAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	96	92	7	0	0	0	0	0	9	50	65	56	11	0	3	1	0	0
4	0	22	97	94	21	0	0	0	0	0	0	10	49	51	12	2	0	2
5	0	2	0	14	93	98	97	20	0	0	0	1	0	8	48	60	56	12
6	2	1	0	0	0	0	16	95	96	99	14	0	0	3	0	1	7	48
7	0	0	1	1	1	0	1	0	0	18	100	88	68	10	2	2	4	0
8	1	0	0	0	0	1	1	1	0	1	0	0	16	33	3	0	0	1
9	0	1	1	2	3	4	1	1	1	1	0	1	12	5	3	0	0	0
10	1	0	0	0	1	1	1	0	0	1	0	1	3	10	0	0	0	0
11	0	1	0	0	0	0	1	2	3	4	4	5	20	53	86	95	94	91
12	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	0	0	0	0	0	0	1	2	3	3	3	0	0	1

GRAPHIC OUTPUT THRESHHOLD = .20



PROB. ARRAYS ON ITER. 6

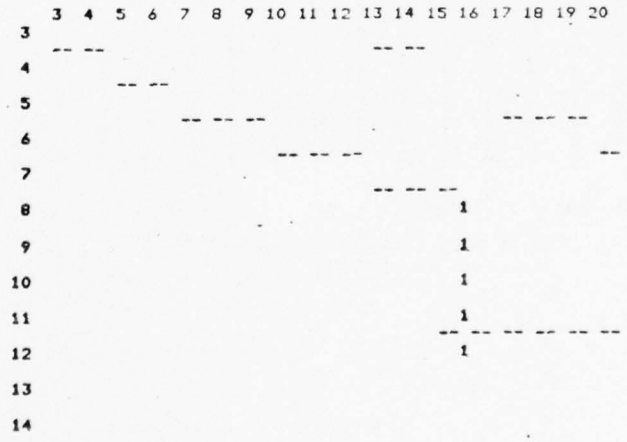
VERTICAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	1	2	0	0	1	3	1	1	1	3	0	0	0	0	0	0	0	0
4	0	7	1	3	0	0	0	0	0	0	2	0	0	0	0	0	0	0
5	0	3	1	6	2	12	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	11	2	6	0	15	0	0	0	0	0	2	0	0	1
7	0	0	0	0	0	0	3	1	9	3	1	8	9	0	0	0	0	2
8	0	0	0	0	0	0	0	0	0	0	5	0	91	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	93	0	3	2	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	46	0	1	1	0	10
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

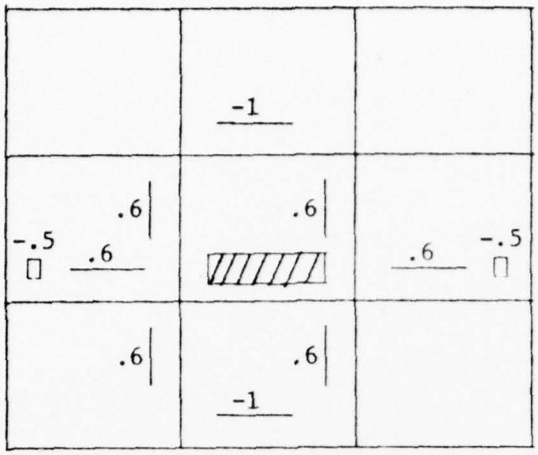
HORIZONTAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	99	91	0	0	0	0	0	0	0	12	48	29	0	0	0	0	0	0
4	0	0	99	98	0	0	0	0	0	0	0	0	18	19	0	0	0	0
5	0	0	0	0	97	99	99	0	0	0	0	0	0	0	46	81	55	0
6	0	0	0	0	0	0	0	98	99	99	0	0	0	0	0	0	0	43
7	0	0	0	0	0	0	0	0	0	0	100	99	93	9	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	40	91	99	99	99	99
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GRAPHIC OUTPUT THRESHHOLD = .20



6(c)



PROB. ARRAYS ON ITER. 6

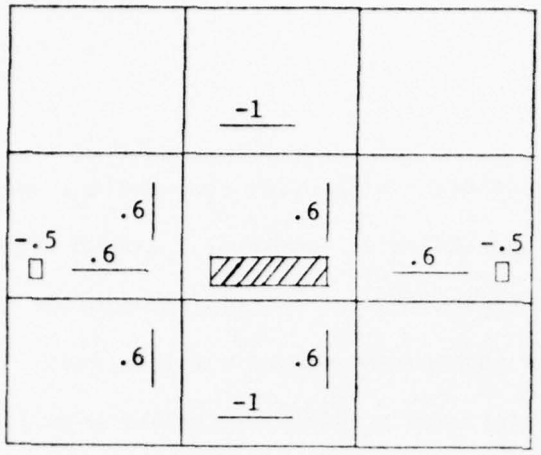
VERTICAL PROBABILITIES

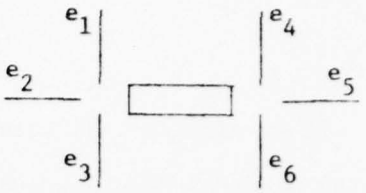
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	42	10	0	0	2	4	3	2	5	21	6	0	0	0	0	0	0	0
4	16	47	10	16	0	0	0	0	0	3	9	0	0	0	0	0	0	0
5	0	16	11	38	23	52	0	0	0	0	0	0	0	0	7	2	0	0
6	0	0	0	0	54	20	41	2	71	0	0	0	0	0	9	1	0	4
7	0	0	0	0	0	0	14	6	59	16	2	50	23	0	0	0	0	7
8	0	0	0	0	0	0	0	0	0	2	29	6	99	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	98	0	18	10	4	15
12	0	0	0	0	0	0	0	0	0	0	0	0	77	0	13	8	1	42
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HORIZONTAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	99	98	0	0	0	0	0	0	0	43	89	53	0	0	0	0	0	0
4	0	3	99	99	3	0	0	0	0	0	0	0	3	3	0	0	0	0
5	0	0	0	0	99	99	99	0	0	0	0	0	0	0	57	91	57	0
6	0	0	0	0	0	0	1	99	99	99	0	0	0	0	0	0	0	52
7	0	0	0	0	0	0	0	0	0	1	100	99	99	21	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	9	58	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	20	4	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	75	98	99	99	99	99
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GRAPHIC OUTPUT THRESHHOLD = .20





Prob(unconnected edge) =

$$1 - \text{MAX}[P(e_1), P(e_2), P(e_3)] * \text{MAX}[P(e_4), P(e_5), P(e_6)]$$

$$r_{\text{unconnected}} = -.5$$

PROB. ARRAYS ON ITER. 6

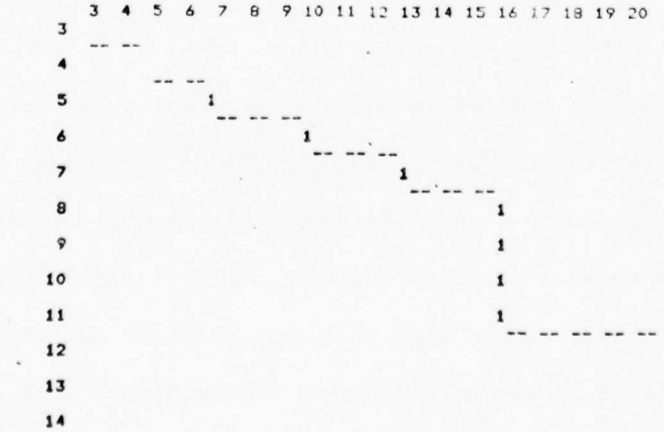
VERTICAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	22	0	5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	4	0	27	0	8	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	2	26	0	3	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	2	0	95	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	2
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HORIZONTAL PROBABILITIES

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	99	67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	97	93	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	92	99	98	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	93	99	99	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	100	99	96	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	3	89	99	99	99	99
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GRAPHIC OUTPUT THRESHHOLD = .20



6(e)

It is difficult to balance the effects of keeping the vertical edges in the diagonal boundary and the suppression of growth of spurious edges. Much of this problem is due to the limitations of using a function in which the points contribute in a linearly independent manner. Figure 6(e) shows the use of one additional factor, the probability that an edge is unconnected. For a given edge to be a part of a continuing boundary, there should be at least one high probability edge emanating from each end of our given edge. If the three possible edges from each side are called e_1, e_2, e_3 and e_4, e_5, e_6, respectively, then

$$P(\text{unconnected edge}) = 1 - \text{MAX}[P(e_1),P(e_2),P(e_3)]*\text{MAX}[P(e_4),P(e_5),P(e_6)].$$

If this probability is associated with a negative weight, it will keep spurious lines from growing off a strong edge into areas where there are only low probability edges. However, this factor is a non-linear function of the probability of six labels and is an extension of the theory as presented.

The result of using this negative contribution is shown in Figure 6(e). Now larger positive weights on other coefficients can be used. The results after 6 iterations show the desired effect with all edges in the major boundary growing stronger. However, the other diagonal boundary disappears because weak points within it caused it to appear disconnected and it broke up.

If the relaxation process is to just carry out gross strengthening of boundaries without worry about producing thick lines with multiple edges, it probably can be used quite reliably. However, if the goal is refined edges as we have been seeking here, then it appears that contributions from independent labels will be quite difficult to tune and contributions from sets of labels will probably be required. This area will be left for future research.

4.5 Grouping Edges into Line Segments

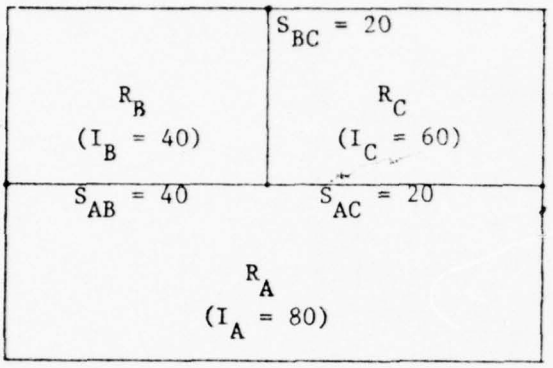
The problems have not been exhausted. Although the results in Figure 6 appear to be good segmentations at a macro-level, upon close examination there may still be incomplete boundaries, textural edges, and noise points.

If local edges which are a part of a common boundary are to be grouped into distinct line segments, then some criterion of similarity is needed. Certainly orientation is important when straight lines are being tracked, but in the general case this characteristic cannot be relied upon. If a pair of edges are of approximately equal strength, it is a strong cue that the edges should be joined. However, the regions surrounding any given region are bound to have different properties. Therefore, no matter upon what feature the strength of the gradient is based, one must expect widely varying values as the boundary of a single region is tracked. Figure 7a depicts three regions with the edge strength based upon intensity: the strength of two line segments S_{AB} and S_{AC} bounding region R_A are quite different.

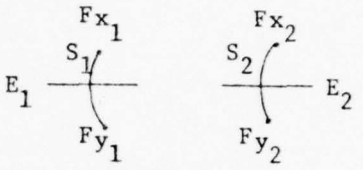
This problem calls for the goal of forming line segments each of which lies between only one pair of regions. Then, one can expect local edges to exhibit characteristics which have less variance. In addition, the comparison of features of the regions to either side of a pair of adjacent edges, Figure 7b, can be very useful in directing the edge binding process [Perkins [1976]). Notice that S_{BC} and S_{AC} which are equal in strength have the properties of R_C in common, but differing properties on their other sides (R_A vs. R_B) leads to opposite signs on the direction of the gradient. The similarity of two edges E_1 and E_2 can now be based upon much more complete information, a comparison of (Fx_1, Fx_2) and (Fy_1, Fy_2) as well as S_1 and S_2. Thus, S_{AC} and S_{BC} can be detected as distinct segments yet retain information that they bound a common region.

Figure 7: Use of region information in the grouping of edges.

(a) Three adjacent regions R_α, with associated intensities I_α, α = A,B,C, produce edge strengths S_{AB}, S_{AC}, and S_{BC}. (b) The features to either side of a pair of edges can be used to group the edges into boundary segments. (c) Edges are grouped and segments of boundaries are symbolically labelled with distinct numerical symbols. (d) Segments obtained across the entire image. (e) The remaining long segments after thresholding by length.


S_{BC} = 20
R_B
(I_B = 40)
R_C
(I_C = 60)
S_{AB} = 40
S_{AC} = 20
R_A
(I_A = 80)

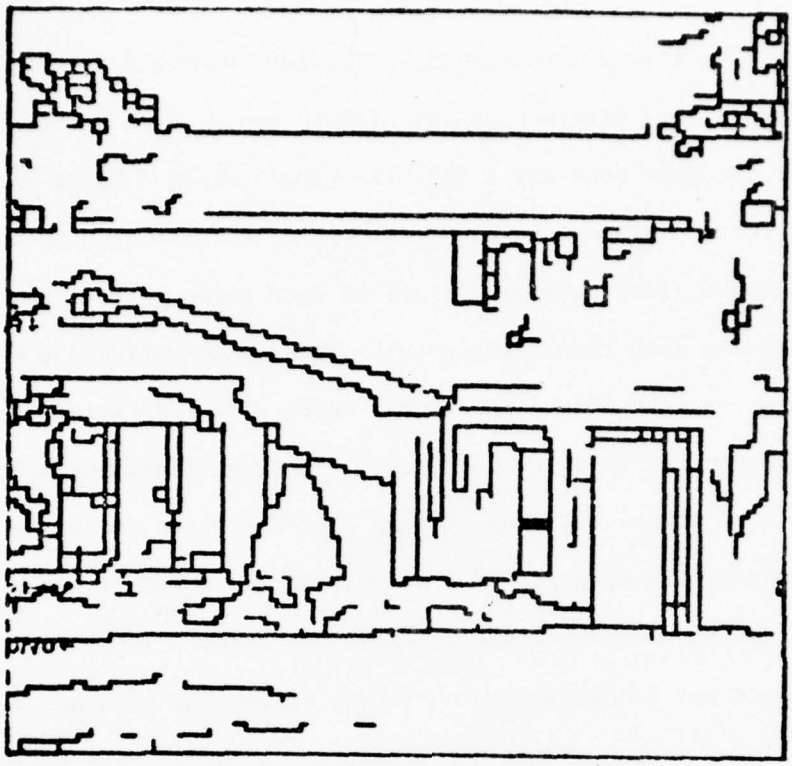
7(a)


Fx_1
S_1
E_1
Fy_1
Fx_2
S_2
E_2
Fy_2

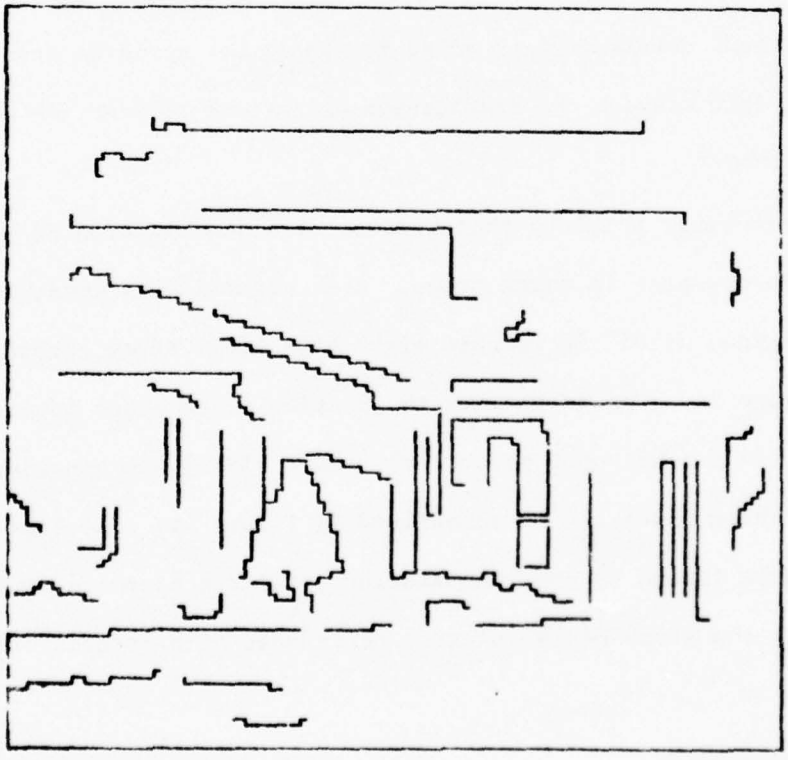
7(b)

BEST AVAILABLE COPY

7(c)



7(d)



7(e)

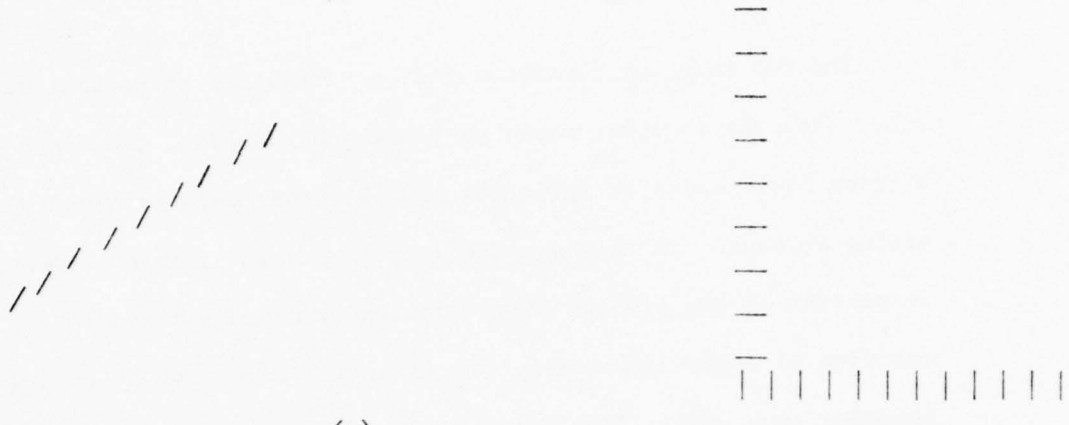
Typical results of grouping edges (Prager, Hanson & Riseman [1976]) are shown in Figure 7c. Distinct numeric labels are used to denote edges which are part of a common boundary. Note the places where a local variation caused a boundary to be divided into subparts. It is easy to join these back together by comparing the global average values of each segment (when there is not a vertex involving more than 2 segments). Then small variations will have little impact on long lines and context again allows decisions to be made that otherwise would be quite difficult. The result binding of edges produce the segments of Figure 7d; segments can be thresholded on the basis of length to obtain the most reliable boundaries as in Figure 7e. Further analysis of these procedures are available in Prager, Hanson and Riseman [1976].

It should not be difficult to utilize region and boundary information in an integrated manner within the relaxation process. The similarity of the regions associated with contiguous edges might be a weighting factor for the mutual support of the edges. This could be used to limit the mutual development of edges to those that would be grouped into a line and might prevent the aggregation of texture element edges into a spurious line.

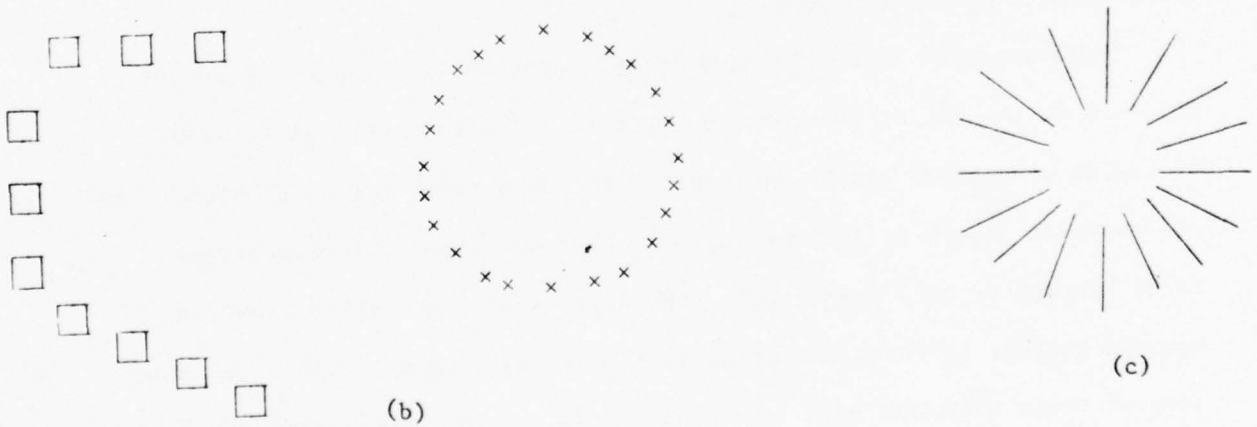
There are other problems that remain. The quantization of direction by the Kirsch operator is quite crude. A straight line segment whose slope is not a multiple of 45° increments might have local edges appearing as shown in Figure 8a. One is faced with grouping these edges into the slope of the line (as a continuous parameter). Marr [1975] has considered a similar problem which can be summarized by Figure 8b. The line to be detected can be formed by grouping similar primitive elements, which could be defined by the shape of the element or an edge of a certain orientation.

Figure 8c points out that an abstract line can be perceived by grouping a set of places where each place is specified by an element (a line, endpoint, or some other entity of arbitrary complexity) at some point in space. In addition, curves may have to be fit to any type of boundary; line or curve descriptors need not be restricted to the orientations of the detectors of small segments. All of these additional topics deserve careful treatment but will not be considered any further in this paper.

Figure 8: Additional problems in boundary formation discussed by Marr [1975]. (a) The *orientation* of global lines may be different than the orientation of the local edges being grouped. (b) Lines formed by grouping similar primitive elements. (c) An abstract line can be formed by grouping distinguished 'places', in this case the endpoints of other lines.



(a)



(b)

(c)

Figure 8

5. Region Formation

The two main approaches to region formation of natural scenes, other than the indirect route of forming boundaries, are based on either merging local areas or splitting global areas, both eventually determining regions. In this section we will examine some of the fundamental properties of the problem domain that have been utilized in a few specific examples of region formation. It is argued that most of this work has focussed upon either local features and two-dimensional spatial properties of the image or global features of the image, but that these different types of information have not been fully integrated. By integrating the types of feature activity in a scene with an analysis of their relative spatial distributions, local region formation can proceed under the guidance of a global analysis.

The discussion is complicated in some cases by the issues of semantic guidance in the region segmentation process. The lack of a global view of region properties can be compensated for by providing (the probabilities of) semantic labels to various regions, thereby allowing region merges to be blocked or made more likely. However, there is some controversy how and whether to bring semantics to the initial segmentation of a scene. Some of these questions will be considered in the approaches that use local spatial analysis and semantic guidance to merge small regions.

Let us examine three approaches to region formation. Recent articles (Zucker [1975], and Weiman [1976]) can provide the reader with additional approaches. The three efforts focus upon different characteristics of scenes:

1) Local spatial examination of the scene — This involves the merging of local areas under syntactic (comparison of visual features of the areas) or semantic guidance; regions are built up from small pieces which have a high probability of entirely belonging to a final goal region (Brice and Fenema [1970], Yakimovsky and Feldman [1973], Tenenbaum and Weyl [1975], and Tenenbaum and Barrow [1976], Barrow and Tenenbaum [1976]).

2) Global examination of feature distributions across the scene — Here, peaks and clusters of activity in one-dimensional histograms are used to threshold the scene and recursively split the image; large pieces of the image are broken down into smaller areas until there is a high confidence that they are homogeneous under the features of interest (Ohlander [1975], Tomita, Yachida and Tsuji [1973], Schachter, Davis and Rosenfeld [1975]); and

3) Interfacing spatial analysis with feature analysis — Clusters of activity in two-dimensional histograms are used to label local areas of the scene, followed by a spatial analysis of these labels to guide the formation of the desired regions (Hanson, Riseman and Nagin [1975]).

5.1 Region Growing via Local Analysis

There has been a range of work on techniques for locally merging areas. One can break any scene into 'atomic' areas by merging all adjacent points (either 4-neighbor or 8-neighbor adjacency) into the same region if they differ in some property by less than a threshold θ.

These algorithms are usually programmed to sequentially add points adjacent to a given region or point[1]. If θ equals 0, these areas are formed in the most conservative manner possible (although even here because of problems such as shadows one is not assured that these regions each lie entirely within an area encompassed by a single object). With only a little experience in region growing, it becomes obvious that there does not exist any single threshold for region merging that is acceptable, even for several different areas in a single scene.

Consider subimage B of Figure 1d which includes on the right side an area of sky above the somewhat speckled roof, and on the left side tree foliage (reflective highlights and shadows), as well as sky or roof showing through in some places. Figure 9 shows the results of region growing (using 4-neighbor adjacency) with two values of θ; the conservative value does not grow the tree together but a small increase (on a gray scale of 64 values) joins the roof to the tree. What is noise or textural variation in one area becomes a meaningful boundary in another. Thus, dynamic setting of thresholds is needed in the different areas, but that is a complex process to automate without global guidance or a priori knowledge. It is always difficult to determine whether or not a local discontinuity with respect to some feature(s) should bar further region growth or should be bridged as an internal variation of the region being formed. However, one meta-strategy is to form atomic areas far

---

[1] It is easy to formulate parallel region growing algorithms. In a spatial array processor such as the processing cones (described in section 3.3), every image point can act as an initial 'seed' point and all regions can grow simultaneously (with some being gobbled up by others).

too conservatively and then seek additional means of merging these areas (Brice & Fenema [1970]). This can be effective but it is very difficult to avoid all incorrect local merges; a single 'leak' between regions might cause very large changes in the final segmentation. Semantic constraints have been used to provide greater reliability.

Freuder [1976] provides an intersting variation to the region merging process by grouping those regions which are relatively more similar to each other than to other regions. This is continued and a tree of regions is constructed up to a single region over the scene. This whole structure would be passed to a global semantic processor which must extract the relevant information for different parts of the picture from nodes of the tree at varying levels of grouping. Potentially, this can be a powerful and flexible way to present information to semantic processes. However, it seems that the tree should be greatly pruned prior to semantic processing if it is to be useful. This leads to the difficult questions concerning texture that remain to be solved if this is to be a viable approach.

Figure 9: A simple region grower, where regions are represented by a unique symbolic label (mod 99).
(a) Regions growing on intensity values of subimage B with $\theta = 3$. (b) Regions grown with $\theta = 5$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	1	2	3	4	5	6	6	6	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
2	8	9	10	11	12	13	14	15	16	17	18	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	19	20	10	10	10	21	22	23	16	24	25	26	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	27	27	28	28	29	30	31	23	32	33	34	35	36	36	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	37	37	28	38	38	30	39	40	41	33	34	34	42	42	43	43	44	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	45	38	38	38	38	46	47	48	41	49	50	50	42	42	51	52	53	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	54	55	38	56	38	47	47	57	50	50	50	58	59	60	61	61	53	62	63	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	64	65	66	67	68	68	69	70	70	71	50	58	58	72	73	74	75	75	76	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	77	65	77	78	78	78	69	79	80	81	58	58	82	72	83	83	84	76	76	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	77	77	77	78	85	86	86	86	87	88	58	58	58	89	83	90	90	90	91	92	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
11	93	93	77	94	95	96	97	86	86	86	58	58	98	0	1	90	90	90	2	88	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
12	94	93	85	85	86	96	87	88	88	89	90	91	91	92	91	93	94	94	95	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
13	84	96	97	97	98	98	0	88	89	89	89	1	91	91	91	2	91	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
14	98	98	98	98	98	98	98	98	6	7	1	1	91	91	91	91	91	91	8	9	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
15	98	9	98	98	9	10	11	12	13	14	1	1	1	91	91	91	91	91	91	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
16	15	9	9	9	9	9	12	12	16	16	16	91	91	91	91	91	91	91	91	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
17	9	9	9	9	17	18	12	12	16	16	16	16	91	91	91	91	19	20	21	91	91	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
18	9	9	9	9	22	12	12	12	16	16	16	16	23	23	24	24	24	25	26	91	27	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
19	9	28	9	29	29	30	31	12	32	16	33	16	34	23	35	35	35	36	37	37	38	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
20	9	9	9	29	39	40	41	12	42	43	33	16	23	23	44	44	45	36	37	37	46	47	8	8	8	8	8	8	8	8	8	8	8	8	8	8
21	9	48	9	29	39	40	41	42	42	49	50	51	23	23	44	44	52	52	52	52	52	53	54	8	8	8	8	8	8	8	8	8	8	8	8	8
22	9	55	55	29	39	40	41	42	42	51	51	51	56	57	58	58	52	52	52	52	53	53	53	59	59	8	8	8	8	8	8	8	8	8	8	8
23	9	9	55	60	39	42	42	42	42	51	51	51	56	57	61	61	62	52	52	52	63	64	53	53	59	59	8	8	8	8	8	8	8	8	8	8
24	65	66	67	60	68	68	69	42	42	51	51	51	51	51	51	51	70	70	71	64	64	53	53	8	8	8	8	8	8	8	8	8	8	8	8	8
25	67	67	67	69	68	68	42	42	51	51	51	51	51	51	51	51	72	71	71	71	73	74	8	8	8	8	8	8	8	8	8	8	8	8	8	8

9(a)

BEST AVAILABLE COPY

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	1	2	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
2	1	1	5	5	6	6	7	7	8	8	9	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	10	11	5	5	5	12	13	7	8	14	15	16	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	17	17	17	17	17	18	19	7	7	20	21	22	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	17	17	17	23	23	18	24	25	26	20	21	21	27	27	27	27	27	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
6	28	23	23	23	23	29	30	31	26	32	33	33	27	27	34	34	27	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	35	36	23	37	23	30	30	33	33	33	33	33	33	33	38	38	27	39	40	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
8	41	35	36	42	43	43	44	45	45	33	33	33	33	33	46	47	48	48	49	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
9	50	35	50	43	43	43	44	51	45	33	33	33	33	52	33	53	53	49	49	49	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
10	52	50	50	43	43	43	43	43	53	33	33	33	33	33	33	53	54	54	54	54	55	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
11	56	56	50	57	57	58	58	43	43	43	33	33	33	33	59	54	54	54	54	54	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
12	60	56	61	61	61	58	62	43	43	63	64	33	33	33	33	65	66	66	67	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
13	60	56	56	56	56	56	43	43	63	63	63	33	33	33	33	68	33	69	69	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70	70
14	55	56	56	56	56	56	56	56	71	63	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
15	56	56	56	56	56	56	72	56	73	74	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
16	56	56	56	56	56	56	56	56	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
17	56	56	56	56	56	56	56	56	33	33	33	33	33	33	33	33	33	75	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
18	56	56	56	56	56	56	56	56	33	33	33	33	76	76	77	77	77	78	78	33	79	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
19	56	80	56	56	56	56	56	56	56	33	81	33	76	76	82	82	82	83	83	83	83	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
20	56	56	56	56	84	56	56	56	56	85	81	33	76	76	86	86	86	83	83	83	83	87	33	33	33	33	33	33	33	33	33	33	33	33	33	33
21	56	56	56	56	84	56	56	56	56	83	81	86	76	76	86	86	86	86	86	86	86	87	33	33	33	33	33	33	33	33	33	33	33	33	33	33
22	56	56	56	56	84	56	56	56	56	86	86	86	86	89	89	89	86	86	86	86	87	87	87	33	33	33	33	33	33	33	33	33	33	33	33	33
23	56	56	56	56	84	56	56	56	56	86	86	86	86	89	86	86	90	86	86	86	87	87	87	33	33	33	33	33	33	33	33	33	33	33	33	33
24	91	91	56	56	56	56	56	56	56	86	86	86	86	86	86	86	86	86	86	86	87	87	87	33	33	33	33	33	33	33	33	33	33	33	33	33
25	56	56	56	56	56	56	56	56	86	86	86	86	86	86	86	86	86	86	86	86	92	93	33	33	33	33	33	33	33	33	33	33	33	33	33	33

9(b)

5.2 Merging Regions Under Semantic Guidance

The focus of this paper has been upon techniques that can be applied independently of the semantic context in which the computer vision system is operating. In many systems stored models are used to match and then refine noisy and incomplete segmentation. Our general position was outlined in the introduction--that it is desirable to perform an initial segmentation without use of semantic information. However, there are many opportunities to use such knowledge in the kinds of processes we have been examining. This section outlines a couple of the more general attempts to integrate segmentation and interpretation by controlling the merging of atomic areas.

The decision-theoretic approach for image interpretation of Yakimovsky and Feldman [1973] addresses the difficulties previously outlined by introducing semantics in a decision-theory framework. Their segmentation process is based on merging atomic areas if the probability of a global interpretation is improved:

$$P\{\text{global interpretation} \mid \text{context, measurements}\}$$

$$= \prod_{i} P\{R(i) \text{ is } INT(i) \mid \text{values of measurements on } R(i)\}$$

$$* \prod_{B(i,j)} P\{B(i,j) \text{ is between } INT(i) \text{ and } INT(j) \mid \text{measurements on } B(i,j)\}$$

where R(i) is region i, B(i,j) is the boundary between regions i and j, and INT(i) is the semantic interpretation of region i. There are several important points to note. The identities of regions are assumed to be independent of each other except for the relationship across the borders; borders are also assumed to be independent of each other and depend only upon regions to either side. These assumptions seem to be reasonable approximations for local region interpretation. However, when a roof can appear on either side of a tree, such an assumption fails. Of key importance, though, is that semantic information is introduced at the segmentation level--regions can be merged if they improve the meaning of the partitioned scene. The boundaries between pairs of regions are linked into the region analysis, influencing the segmentation and interpretation processes. Excellent results were obtained on several road scenes and chest x-rays.

This approach integrates the segmentation and interpretation phases. It also captures the flavor of the more recent relaxation schemes by allowing a local hypothesis to be influenced by the context of other local hypotheses. However, it loses a parallel computational flavor for determining a model because decisions for merging regions are carried out sequentially. Yakimovsky and Feldman avoid an exhaustive sequential search for the best global interpretation by approximating the best interpretation as a result of a heuristic search.

This leaves the extremely difficult problem of the determination of the probabilities for this scheme. The probabilities for interpreting R(i) seem feasible but the relationships of boundaries in an inherently

three-dimensional world often vary uncontrollably. Although one knows that sky is virtually never below ground (actually a protrusion on a mountain or a change in one's viewpoint would allow this), one cannot fix the probability of seeing a car roof adjacent to grass without having a very restricted micro-world. On the other hand, the inherent two-dimensional spatial relationships of a chest in an x-ray photograph are relatively easy to approximate. In the most general applications their approach might have less difficulties at later stages of processing after initial segmentation.

Tenenbaum and Weyl [1975] present a detailed analysis of a range of strategies including syntactic and semantic merging criteria. The simplest nonsemantic measures involve comparisons of average differences in properties of local areas immediately to either side of the common boundary (one of the techniques employed by Brice and Fenema [1970] in the fundamental early work on region growing) or average properties of the entire areas. The two regions with the weakest boundary are merged and this process is repeated. All the algorithms performed many correct merges, but a few bad merges ('leak' of one region into another) can produce disastrous consequences. Another difficulty is the lack of meaningful stopping criteria for the algorithms.

Tenenbaum and Barrow [1976] demonstrated that the interactive human semantic labelling of regions could be used to block most erroneous merges made by nonsemantic rules. They interactively supplied labels of identities to initial conservatively formed atomic regions whose size is greater than some threshold θ_2. Then, an attempted merger of two regions with differing labels can be blocked, while the merger of an unlabelled region with a labelled region will inherit the available label, and finally the merger of two unlabelled regions will remain unlabelled. For those unlabelled regions that grow larger than θ_S, the human again supplies the proper label. For a simple office scene and outdoor scene, the final results are quite reasonable when θ_S is set so that about 20 regions are labelled during this process.

This approach led Tenenbaum and Barrow to employ a generalization of Waltz's [1975] constraint satisfaction approach on the region labels. Constraint satisfaction can be viewed as a special type of relaxation procedure where relationships between labels in a local context can be used to eliminate some of the alternative labels. They extend the semantic region merging process by alternating this merging process with the propagation of semantic constraints on the identity labels. For this approach to be automated it requires the initial labelling of all elementary regions (even individual picture elements!) and the specification of computationally effective procedures to extract the semantic relationships between regions.

However, the degree to which one can satisfactorily label the possible interpretations of a small section of an object on the basis of purely local information is still uncertain; with a large number of possible objects this problem may be serious. The authors demonstrate examples

with this labelling supplied manually or directed via pre-defined geometric models. The results are quite interesting, but the extensibility of this approach to automatic segmentation of general scenes seems to be quite difficult. Discussion of these problems is presented in a bit more detail in Arbib and Riseman [1976].

When one examines the effort and ingenuity involved in trying to keep one region from leaking into another, one can only conclude that better nonsemantic data will be required to guide segmentation. In particular, it suggests the need for analysis of feature activity in the context of the area under consideration. For merging purposes, one can only determine relative similarity of atomic areas in the context of the characteristics of atomic areas in the vicinity. This consideration

leads to the more global feature analysis of the second major approach.

5.3 Region Formation via Global Feature Analysis

This approach is based on the premise that the global distribution of feature activity in a scene contains sufficient information for segmentation of major areas. If two regions have a distinct difference in intensity one would expect the intensity histogram to form major peaks about their respective means. Figure 10 is a set of one-dimensional histograms for subarea B in Figure 1d and in certain places they have a multimodal distribution of the type expected. One can try to form the desired regions by separately turning on all image points in one or another of the clusters. Automatic determination of cluster boundaries based on histogram peaks may be simple or difficult depending on the particular case. If one examines cluster 2 of Figure 10a, and the points that are turned on in the image (Figure 11a), it appears that this approach works nicely.

Ohlander [1975] developed a technique of recursively partitioning an image by setting thresholds at valleys of 1D-histograms of various features. The first partition will form around the clearest peak in any histogram; then, the associated points in the image are turned on and adjacent points with the same label can be merged into a region by growing on the symbolic labels; these regions are smoothed by blurring, and each of these distinct regions will be the basis for further analysis by histograms. A region is kept intact only when it is unimodal in all histograms employed. In order for this process to work, Ohlander subtracts out 'busy areas' of texture and smaller detail by using a measure of the amount of edge

Figure 10: One-dimensional histograms of subimage B with a few possibly useful clusters marked.

(a) Hue is represented on a scale of 0 to 120 representing θ of Figure 2e varying from 0 to 360°; therefore, red = 0 and 120, green = 40, and blue = 80. (b) Saturation is represented on a scale from 0% to 100%. (c) Intensity is represented on a scale from 1 to 96 obtained by $[(R + 1) + (G + 1) + (B + 1)]/2$.

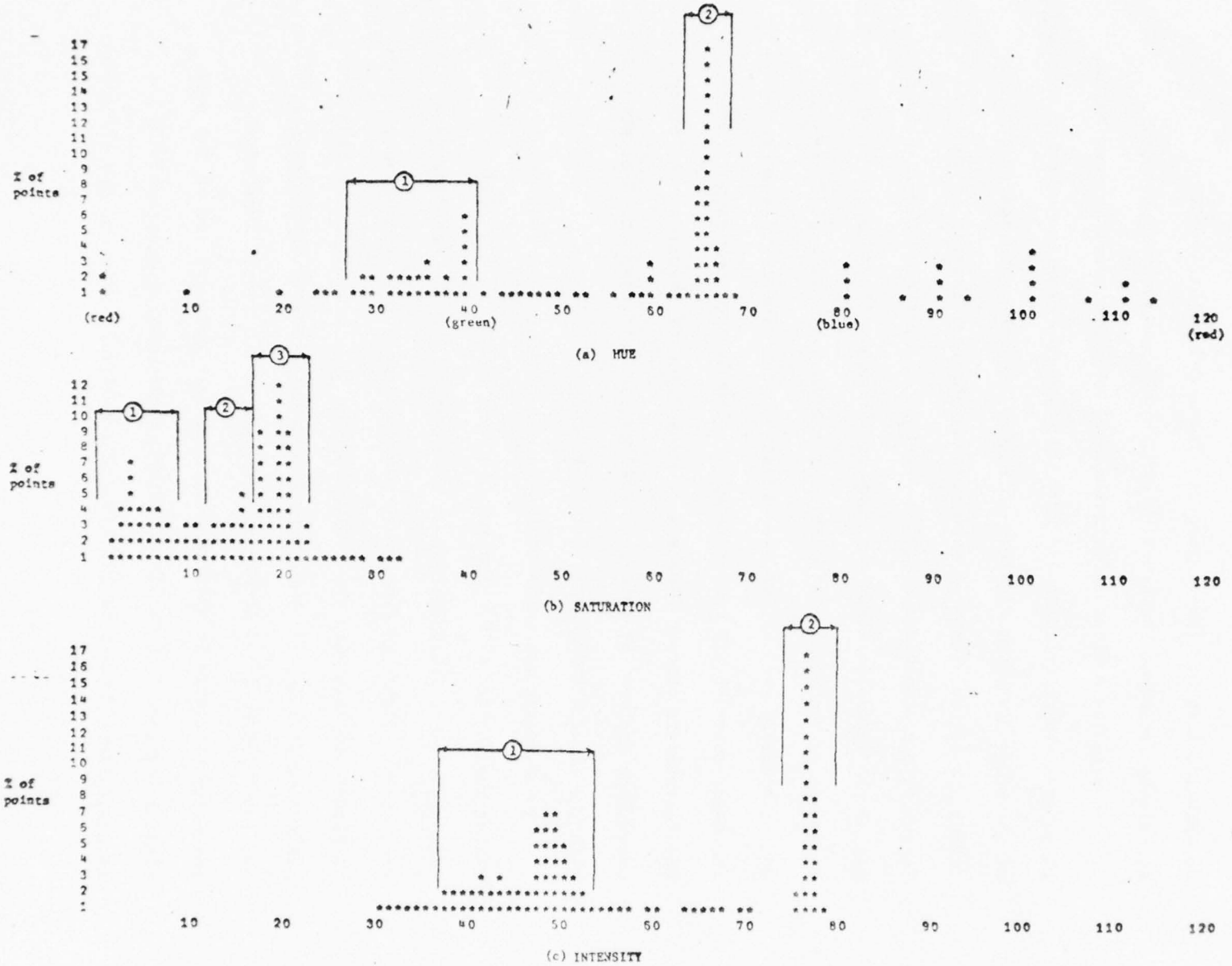


Figure 10

BEST AVAILABLE COPY

in each local area. These areas are processed by different techniques including the blurring operation previously mentioned.

Despite the obvious effectiveness of this procedure in some cases, there are several deficiencies with this type of histogram analysis. Consider some of the other histogram clusters in Figure 10--the peaks and cluster widths are not so clear. A more serious problem, though, is that different objects can have partially overlapping distributions in one or all of the features. This can cause peaks and valleys to appear and disappear as the particular combination of objects is varied, despite the possibility that all of the objects appear visually distinct to the human observer. A companion difficulty is that one cannot expect HSI features to produce clusters in many types of texture. The blurring operations employed by Ohlander will not be sufficient to deal properly with the general characteristics of texture.

These points are emphasized when one examines the rest of the HSI histograms in Figure 10. The two clusters in Figure 10a produce a reasonable first approximation in delineating the sky area and tree area (Figure 11a). However, when one maps the three saturation clusters of Figure 10b back onto the image (Figure 11b), one finds the clusters associated with sky (3) and roof (1) are interspersed in the tree with the last cluster (2). It begins to appear quite messy. Use of the intensity histogram is also poor in that tree and roof are in the same cluster together (Figure 11c). Although the other histograms will separate these regions in the recursive splitting process, the formation

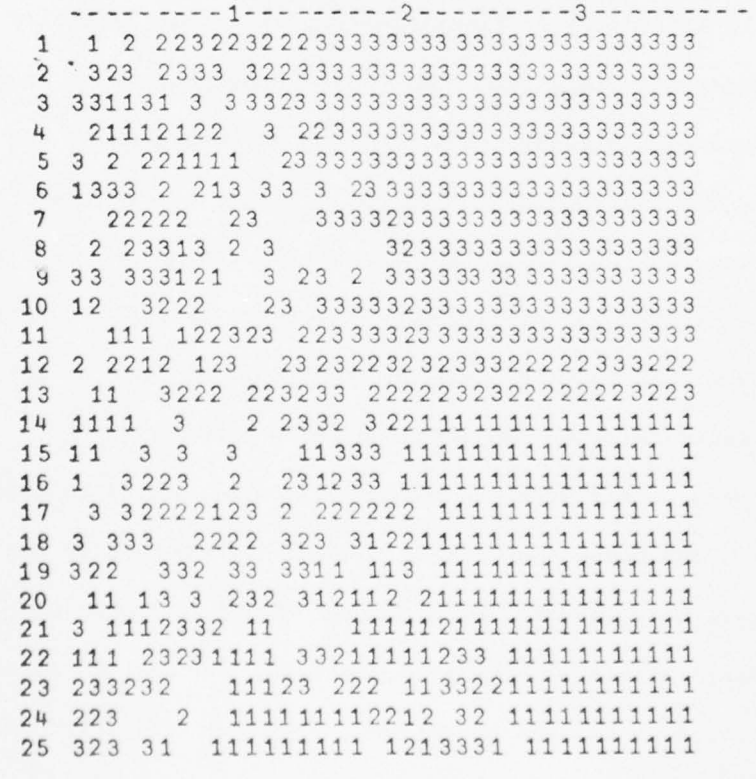
of the tree as a distinct region will not occur because the hue mapping will be split by the saturation mapping.

One can hope that the sequential determination of largest regions can be used to continually subtract away the data which obscures the presence of less noticeable peaks. However, the quality of this algorithm seems to be subject to an arbitrary condition, namely the particular mix of regions being examined. This problem would probably be reduced if the image were broken into smaller areas; this can be thought of as a foveal window where the system initially focusses in a directed manner upon a subarea of the entire scene in far more detail. Similarly, the peaks would have less chance of being obscured if multi-dimensional histograms were employed (although then the detection of peaks and clusters is less straightforward). Figure 12 depicts possible difficulty in discriminating different intensity and hues with one-dimensional histograms. One might hope that other features would detect differences in these cases. Of course this problem can occur in 2D histograms and require one to go to higher dimensional spaces, but at least with 2D histograms pairwise dependencies are available and this might be sufficient. But there is a still more significant drawback that must be overcome; that is, the lack of information on the spatial relationships of the features being examined.

When a histogram of a feature based on individual points is used to form a region in the manner described, spatial information is employed during this analysis only in terms of adjacency of points which have similar labels. On the basis of global histogram analysis, one cannot determine the difference between a red area bordering a yellow area and red polka

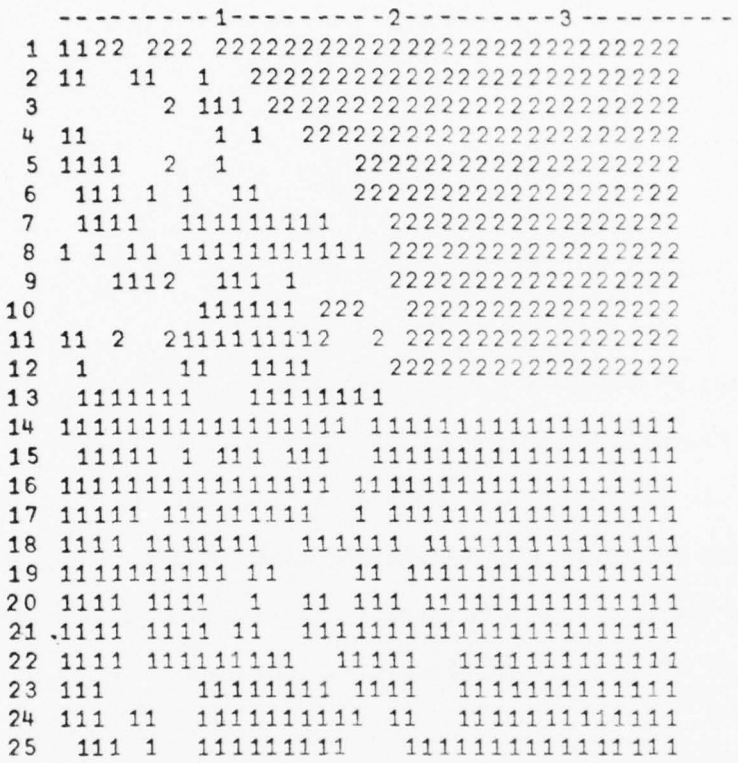
---------1---------2---------3---------
1 22 2222222222222222222222222222
2 1 2 2 22222222222222222222222
3 2 2 222222222222222222222
4 1 2 2222222222222222222222
5 1 1 2 22222222222222222
6 111 1 2 22 22222222222222222
7 1 11111 2 222222222222222
8 1111 11 1 22222222222222
9 11 11 111 2 2222222222222222
10 1 1 111 11 1 222 2 2222222222222222
11 1 11 1111111 22222222222222222222
12 1 11 111111 22222222222222222
13 1 111111 11 1 2 2
14 1 11111111111 11 1
15 11111111 11 1111
16 11111 1111111 1111 11
17 11111 11 11111111111
18 1111111 111 111111
19 11 11111 11111
20 1 11111111111 1
21 1 1 11 1 111111
22 1 1111 1 1111 11 1
23 1111111 1 1 1 11 11
24 111 1 1111111111
25 111111 111 11

(a)


---------1---------2---------3---------
1 1 2 223223222333333333333333333333
2 323 2333 322333333333333333333333333
3 331131 3 33323333333333333333333333
4 21112122 3 223333333333333333333333
5 3 2 221111 233333333333333333333333
6 1333 2 213 33 3 233333333333333333333
7 22222 23 333323333333333333333
8 2 23313 2 3 32333333333333333
9 33 333121 3 23 2 333333 33 3333333333
10 12 3222 23 333332333333333333333
11 111 122323 223333323333333333333333
12 2 2212 123 23 232232323332222233322
13 11 3222 223233 22222323222222223223
14 1111 3 2 2332 3 2211111111111111111
15 11 3 3 3 11333 1111111111111111 1
16 1 3223 2 231233 11111111111111111
17 3 32222123 2 222222 1111111111111111
18 3 333 2222 323 3122111111111111111
19 322 332 33 3311 113 111111111111111
20 11 13 3 232 312112 2111111111111111
21 3 1112332 11 111112111111111111111
22 111 23231111 33211111233 11111111111
23 233232 11123 222 113322111111111111
24 223 2 111111112212 32 11111111111
25 323 31 111111111 1213331 1111111111

(b)

Figure 11



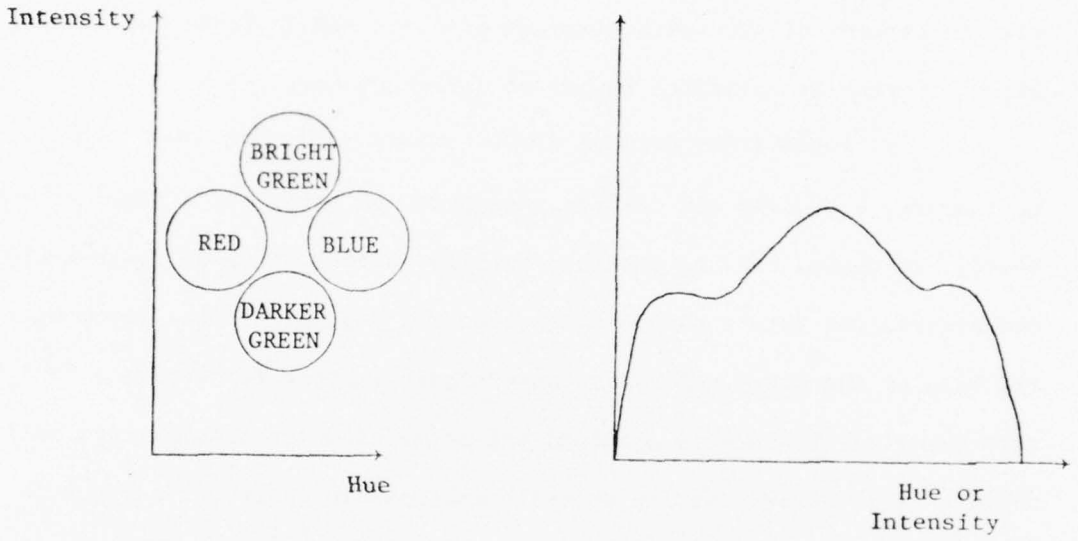
(c)

Figure 11

Figure 11: Projection of 1D cluster labels from Figure 10 back onto the points in the image.

(a) Hue, (b) Saturation, and (c) Intensity.

Figure 12: Potential problems with one-dimensional histograms. (a) Assume four clusters of activity in a 2D histogram of hue vs. intensity; the peaks of activity are assumed to be at the center of each circle and topologically are on the axis coming out of the paper. (b) A 1D histogram of either hue or intensity will obscure the clusters.


Intensity
BRIGHT GREEN
RED
BLUE
DARKER GREEN
Hue
Hue or Intensity

(a) (b)

Figure 12

dots within a yellow area--they can produce identical histograms and the difference in structure is not seen. More generally a texture can be composed of a set of micro- or sub-texture elements. Each micro-texture i could contribute toward a macro-texture with a given spatial distribution p_i. These distributions might tend towards spatial randomness or they might form more structured geometrical relationships. The relative occurrence of each micro-texture type can remain fixed and still allow a virtually unlimited number of different textures.

If many local areas possess similar characteristics, this is a cue to texture, e.g., sky and foliage, newsprint on paper, or a simple checkerboard. It should thus be possible to bind various types of micro-texture characteristics into a single macro-textured region. If one turns on either the blue or the green patches of sky-foliage texture for separate analysis, only a partial texture is obtained and other kinds of problems are introduced. It is clear that the spatial relationship of these features is a fundamental aspect of this texture. Blurring to smooth these regions and make them homogeneous is an alternative, but this produces its own problems and does not get at the basis of texture. Our solution for effective region growth calls for utilizing the strengths of both approaches, integrating global feature activity with a local spatial region growing process.

5.4 Integrating Spatial Analysis with Global Feature Analysis

The scheme we present for binding feature histograms to spatial relationships in the image is described in more detail by Hanson, Riseman, and Nagin [1975], and bears some similarity to a previous investigation by Tomita, Yachida and Tsuji [1973]. In this approach, histograms of

various feature pairs are employed to find clusters of feature activity. The system restricts its attention to a two-dimensional feature space because the analysis of the histograms can be carried out by operating on them as pseudo-images in the processing cones described earlier (since the cone is essentially a general 2D array processor). The algorithms assume the existence of a process which can dynamically select relevant features and make them 'active'. The point has already been made that the composition of the relevant features varies with the situation and with the movement of a fovea if one exists in the system. A low-level system will need a front end for this selection process and it would have to be interfaced to the several other types of algorithms that we have presented.

Each feature is computed as a function of a local window of some size (the windows may be overlapping or non-overlapping). As the size of this window increases, the quality of the information changes depending on the situation. If the window is centered entirely within a region, then the average and variance computed over the local area becomes statistically more meaningful as the size increases. On the other hand, as the size increases it becomes more and more likely that the window will overlap different regions and a 'mutant' value for the average and variance will be produced. This is the window problem that we have already mentioned, and we shall see shortly that this type of noise causes problems.

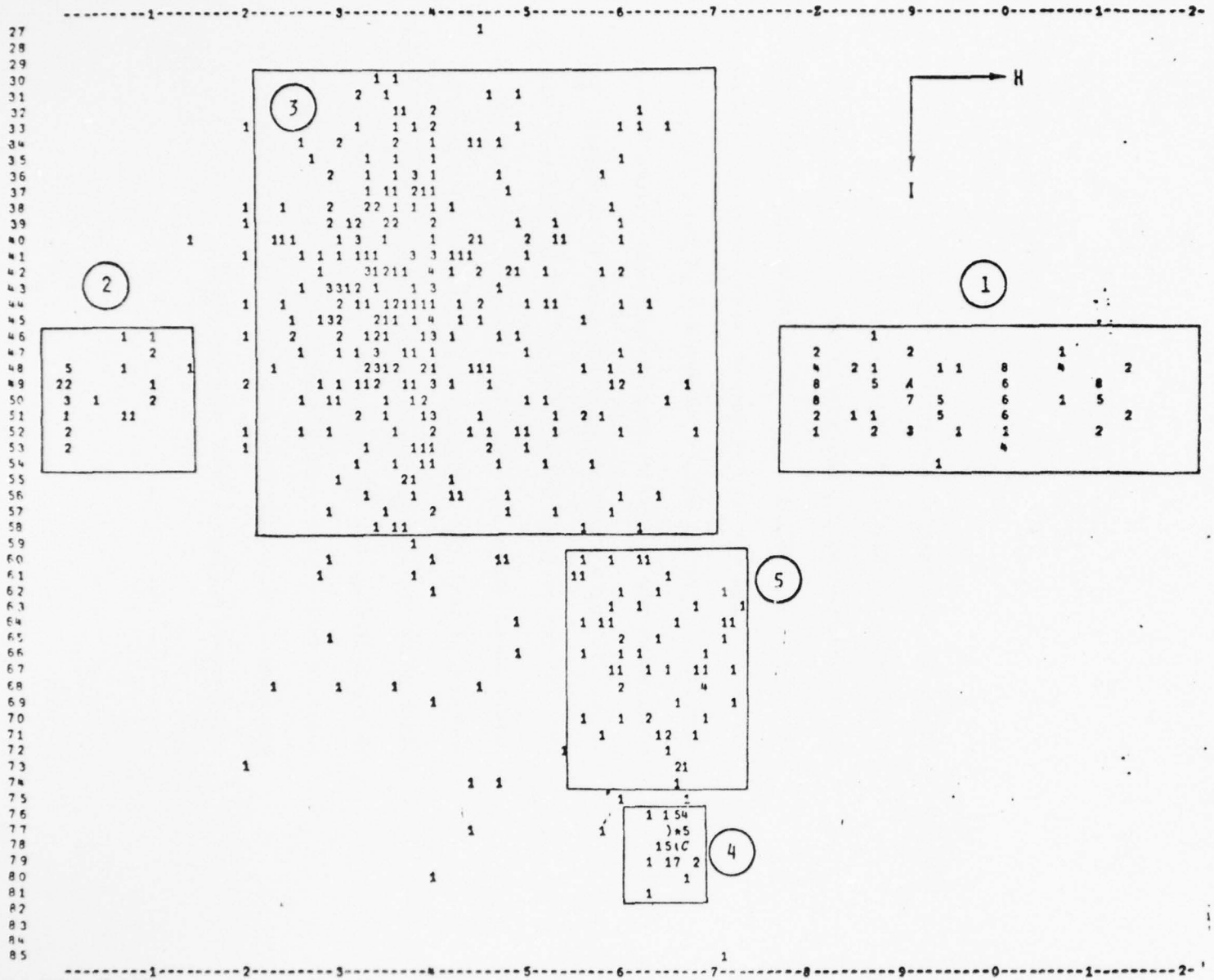
Figures 13 and 14 are two simple examples of two-dimensional histograms of subarea B (roof, tree, and sky) of our example scene, in particular hue vs. intensity (H vs. I), and intensity vs. saturation variation (I vs. S_V). Despite the noise it is quite clear

that several major clusters in the different feature spaces are correlated with the color image. The sky is bright blue (cyan) and the roof a speckled red; these areas have little variation in intensity and saturation. The tree area has a hue of various greens, blue, and white; saturation and intensity variation is high because the surface is irregular and there is large variation between figure and background sky showing through.

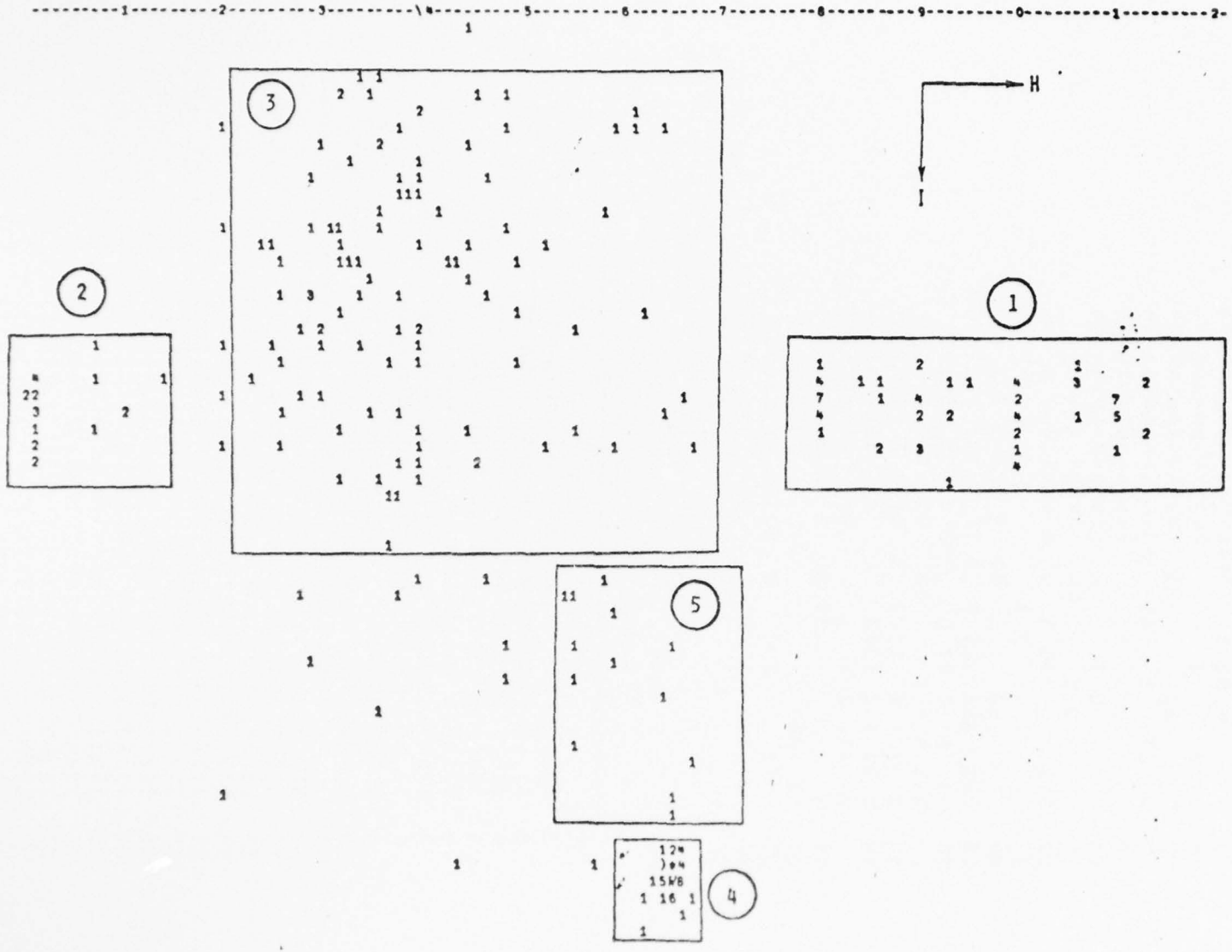
Useful information appears in all of the histograms and the clusters in the different histograms are certainly correlated. A low-level system should extract these dependencies and use the redundancy to increase the confidence in the results of this processing. Although some clusters are quite clear and easy to extract, others are more amorphous and widespread. The clarity and definition of a cluster may be aggravated by the window problem. A window that overlaps two adjacent homogeneous regions will produce a value in between these clusters when computing the mean of a feature, and false activity in the case of variation of a feature. This can produce trails between clusters.

In order to reduce the impact of the window problem on histograms, the technique of non-maxima suppression discussed in Section 4.2 can be utilized. In this application we have chosen to generalize it to a process of non-extremum suppression: values which are not local minima or local maxima in 8-adjacency neighborhoods will not be allowed to contribute to the histogram. This will have the effect of suppressing much of the trails between clusters associated with smooth regions. As the window moves across a boundary, the values of the means of a window will be changing from min to max (or vice versa), while the values of variations will increase and then decrease so that only a

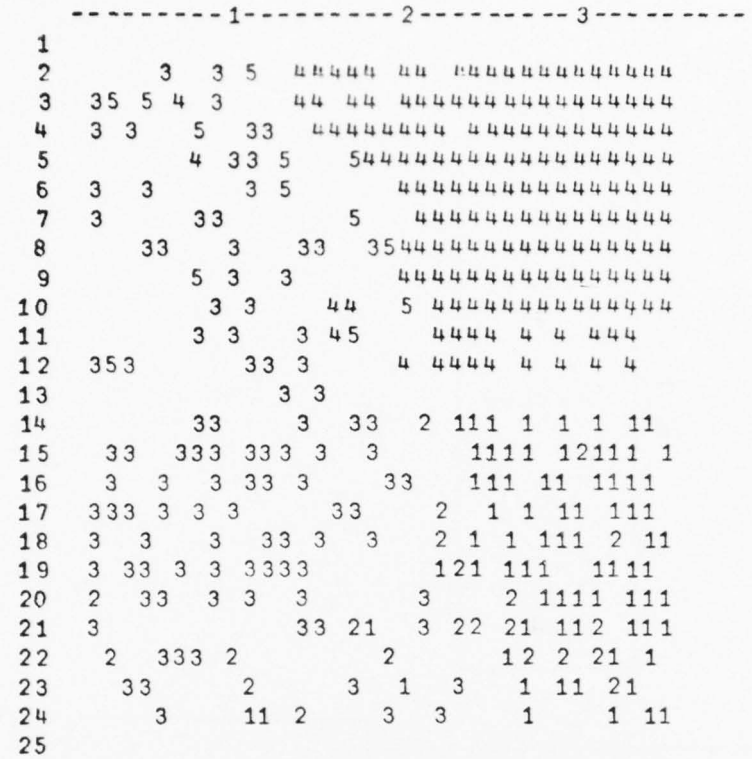
Figure 13: Use of two-dimensional histograms to guide region formation in subimage B - Example 1, Hue vs. Intensity.
(a) A two-dimensional histogram of hue vs. intensity with each character representing the number of points having the associated values of H and I on the respective axes; note that alphabetic characters represent values between A = 10 and Z = 35, punctuation and special characters represent values between 36 and 45, while a period represents all values greater than 45. (b) Non-extremum suppression is used to suppress image points whose values are not local minima or local maxima. Analysis of both the unsuppressed and suppressed histograms might allow the delineation of the clusters which are enclosed in rectangles in the two histograms. (c) Projection of clusters in the suppressed histogram back onto the image with suppressed points appearing as blanks. (d) Projection of the unsuppressed histogram back onto the image.



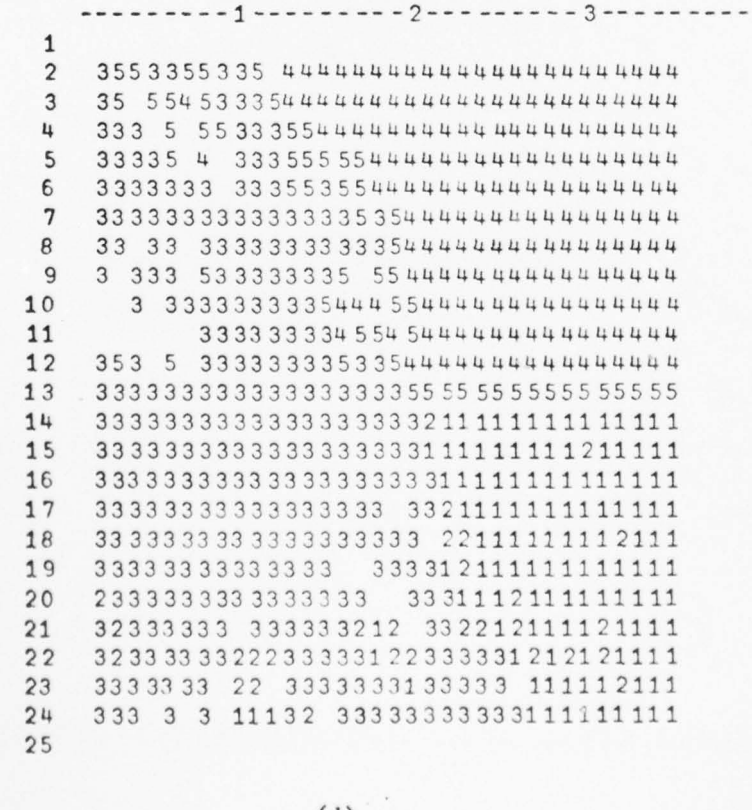
MAX VALUE OF HIST (66 , 77)= 78



MAX VALUE OF HIST (66, 77)= .70



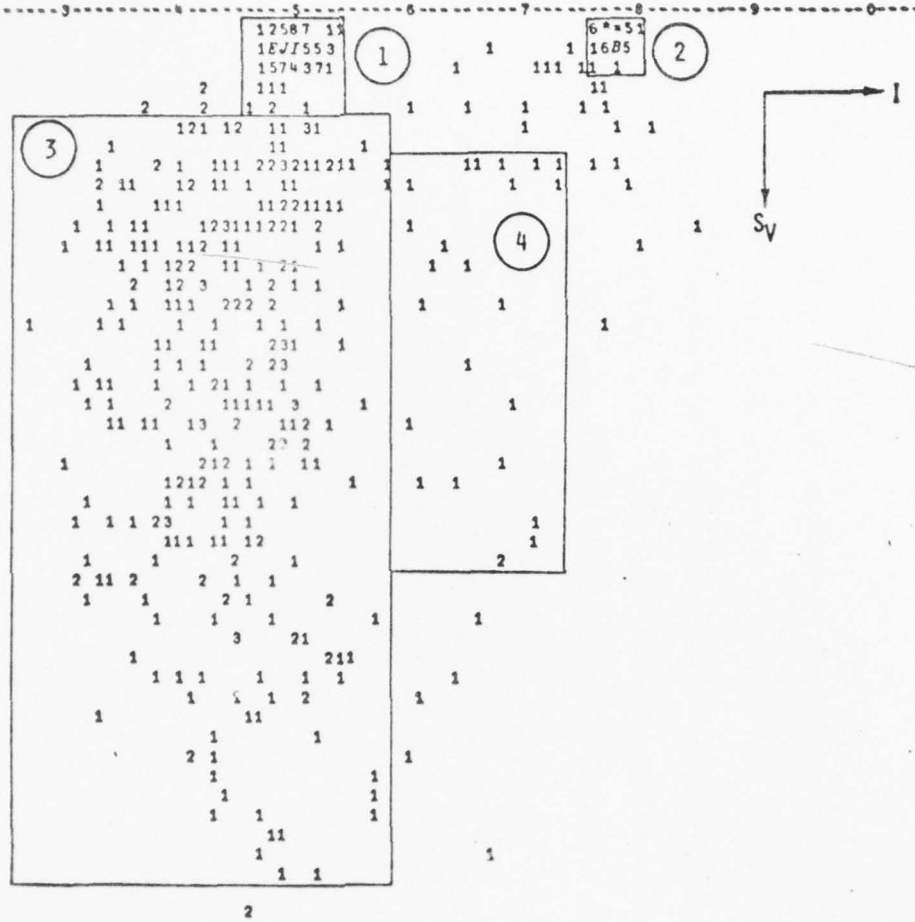
(c)



(d)

Figure 13

Figure 14: Use of two-dimensional histograms to guide region formation in subimage B - Example 2, Intensity vs. Saturation Variance. Refer to Figure 13 for more detailed explanation. (a) A two-dimensional histogram of intensity vs. saturation variance (S_V), where S_V of each point X is computed across a 3 × 3 window centered on X. (b) 2D histogram after non-extremum suppression is applied. (c) Projection of clusters in the suppressed histogram back onto image. (d) Projection of clusters in the unsuppressed histogram (a) back onto the image.


1
2
3
4
I
S_V

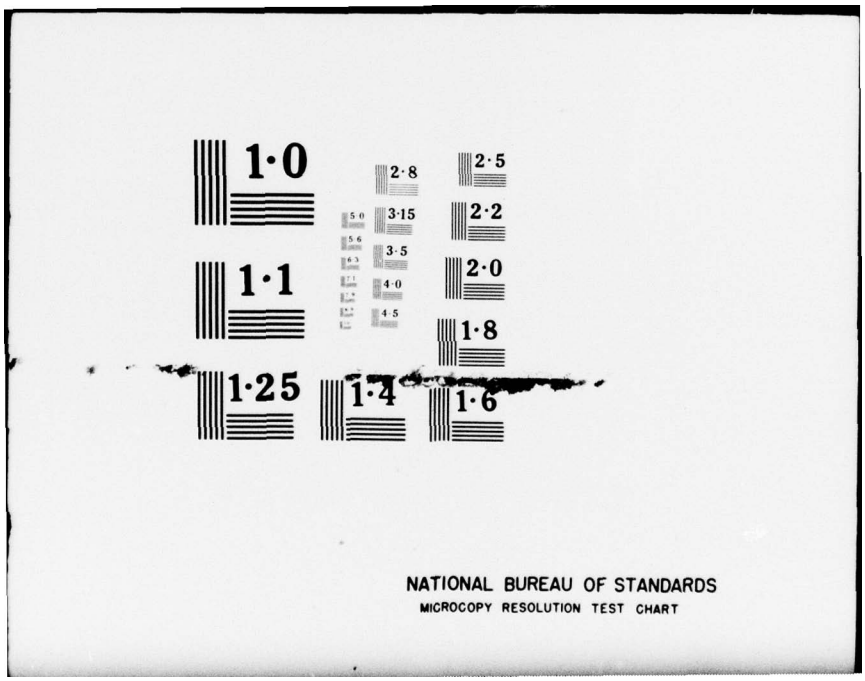
AD-A039 665 MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/G 9/2
COMPUTATIONAL TECHNIQUES IN THE VISUAL SEGMENTATION OF STATIC S--ETC(U)
MAR 77 E M RISEMAN, M A ARBIB N00014-75-C-0459

UNCLASSIFIED COINS-TR-77-4 NL

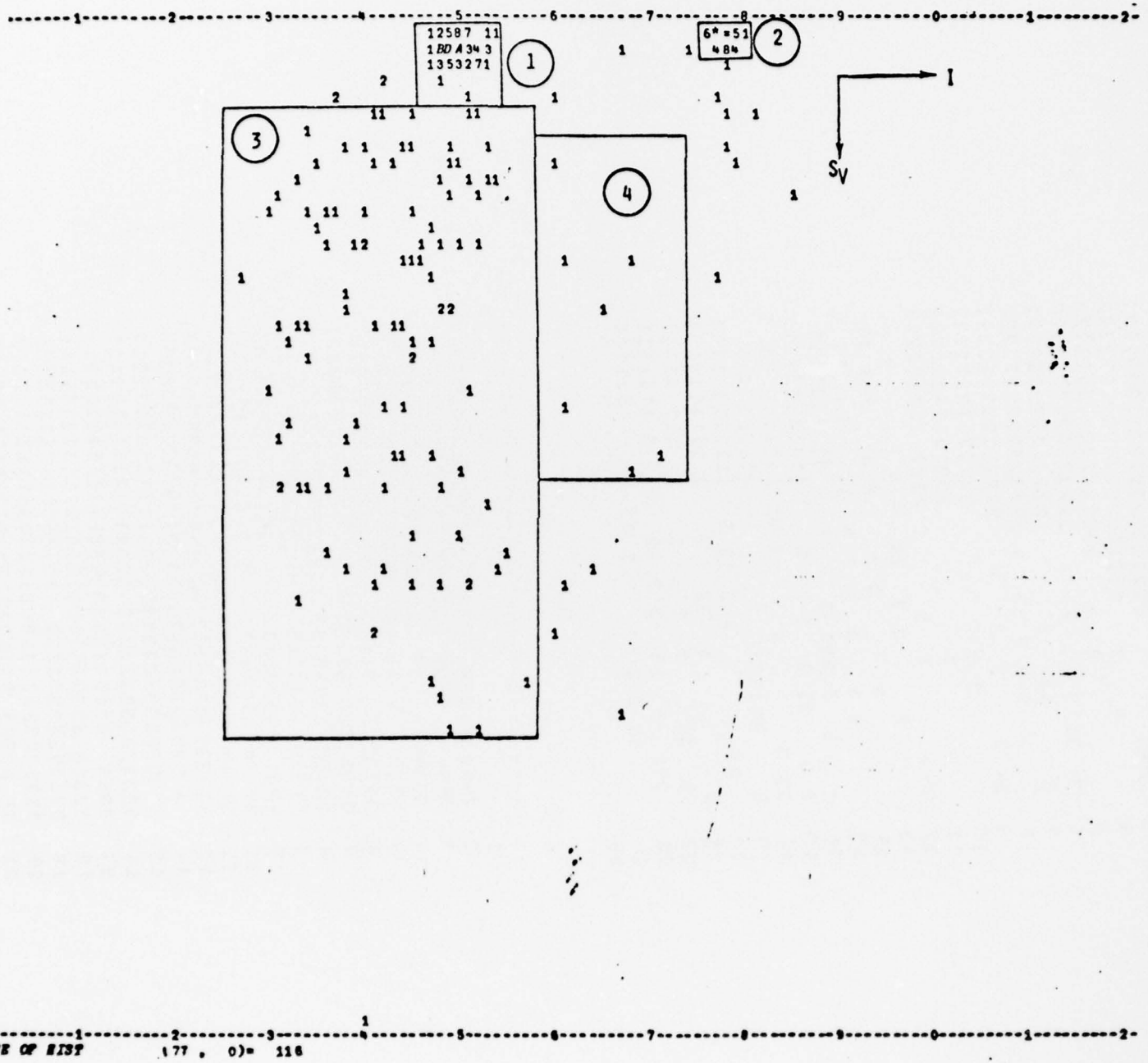
2 OF 2
AD
A039665

END

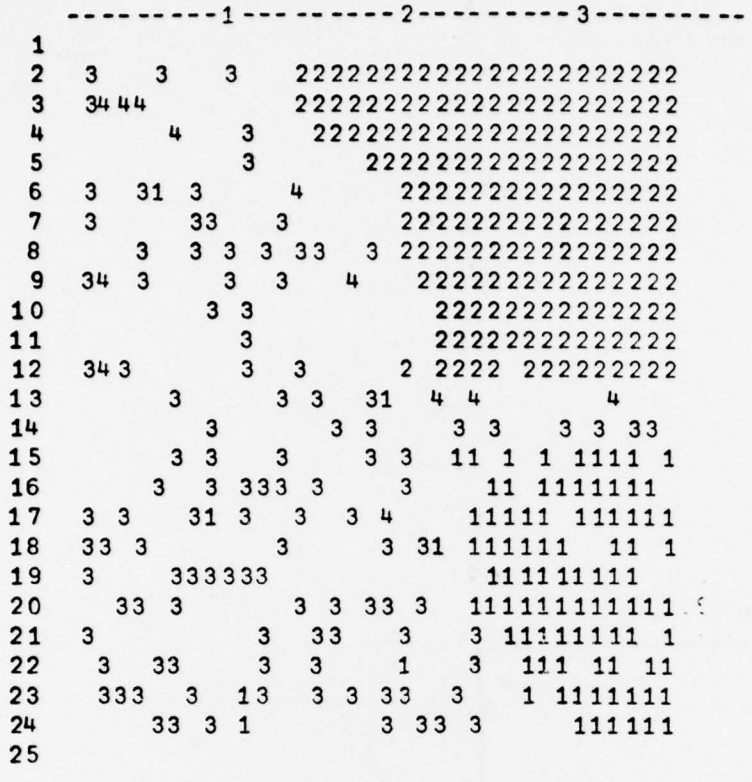
DATE
FILMED
6-77


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
6·3
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

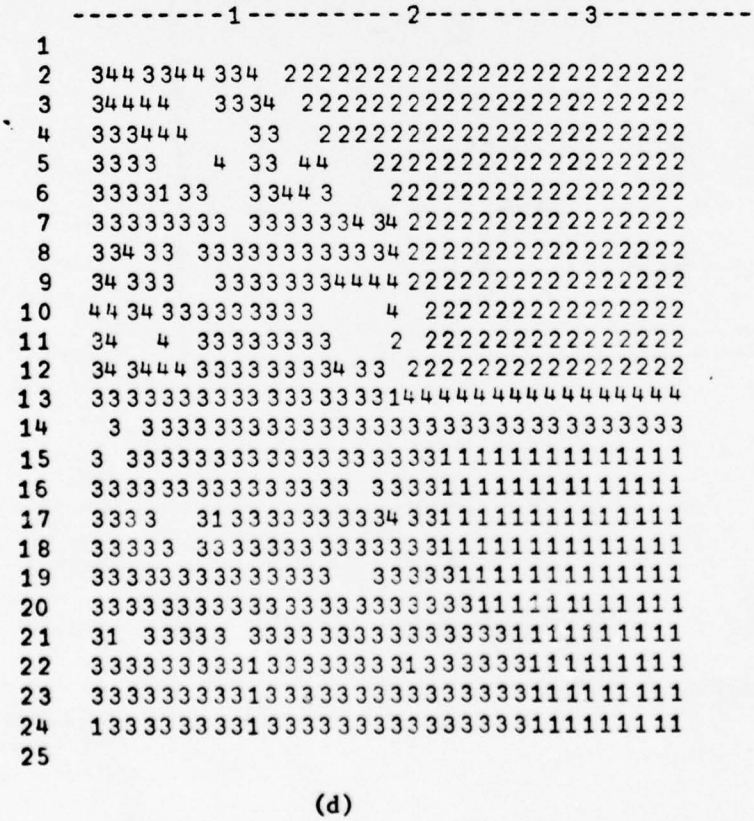
b)



MAI VALUE OF HIST (177 , 0)= 116


---------1---------2---------3---------
1
2 3 3 3 22222222222222222222
3 3444 22222222222222222222
4 4 3 22222222222222222222
5 3 222222222222222222
6 3 31 3 4 2222222222222222
7 3 33 3 2222222222222222
8 3 3 3 3 33 3 2222222222222222
9 34 3 3 3 4 222222222222222
10 3 3 22222222222222
11 3 22222222222222
12 343 3 3 2 2222 222222222
13 3 3 3 31 4 4 4
14 3 3 3 3 3 3 33
15 3 3 3 3 3 11 1 1 1111 1
16 3 3 333 3 3 11 1111111
17 3 3 31 3 3 3 4 11111 111111
18 33 3 3 3 31 111111 11 1
19 3 333333 11111111
20 33 3 3 3 33 3 111111111111
21 3 3 33 3 3 1 11111111 1
22 3 33 3 3 1 3 111 11 11
23 333 3 13 3 3 33 3 1 1111111
24 33 3 1 3 33 3 111111
25

(c)


---------1---------2---------3---------
1
2 3443344334 22222222222222222222222
3 34444 3334 22222222222222222222222
4 333444 33 22222222222222222222222
5 3333 4 33 44 22222222222222222222
6 33331 33 33443 222222222222222222
7 33333333 3333334 34 222222222222222
8 33433 333333333334 2222222222222222
9 34333 3333333444422222222222222222
10 4434333333333 4 222222222222222
11 34 4 33333333 2 222222222222222
12 343444333333334 33 2222222222222222
13 3333333333333333144444444444444444
14 3 3333333333333333333333333333333
15 3 33333333333333333111111111111111
16 333333333333333 33331111111111111111
17 3333 3133333333343311111111111111
18 33333 333333333333331111111111111111
19 333333333333 33333111111111111111
20 3333333333333333333331111111111111
21 31 33333 33333333333333311111111111
22 33333333313333333313333331111111111
23 33333333313333333333333311111111111
24 13333333313333333333333311111111111
25

(d)

Figure 14

```
    ---------1---------2---------3---------
 1
 2   3333333333 22222222222222222222
 3   33333  33333 22222222222222222222
 4   33333333  33  22222222222222222222
 5   3333    3  333333 22222222222222222
 6   33331 33   33333   222222222222222222
 7   33333333  3333333332222222222222222
 8   33333  333333333333222222222222222
 9   33333  3333333333332222222222222222
10   333333333333333    3 222222222222222
11   33   3  33333333    2 222222222222222
12   3333333333333333333 22222222222222222
13   33333333333333333313333333333333333
14    3  3333333333333333333333333333333
15   3  33333333333333333311111111111111
16   3333333333333333  33331111111111111
17   3333   31333333333333311111111111111
18   33333  33333333333333311111111111111
19   33333333333333333333311111111111111
20   333333333333333333333311111111111
21   31  33333333333333333333311111111111
22   33333333313333333313333331111111111
23   33333333313333333333333331111111111
24   13333333313333333333333331111111111
25
```

(e)

Figure 14

subset of the mutant values (the peaks) will be present. When forming histograms of the features of individual points (e.g., HSI), then the points which are on a gradient between min and max values will be suppressed; thus, in the case of tree-type texture, just the highlight and shadow points will contribute. Figures 13c and 14c portray the position of non-extremum points as blanks in the subimage. It should be noted that in the case of two features, a point is not suppressed if its value is a min or max in either of the two features. Figures 13b and 14b portray the 2D-histograms under non-extremum suppression and it clearly enhances the clusters of interest. Although the results are not entirely predictable in non-trivial areas of an image, generally cluster clarity should be improved.

There are many approaches to the extraction of clusters. A very simple technique utilized by Hanson et al. [1975] employed the cone structure to extract clusters. Clusters of activity are extracted by blurring (averaging) so that the activity is smoothed, and scaling so that low activity valleys between clusters disappear with only the peaks remaining. This corresponds to Ohlander's threshold setting process, and with some difficulty, it can be automated. All of these operations (besides the formation of the histogram itself) can easily be computed as local parallel operations in the processing cone. Afterwards, the boundaries of each cluster can be grown outward to capture the areas blurred and scaled away. Although there are many more sophisticated clustering algorithms (see Meisel [1972]) which might produce higher quality results, they may be exorbitant in computation or not be as intrinsically parallel in their nature.

It is clear that a system can use the information in the suppressed and unsuppressed histograms to define clusters in the unsuppressed histograms. We will assume that the clusters outlined in the histograms of Figures 13 and 14 have been extracted. Each cluster is symbolically labelled with a distinct numeric symbol. The next stage of processing involves a feedback loop to correlate these features with their spatial relationships in the original image. Points in the original image can be labelled according to the cluster to which they belong. Figures 13c-d and 14c-d represent the image labelled by the clusters in both the suppressed and unsuppressed histograms of Figures 13 and 14 respectively. In Figure 14d, the I-S_V histogram, it is clear that two of the clusters represent the roof (medium I, low S_V) and sky (high I, low S_V). Note the two unlabelled rows in between these areas. They are caused by false variations due to the window placement, and are incorrectly grouped into clusters 3 and 4 representing the tree-sky area as shown in Figure 14d. Since our window is 3 × 3 there will only be two rows which have these false values, but the problem becomes more critical as the window size increases. An 'intelligent' low-level system would understand the strengths and weaknesses of each segmentation procedure. Thus, this problem can be expected and, at least in this case, easily cleared up by a little more processing.

Now it is quite straightforward to grow regions across adjacent points with the same label and retain those regions which are relatively large.

But areas of heavy texture might form several clusters in the histograms, from highlights and shadows or from sky and foliage. Thus, the textured area which one wants to extract may have local areas of varying labels, with possibly many disconnected local areas of the same label. This effect is hinted at in Figures 14c-d and is seen more vividly in Figure 15 which is a result of a histogram of intensity mean vs. intensity variance on 3 × 3 windows.

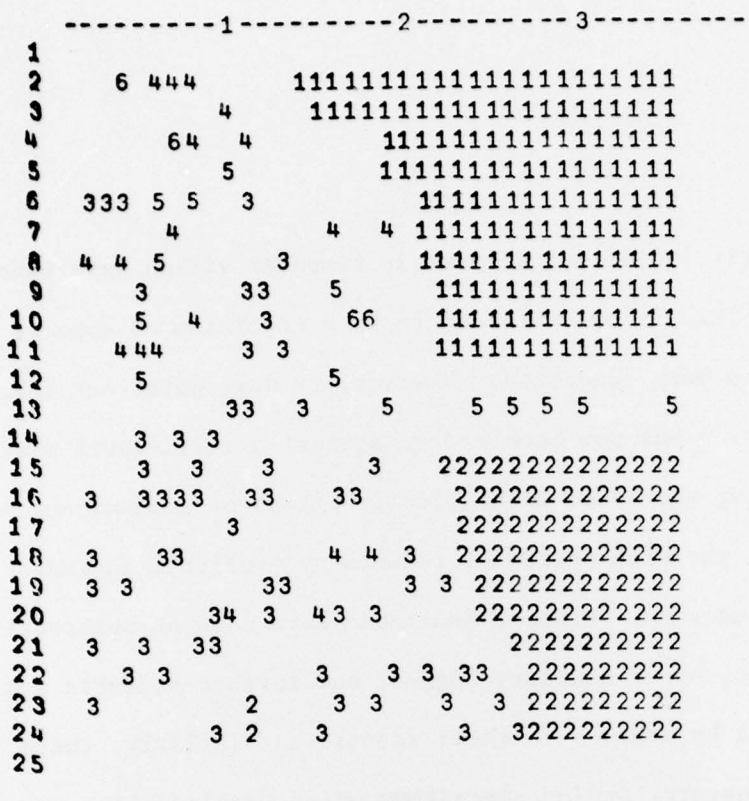
At this point we have atomic areas which represent either regions or possible texture elements depending on the resolution and the particular object. Certainly any large area so formed which consists of a single type of activity will be evaluated as an entity in itself, a region. If these areas are small, however, they might be considered microtexture elements.

The labelled points and areas provide the system access to statistical and structural spatial properties of the feature types. This analysis can be used to guide region growth across the symbolic labels. In an initial attempt to extract simple properties, the VISIONS group (Hanson, Riseman and Nagin [1975]) utilized an adjacency matrix as a measure of the degree to which atomic areas of different types are interspersed. Large numbers imply that two texture types are often adjacent to each other and signal the possibility that they form one or more regions with a macrotexture of these two (or more) types. By growing across the labels representing those two microtextures, a single macrotextured region is formed. We have depicted this case in Figure 14e by enlarging the cluster to cover both clusters 3 and 4 in Figure 14a. Thus, we have captured texture patterns in a single region and it is a simple matter to extract

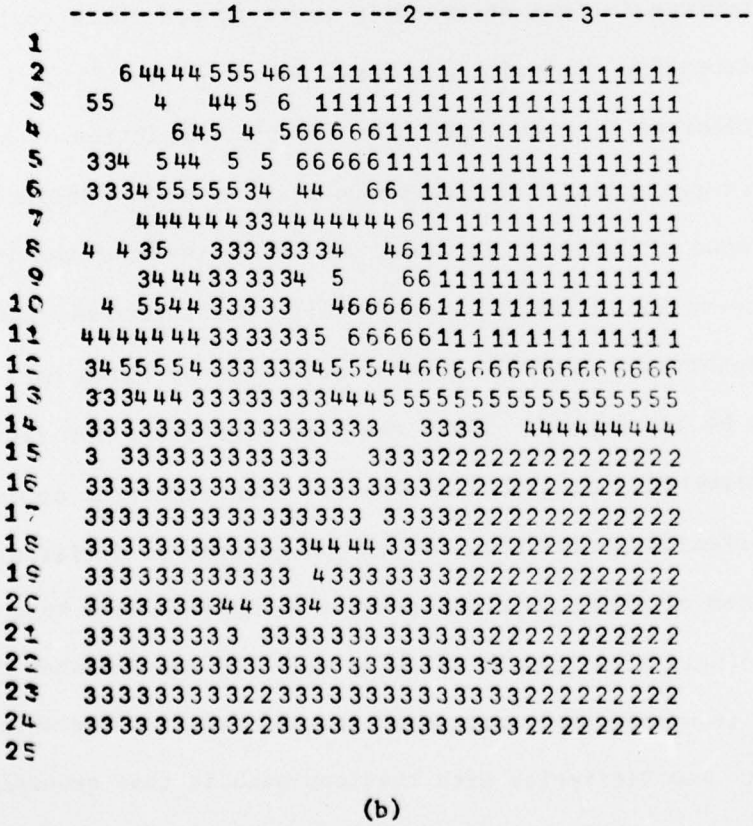
symbolic descriptors of the different microtextures used in the construction of the macrotexture. We leave the analysis of the redundancy of these different segmentation results for the reader's inspection.

A more sophisticated analysis of the statistical and structural characteristics is desired. One would like to note the difference between blue and green vertical stripes, and blue irregular shaped blobs amidst a green background. This implies the utility of a hierarchical feature-selector/texture-analyzer which is the subject of ongoing research.

Figure 15: Interspersion of cluster labels in textured areas. These projections of cluster labels are derived from a 2D histogram of intensity mean vs. intensity variance computed on 3 × 3 windows. (a) Clusters projected from suppressed histograms. (b) Clusters projected from unsuppressed histograms.



(a)



(b)

Figure 15

6. Conclusion

Until recently individual efforts in computer vision have been rather limited. This is not intended to be a criticism of some of the fine work that has been conducted. However, it does point out that the complexity of vision has not been tested against a multi-level systems approach of modular processes with effective means of communication between them. Humans use the high degree of redundancy available in images in order to understand them. Distant mountains have cues of perspective, a blue color shift, upper boundary shapes, and further semantic constraints which allow strong hypotheses to their identity. Similarly, there is a redundancy of features and/or algorithms which can lead to consistent segmentations in terms of regions and boundaries. Animals seem to exhibit multiple representations at an early level to aid their goal-oriented visual perception (Lettvin, Maturana, McCulloch and Pitts [1959]).

This paper exhibited several algorithms for the extraction of boundaries or regions. Since a representation of either boundaries or regions implicitly defines the other, we have a means of integrating their results in terms of the ideas of competition and cooperation (Arbib & Riseman [1976]). Relaxation and constraint-satisfaction algorithms may afford a general mechanism by which many kinds of information can be integrated. There are also algorithms for region growth on labels determined by global analysis. Each algorithm approaches the data from a different perspective and may be subject to different weaknesses. A system of these routines could allow performance beyond the capability of any single algorithm by allowing multiple and somewhat redundant representations to determine the portions of the segmentation for which there is high confidence. The difficulty with this approach is that several partially reliable segmentations could produce a maze of inconsistencies which are not easy

to resolve. There is also a significant overhead in additional computation on a serial machine. However, it seems quite reasonable to view these computations taking place in future on parallel hardware and in real time.

A low-level system for segmentation should have a front-end that allows any subset of a pool of features to be invoked for use by the different algorithms. This requires mechanisms to select relevant features entirely upon the basis of limited processing of the specific image under consideration. It is one example of the need for binding global feature analysis with local spatial analysis. One might use global histogram analysis to identify clusters of feature activity for ordering the potential importance of the features. Feedback from semantic processes after initial segmentation and interpretation can provide powerful guidance to the invocation of specific features.

We have examined the difficulties produced by overlapping feature distributions from different parts of a scene. This confusion in the analysis might be reduced by performing a coarse segmentation by edge analysis on a blurred image so that major areas can be delimited and processed independently. In our example, then, areas of tree texture and the straight lines within the house might be analyzed by distinct algorithms and/or features which are most suitable to each. At this point in our development of computer vision systems, such a level of generality and flexibility is extremely difficult to achieve. However, it appears to be a natural direction for integrating the broad range of efforts underway.

Some workers believe that implementation of general computer vision systems will not be within our grasp for some time. This paper has shown that such a conclusion is not without justification. However, the effectiveness of an integrated system approach has yet to be evaluated. While research on general computer vision systems continue, they should provide spin-offs in more constrained applications. There has already been a focus upon ERTS satellite imagery, bio-medical applications, and limited industrial assembly line work with promising results. It is important to pursue the goals of both general vision and limited goal-oriented vision systems during the coming years.

Acknowledgements

We wish to thank the many students who aided in the development of the interactive programs used to produce most of the analyses in this paper; this group principally included Richard Clackenmeyer, Vincent Valvo and Jefferey Conklin. We also with to thank our secretary Ms. Janet Turnbull for her help and patience in producing the various drafts of this lengthy paper.

References

M. A. Arbib and E. M. Riseman [1976], "Computational Techniques in Visual Systems. Part I: The Overall Design", COINS Technical Report 76-10, Department of Computer & Information Science, University of Massachusetts, Amherst.

M. L. Baird and M. D. Kelly [1974], "A Paradigm for Semantic Picture Recognition", Pattern Recognition, June, 6, 61-74.

M. L. Baird [1976], "An Application of Computer Vision to Automated IC Chip Manufacture", Proc. of 3rd Int. Joint Conf. on Pattern Recognition, November, 3-7.

R. Bajcsy [1973], "Computer Description of Textured Surfaces", Proc. of 3rd IJCAI, August, 572-579.

R. Bajcsy and L. Lieberman [1974], "Computer Description of Real Outdoor Scenes", Proc. 2nd Int. Joint Conf. on Pattern Recognition, August, 174-179.

H. G. Barrow and J. M. Tenenbaum [1976], "MSYS: A System for Reasoning About Scenes", Technical Report 121, AI Center, Stanford Research Institute, April.

J. Beck [1975], "Surface Color Perception", Scientific American, August, 233, 82.

P. Bouma [1971], Physical Aspects of Color, MacMillan, London.

C. R. Brice and C. L. Fenema [1970], "Scene Analysis Using Regions", Artificial Intelligence, 1, 205-226.

B. Bullock [1974], "The Performance of Edge Operators on Images with Texture", Technical Report, Huges Research Labs, Malibu.

M. B. Clowes [1971], "On Seeing Things", Artificial Intelligence, Spring, 2, 79-116.

T. N. Cornsweet [1970], Visual Perception, Academic Press.

L. S. Davis [1975], "A Survey of Edge Detection Techniques", Computer Graphics and Image Processing, 4, 248-270.

L. S. Davis and A. Rosenfeld [1976], "Applications of Relaxation Labeling 2: Spring-Loaded Template Matching", Proc. of 3rd Int. Joint Conf. on Pattern Recognition, November, 591-597.

R. O. Duda and P. E. Hart [1973], Pattern Classification and Scene Analysis, John Wiley and Sons.

R. W. Ehrich and J. P. Foith [1976], "Representation of Random Waveforms by Relational Trees", IEEE Trans. on Computers, C-25, No. 7, 725-736.

R. W. Ehrich [1976], "Detection of Global Linear Features in Remote Sensing Data", Proc. Joint Workshop on Pattern Recognition and Artificial Intelligence, June.

R. M. Evans [1948], An Introduction to Color, Wiley.

M. A. Fischler and R. A. Elschlager [1973], "The Representation and Matching of Pictorial Structures", IEEE Trans. on Computers, C-22, 67-92.

J. R. Fram and E. S. Deutsch [1975], "On the Quantitiative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance", IEEE Trans. on Computers, June, C-24, 616-628.

E. C. Freuder [1976], "Affinity: A Relative Approach to Region Finding", Computer Graphics and Image Processing, 5, 254-264.

A. Guzman [1968], "Decomposition of a Visual Scene into Three-Dimensional Bodies", Proc. of Fall Joint Computer Conference, 33, 291-304.

A. Hanson and E. Riseman [1974], "Preprocessing Cones: A Computational Structure for Scene Analysis", COINS Technical Report 74C-7, Department of Computer & Information Science, University of Massachusetts, Amherst.

A. Hanson and E. Riseman [1975], "The Design of a Semantically Directed Vision Processor (Revised and Updated)", COINS Technical Report 75C-1, Department of Computer & Information Science, University of Massachusetts, Amherst.

A. Hanson, E. Riseman and P. Nagin [1975], "Region Growing in Textured Outdoor Scenes", Proc. 3rd Milwaukee Symposium on Automatic Computation and Control, 407-417.

R. Haralick, K. Shanmugan and I. Dinstein [1973], "Textured Features for Image Classification", IEEE Trans. on Systems, Man, and Cybernetics, November, SMC-3, 610-621.

K. C. Haynes, Jr., A. N. Shah and A. Rosenfeld [1974], "Texture Coarseness: Further Experiments", IEEE Trans. on Systems, Man, and Cybernetics, September, SMC-4, 467-472.

B. K. P. Horn [1973], "The Binford-Horn Linefinder", MIT AI Memo 285, July, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge.

B. K. P. Horn [1975], "Image Intensity Understanding", MIT AI Memo 335, August, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge.

M. Hueckel [1973], "A Local Visual Operator Which Recognizes Edges and Lines", Journal of the ACM, October, 20, 634-647.

D. A. Huffman [1971], "Impossible Objects as Nonsense Sentences", Machine Intelligence 6 (B. Metzler and D. Michie, eds.), American Elsevier, 295-323.

B. Julesz, H. L. Frisch, E. N. Gilbert and L. A. Shepp [1973], "Inability of Humans to Discriminate Between Visual Textures that Agree in Second-Order Statistics--Revisited", Perception, 2, 391-405.

B. Julesz [1975], "Experiments in the Visual Perception of Texture", Scientific American, April, 232, 34-43.

R. A. Kirsch [1971], "Computer Determination of the Constituent Structure of Biological Images", Computers and Biomedical Research, 4, 315-328.

A. Klinger and C. R. Dyer [1974], "Experiments on Picture Representation Using Regular Decomposition", Technical Report UCLA-ENG 7497, University of California, Los Angeles.

J. Y. Lettvin, H. Maturana, W. S. McCulloch and W. H. Pitts [1959], "What the Frog's Eye Tells the Frog's Brain", Proc. IRE, 47, 1940-1951.

A. K. Mackworth [1973], "Interpreting Pictures of Polyhedral Scenes", Proc. 3rd Int. Joint Conf. on Artificial Intelligence, 556-561.

D. Marr [1975], "Early Processing of Visual Information", AI Memo 340, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge.

J. W. McKee and J. K. Aggarwal [1975], "Finding the Edges of the Surfaces of Three-Dimensional Curved Objects by Computer", Pattern Recognition, June, 7, 25-52.

W. S. Meisel [1972], Computer Oriented Approaches to Pattern Recognition, Academic Press.

R. Nevatia [1975], "Object Boundary Determination in a Textured Environment", Proceedings of 1975 National Conference of ACM.

F. O'Gorman and M. B. Clowes [1973], "Finding Picture Edges Through Collinearity of Feature Points", Proceedings of 3rd International Joint Conference on Artificial Intelligence, 543-555.

R. Ohlander [1975], "Analysis of Natural Scenes", Ph.D. Thesis, Carnegie-Mellon University, April.

W. A. Perkins [1976], "Multilevel Vision Recognition System", Proc. of 3rd Int. Joint Conf. on Pattern Recognition, November, 739-744.

J. Prager, A. Hanson and E. Riseman [1976], forthcoming Technical Report, Department of Computer & Information Science, University of Mass., Amherst.

L. G. Roberts [1965], "Machine Perception of Three-Dimensional Solids", Optical and Electro-Optical Information Processing (J. T. Tippett et al., eds.) MIT Press.

A. Rosenfeld and M. Thurston [1971], "Edge and Curve Detection for Visual Scene Analysis", IEEE Trans. on Computers, May, C-20, 562-569.

A. Rosenfeld and E. B. Troy [1970], "Visual Texture Analysis", IEEE Conference on Feature Extraction and Selection in Pattern Recognition, 115-124.

A. Rosenfeld and A. C. Kak [1976], Digital Picture Processing, Academic Press.

A. Rosenfeld, M. Thurston, and Y. H. Lee [1972], "Edge and Curve Detection: Further Experiments", IEEE Trans. Computers, July, C-21, 677-714.

A. Rosenfeld, R. A. Hummel, S. W. Zucker [1976], "Scene Labelling by Relaxation Operations", IEEE Trans. on Systems, Man and Cybernetics, SMC-6, 420-433.

B. Schachter, L. S. Davis, and A. Rosenfeld [1975], "Scene Segmentation by Cluster Detection in Color Space", Tech. Report 424, Computer Science Center, University of Maryland.

S. Shapiro [1975], "Transformations for the Computer Detection of Curves in Noisy Pictures", Computer Graphics and Image Processing, 4, 328-338.

Y. Shirai [1973], "A Context Sensitive Line Finder for Recognition of Polyhedra", Artificial Intelligence, 4, no. 2, 95-119.

K. R. Sloan, Jr. and R. Bajcsy [1975], "A Computational Structure for Color Perception", Tech. Report, Department of Computer and Information Science, University of Pennsylvania, April.

S. L. Tanimoto and T. Pavlidis [1975], "A Hierarchical Data Structure for Picture Processing", Computer Graphics and Image Processing, June.

J. M. Tenenbaum and H. G. Barrow [1976], "Experiments in Interpretation-Guided Segmentation", Joint Conference on Pattern Recognition and Artificial Intelligence, Hyannis, Mass., June.

J. M. Tenenbaum and H. G. Barrow [1976], "IGS: A Paradigm for Integrating Image Segmentation and Interpretation", Proc. of 3rd Int. Joint Conf. on Pattern Recognition, November, 504-513.

J. M. Tenenbaum and S. Weyl [1975], "A Region-Analysis Subsystem for Interactive Scene Analysis", SRI Technical Note 104, Artificial Intelligence Center, Stanford Research Institute.

J. M. Tenenbaum, T. D. Garvey, S. Weyl and H. C. Wolf [1974], "An Interactive Facility for Scene Analysis Research", SRI Technical Note 87, Artificial Intelligence Center, Stanford Research Institute.

F. Tomita, M. Yachida and S. Tsuji [1973], "Detection of Homogeneous Regions by Structural Analysis", Proc. of 3rd Int. Joint Conf. on Artificial Intelligence, 564-571.

E. B. Troy, E. S. Deutsch, and A. Rosenfeld [1973], "Gray-Level Manipulation Experiments for Texture Analysis", IEEE Trans. on Systems, Man and Cybernetics, 91-98.

G. J. Vanderbrug [1975], "Experiments in Iterative Enhancement of Linear Features", Technical Report TR-425, Computer Science Center, University of Maryland, November.

D. L. Waltz [1975], "Understanding Line Drawings of Scenes with Shadows", The Psychology of Computer Vision (P. H. Winston, ed.), McGraw-Hill.

H. Wechsler and J. Sklansky [1975], "Automatic Detection of Rib Contours in Chest Radiographs", Proc. of 4th Int. Joint Conf. on Artificial Intelligence, USSR, 688-694.

C. Weiman [1975], "Scene Analysis: A Survey", Courant Computer Science Technical Report #9, Computer Science Department, New York University, December.

Y. Yakimovsky and J. A. Feldman [1973], "A Semantics-Based Decision Theory Region Analyzer", Proc. of 3rd Int. Joint Conf. on Artificial Intelligence, 580-588.

Y. Yakimovsky [1976], "Boundary and Object Detection in Real World Images", Journal of the ACM, 23, #4, 599-618.

S. W. Zucker [1975], "Region Growing: Childhood and Adolescence", Technical Report 370, University of Maryland.

S. W. Zucker, R. A. Hummel, A. Rosenfeld [1975], "Applications of Relaxation Labelling, 1: Line and Curve Enhancement", Technical Report 419, Computer Science Center, University of Maryland.

S. W. Zucker, A. Rosenfeld, L. Davis [1975], "General Purpose Models: Expectations about the Unexpected," Proc. of 4th IJCAI, USSR, 716-721.

S. W. Zucker [1976], "Relaxation Labelling and the Reduction of Local Ambiguities", Proc. of 3rd Int. Joint Conf. on Pattern Recognition, November, 852-862.

S. W. Zucker, E. V. Krishnamurty and R. L. Hoar [1976], "Relaxation Processes for Scene Labelling: Convergence, Speed and Stability", Technical Report 477, Computer Science Center, University of Maryland, August.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER COINS-TR-77-4	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTATIONAL TECHNIQUES IN THE VISUAL SEGMENTATION OF STATIC SCENES		5. TYPE OF REPORT & PERIOD COVERED INTERIM 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Edward M. Riseman Michael A. Arbib		8. CONTRACT OR GRANT NUMBER(s) ONR N00014-75-C-0459, NSF-DCR75-16098 NIH 5R01 NS09755-06 COM
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer and Information Science University of Massachusetts Amherst, Massachusetts 01003		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE 3/77 13. NUMBER OF PAGES 104
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)

Distribution of this document is unlimited.

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

18. SUPPLEMENTARY NOTES

407 701

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

computer vision
segmentation
boundary formation
region growing
texture
feature clusters
relaxation procedures

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

A wide range of segmentation techniques continues to evolve in the literature on scene analysis. Many of these approaches have been constrained to limited applications or goals. This survey analyzes the complexities encountered in applying these techniques to color images of natural scenes involving complex textured objects. It also explores new ways of using the techniques to overcome some of the problems which are described. An outline of considerations in the development of a

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

general image segmentation system which can provide input to a semantic interpretation process is distributed throughout the paper.

In particular, the problems of feature selection and extraction in images with textural variations are discussed. The approaches to segmentation are divided into two broad categories, boundary formation and region formation. The tools for extraction of boundaries involve spatial differentiation, non-maxima suppression, relaxation processes, and grouping of local edges into segments. Approaches to region formation include region growing under local spatial guidance, histograms for analysis of global feature activity, and finally an integration of the strengths of each by a spatial analysis of feature activity. A brief discussion of attempts by others to integrate the segmentation and interpretation phases is also provided. The discussion is supported by a variety of experimental results.