# STUDIES IN IMAGE SEGMENTATION ALGORITHMS BASED ON HISTOGRAM CLUSTERING AND RELAXATION

Paul A. Nagin

COINS Technical Report 79-15

(September 1979)

This work was supported by The Office of Naval Research under Grant Number N00014-75-C-0459.

# STUDIES IN IMAGE SEGMENTATION ALGORITHMS BASED ON HISTOGRAM CLUSTERING AND RELAXATION

A Dissertation Presented

By

PAUL ALEXANDER NAGIN

Submitted to the Graduate School of the
University of Massachusetts in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 1979

Computer and Information Science

# STUDIES IN IMAGE SEGMENTATION ALGORITHMS BASED ON HISTOGRAM CLUSTERING AND RELAXATION

A Dissertation Presented

By

PAUL ALEXANDER NAGIN

Approved as to style and content by:

Edward M. Riseman

Edward M. Riseman, Chairperson of Committee

Allen R. Hanson

Allen R. Hanson, Member

D.N. Spinelli, Member

RMGraham

Robert M. Graham, Department Head
Computer and Information Science

To Dr. Howard Landy and his family

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Edward Riseman and Dr. Allen Hanson without whose guidance and support this thesis would not have been possible. I would also like to thank Dr. Nico Spinelli for his comments and criticisms.

I would like to thank Tom Williams, Ralf Kohler, Bryant York, and Steve Epstein for the many hours they spent with me discussing the technical issues raised in this work.

Finally, I would like to thank Janet Turnbull and Linda Strzegowski for the excellent job they did in preparing this manuscript.

This work was supported by

The Office of Naval Research

under Grant Number

N00014-75-C-0459

# ABSTRACT

## Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation

September, 1979

Paul Alexander Nagin

B.S., Antioch University

M.S., University of Massachusetts

Ph.D., University of Massachusetts

Directed by: Professor Edward M. Riseman

The research in this thesis has focussed upon the algorithms and structures that are sufficient to generate an accurate description of the information contained in a relatively complex class of digitized images. This aspect of machine vision is often referred to as "low-level" vision or segmentation, and usually includes those processes which function close to the sensory data. The bulk of this thesis devotes itself to the exploration of some of the problems typically encountered in segmentation. In addition, a new and robust algorithm is presented that avoids most of these problems.

The analysis is carried out through the use of a series of computer-generated test images with known characteristics. Segmentation algorithms of varying degrees of complexity are applied to each image and their performance is carefully evaluated. It will be shown that even the most sophisticated algorithms that are currently in use often perform poorly when confronted with certain apparently simple images. In particular, it is shown that techniques which rely on

histogram clustering often generate gross segmentation errors due to overlap in the distributions of the individual objects in a scene. Moreover, the relaxation processes used to correct these errors are themselves prone to errors, but of a different kind. Here, we show that the globally computed compatibility functions are inadequate to preserve image structure, even in some surprisingly simple images.

Both techniques, clustering and relaxation, fail because they are based on information which is too global to be effective in complex scenes. Clustering fails because most algorithms do not take into account the spatial feature information contained in the image. Relaxation-type algorithms take the spatial content into account by utilizing global information applied to local neighborhoods. However, global compatibility functions very often fail to resolve local image structure. This implies that improvements in performance might be obtained by localizing the algorithm to sub-images of the original image. In fact, a dramatic improvement in performance is obtained when this is done. Each sub-image is defined to be small enough so that the distributions of distinct visual elements are revealed as distinct histogram clusters. Moreover, the compatibility coefficients are measured over a sufficiently small area so that their characterization of the local image structure is not diluted by global effects. After segmenting each sub-image, a merging algorithm is applied so that regions that have been artificially split at sub-image boundaries can be sewn together to form the final segmentation.

## TABLE OF CONTENTS

# LIST OF TABLES

LIST OF FIGURES

# CHAPTER I

## INTRODUCTION

The research in this thesis has focussed upon the algorithms and structures that are sufficient to generate an accurate description of the information contained in a relatively complex class of digitized images. This aspect of machine vision is often referred to as "low-level vision" and usually includes those processes which function close to the sensory data. The general goal of our low-level system is the transformation of a large spatial array of pixels (i.e. picture elements) into a more compact description of the image in terms of visually distinct syntactic units and their characteristics. Such a transformation is referred to as a segmentation. By a variety of means, the visual information must be aggregated, labelled with symbolic names and attributes, and then interfaced to higher level knowledge structures.

The complexity of the data which must be examined by the segmentation processes has had significant effect upon the design of those processes. With relatively complex, unconstrained images, such as full color outdoor scenes, any approach to segmentation will be prone to error. Highly textured objects such as trees, shadows and highlights on both regular and irregular surfaces, varied and uncontrollable lighting conditions, all contribute to the difficulty of analysis. Few objects or surfaces can be expected to exhibit truly uniform visual features. Therefore, methods for dealing with this variability must be incorporated not only into the processes themselves, but also into the manner by which the results of the processes are interpreted and used. The system discussed in this thesis incorporates the flexibility of representation and the generality of processes which are necessary to accomplish this task.

## I.1 Evaluation of Segmentation

In spite of the very active and diverse research on image segmentation systems, performance evaluation of these systems remains an open question. In order to evaluate the quality of segmentation, one must specify the goals of the processing. However, these goals vary widely in their form and in their complexity. In one case the goal might be to determine the presence of a dark area on a textured gray background (as in biomedical image applications), while in another it could be to provide information to a system which is to construct a three-dimensional model of the physical surfaces that are present in the imaged environment (as in some image understanding systems [HAN78]).

Let us assume for the moment that the goal is to partition an arbitrarily complex image, say an outdoor image, into objects and surfaces. Although this goal is simply stated, the problem of evaluating a segmentation which is purported to fulfill these conditions is still extremely difficult. Subjective evaluation is clearly not sufficient to provide the quantitative measures necessary to compare either a given segmentation to a goal or to rank two segmentations relative to the goal. Some form of "ground-truth" data would be required in order to define global measures: the question remains as to where this data is to be obtained. In the natural scenes to be analyzed here, information from the physical scene has undergone several stages of degradation, including the photographic process, the digitization process, and a spatial averaging process to reduce the amount of data to managable levels (in this case, 512x512 to 256x256 pixels). The effect of these processes is to introduce noise, blur edges, and to create hybrid feature values -- mixed pixels -- which are not easily classifiable. Moreover, the image contains inherent visual complexities such as irregular texturing, highlights, shadows, object occlusion, and irregular changes in gradients due to changes in surface reflectance.

The presence of these anomalies implies that accurate ground-truth segmentations are difficult or impossible to obtain. Hand-drawn segmentations are inevitably prone to errors and tend to reflect implicit biases and explicit goals of the human perceiver. In many instances the boundaries would be conjectured, based on prior expectations in the form of knowledge of object shape, shadow effects, perspective cues, and occlusion cues. In short, it is generally accepted that a truly accurate segmentation of an image requires the application -- at some point -- of "high-level" knowledge, i.e., knowledge beyond directly measurable features of the data.

The problem of when and what high-level knowledge should be used will not be addressed here. Because the sensory data is sometimes inherently ambiguous, and because the procedures necessary to disambiguate the image may not be definable in a low-level system, it is difficult to decide whether an algorithm has done a good job of characterizing difficult data or whether the algorithm has misinterpreted that data.

We have adopted the goal of image segmentation to be the decomposition of an image into visually distinct regions, that is, regions which have relatively uniform visual properties of intensity, color, texture, etc. One of the algorithms whose results will be presented demands, for each region produced, unimodality in the features used in the segmentation. However, we will show that this does not ensure the proper partitioning of an image, due to problems such as overlap of the feature distributions of adjacent target regions.

In order to avoid many of the problems cited, we have chosen to bypass the objective evaluation of the segmentation of natural scenes -- although we will apply the algorithms and subjectively evaluate the results. On the other hand, the application of the algorithms to machine-generated test data is more likely to lead to insights into the capabilities and limitations of the algorithm. Here, "ground truth" is available, and consequently, the results are amenable to evaluation as well. The algorithms developed in this thesis will be applied to both machine-generated test images and natural scenes.

## I.2 The Processing Cone

There is a serious problem of data overload incurred by the necessity of repeatedly processing images on the order of 256x256 pixels to 1024x1024 pixels. Consequently, a commitment was made to the development of parallel algorithms within the VISIONS* processing cone structure [HAN74,UHR72,TAN75], wherever possible.

The function of the processing cone is the transformation and reduction of the massive amount of image data via local parallel processing, while at the same time providing a hierarchical structure in which information at coarser levels of data resolution can direct more detailed processing of data at finer levels of resolution. This use of "planning" [KEL71, NAG77, PRI77] can significantly reduce the actual amount of computation which must be performed during the analysis of an image.

## I.3 Segmentations Based on Regions and on Edges/Boundaries

The segmentation processes used in the VISIONS image understanding system are based on complementary techniques. The primary technique discussed here groups individual pixels on the basis of their relative similarity with their neighbors. The resulting collections of labelled pixels exhibit uniformity over the characteristics with which they are

*VISIONS stands for: Visual Integration by Semantic Interpretation of Natural Scenes.

aggregated; such collections are referred to as regions. Boundaries may be produced by differentiating with respect to region labels.

The second process makes use of the local differences which exist between pixels in order to form local edges; these edges are then grouped into boundary segments [HAN78]. Regions may be formed by labelling those pixels which are entirely enclosed by a collection of boundary segments.

There is no a-priori reason to assume that the boundaries (or regions) produced by these disparate processes will coincide, either in terms of their physical placement within the image or in terms of the characteristics of the pixels grouped by them. The merging of the region and boundary outputs is currently under investigation [KOH79].

## I.4 Some Basic Terminology and Paradigms

Let us briefly define a few of the more frequently used terms and transformations that are used in image processing. First, the data itself must be defined. For our purposes, an image consists of a discrete sampling of sensory data into a two-dimensional spatial array of cells called picture elements or pixels. In addition, each pixel is quantized to a discrete range of gray levels. The transformation from the real world scene to its digital representation is referred to as digitization and is accomplished via a scanning device and an analog-to-digital converter. Typically, a digitized image contains on the order of 512x512 pixels or 256x256 pixels, with each pixel quantized to 6 or 8 bits (128 or 256 gray levels).

The digitization of a scene may be restricted to the black and white intensity information in the scene. However, color information can be obtained through the use of light filtration during scanning. In the latter case, the scene is usually scanned three times, one each through red, green, and blue (RGB) filters. Notice that a typical image contains a staggering amount of information:

512x512 pixels x 3 colors x 8 bits per pixel $\simeq$

6 million bits per image

A *feature* is a property that is useful in discriminating "elements" of an image, such as objects, surfaces, and regions. Any transformation of the raw data may be thought of as measuring some feature, although some transformations are more useful than others. For instance, in the domain of outdoor scenes, color (hue) is a useful feature for discriminating sky from grass, while black and white intensity might not distinguish those two objects.

A feature need not be computed solely at the level of individual pixels. For instance, edge operators typically involve convolution of an edge mask with the image; thus, a neighborhood around each pixel is employed. Moreover, it is sometimes useful to compute features across predefined regions in the image, or indeed across the entire image itself (e.g. the average brightness level of the scene).

Notice that *preprocessing* may be thought of as a special kind of feature extraction that "prepares" the image for further feature analysis. For example, when an image is digitized, the scanner sometimes measures the intensity value across a boundary between objects. In such a case, the gray level that is recorded is a hybrid value, since it represents the average intensity of two distinct "areas". Algorithms have been designed which detect and correct these "mixed" pixels. The application of such an algorithm is a form of preprocessing, since it is applied to the raw data and logically precedes any other image transformation. It is also feature extraction since its application tends to enhance boundaries.

Once the image has been digitized and a set of features has been computed, the next step in image analysis is to aggregate the data into units that have similar features. For instance, the analysis may use edge contrast as a feature to be measured at each pixel, but the aggregation of edges into lines may be the ultimate goal. Furthermore, line formation could be controlled by local geometric factors such as continuity and linearity. The latter are meta-features, properties of the array of feature values of edges, and only indirectly the properties of the array of pixels. Similarly, a region analysis system, using hue as a feature, might have a region of similar hues as the ultimate goal. When the aggregation process is completed and all pixels have been assigned to a labelled unit -- a region or a line -- the resulting partition is referred to as a segmentation.

Let us look a little more closely into the process of forming regions. Region analysis may be controlled by local factors such as pixel adjacency and local feature similarity, but these are often insufficient in the formation of regions that are meaningful in a larger context. Thus, region analysis often incorporates global measures of feature similarity to group pixels.

One technique of measuring global feature similarity involves the use of histograms, or frequency distributions of gray levels. For instance, the hue-histogram (i.e. the feature space of hue values) of an image that contains green trees and blue sky may be bi-modal, since it is the union of two distributions with strongly separated means. Now, if one assumes in general that an image will have some feature histogram that has as many distinct modes as distinct objects, then one may attempt to extract those objects indirectly by isolating the modes in the distribution, which then identifies the corresponding regions in the image. Thus, region analysis can be transformed into a statistical classification problem and make use of discriminant functions or cluster analysis. Isolating histogram modes is analogous to the problem of finding an optimal decision surface (hyperspace) in feature space.

Let us briefly mention that, in practice, region analysis via such mechanisms of pattern classification is highly prone to error. The reason for this is that the feature distributions of the objects to be classified tend to overlap to varying degrees. Indeed, some clusters may be completely obscured by others, so that there may be fewer clusters than objects. Thus, the classification processes will necessarily be incomplete, with the effect that some pixels will be erroneously grouped.

Recovery from classification errors has been a major focus of this thesis. It will be shown that a class of transformations known as *relaxation labelling processes* (RLPs) can be helpful in error recovery. Briefly stated, RLPs use neighborhood information around each pixel and image-specific statistics to correct pixel classifications. Thus local information can be used to correct errors introduced by global classification.

## I.5 Summary of Remaining Chapters

The remainder of this thesis is organized as follows. Chapter 2 reviews the major work done by other researchers in the field of region analysis. The discussion covers three kinds of approaches: locally-based algorithms using pixel-by-pixel merging, globally-based algorithms using cluster discrimination in feature space, and our type of hybrid system where correction of globally-induced errors via local spatial analysis can be performed.

Chapter 3 is a detailed exploration of histogram-based region analysis. Three problems induced by cluster overlap in histograms will be demonstrated via a series of simple test images. Possible solutions to the problems are presented.

Chapter 4 presents a more complex segmentation algorithm that is shown to improve the histogram-based technique by adding a relaxation labelling process (RLP). The RLP uses three kinds of information to obtain an improved pixel classification or labelling:

(1) probabilistic cluster affiliation is introduced,

(2) neighborhood information is used to condition the probability of a pixel belonging to a class, and

(3) image-wide statistics, called compability coefficients, are used to preserve fine detail while allowing "noise-classifications" to be suppressed.

The augmented algorithm is demonstrated using artificial and natural data. In the latter case, attention is given to the use of opponent-color feature spaces to improve the segmentations.

Chapter 5 presents a series of test images that refute some of the positive results of the previous chapter. It is shown that the relaxation technique cannot be relied upon for recovery from errors that are due to cluster overlap. Moreover, the compatibility statistics are shown to be inadequate to preserve certain image structures.

Chapter 6 proposes a solution to the above problems via "intermediate localization" whereby the image is artificially broken into small sub-images which are independently analyzed and then later merged. The use of sub-images reveals clusters that may be hidden in the global image-wide histograms. It also allows the compatibility coefficients to better represent local image structure without being diluted by global effects. This formulation of the segmentation algorithm yields dramatically improved results when applied to the test images and the natural scene.

Finally, Chapter 7 summarizes the research, outlines the contributions, and proposes improvements to the current work.

# CHAPTER II

# BACKGROUND

The computer analysis of two-dimensional images, known as segmentation, image processing, low-level image analysis, and region formation, has been under investigation for over 10 years. During this time, numerous general-purpose and applications-oriented systems have evolved. The goal of this chapter is to define some of the basic techniques and review some of the more general-purpose systems that have been developed.

For the following discussion, let us divide the region segmentation techniques into three broad categories as in [RIS77, KAN78]:

(1) *Locally-based* (bottom-up): systems that use local spatial criteria to build regions directly from pixels or other primitive elements.

(2) *Globally-based* (top-down): systems that use global spectral criteria to split regions into primitive elements.

(3) *Hybrid* (top-down with local refinement): systems that use global criteria to obtain an approximate segmentation, and then apply local criteria to obtain a refined result.

## II.1 Local Region Analysis

Local region analysis involves any or all of the following steps:

formation of atomic regions (primitive elements), syntactic merging of regions, and semantic merging of regions. The simplest definition of an atomic region is that it consists of pixels that are (1) spatially contiguous, and (2) the difference in feature value between any adjacent pair of pixels is less than some theshold. A group of pixels that satisfy these conditions is given a unique region label.

The threshold for pixel merging may be (1) a fixed constant that is independent of any information in the image, (2) a fixed constant that is dependent on some global, image-specific measurement, e.g. the standard deviation from the mean gray level, or (3) a variable whose value depends on information in a local area around a pixel.

Brice and Fennema [BRI70] developed a strategy which first formed atomic regions according to the most conservative criterion possible, namely, that adjacent pixels may be merged into the same regions if their gray levels are identical. Next, region merging criteria are applied based on the "weakness" of region boundaries. Specifically, region pairs are merged if a sufficiently large portion of their common boundary has a sufficiently low gray level difference.

Barrow and Popplestone [BAR71] used a slight variant of the technique of Brice and Fenema to segment regions that were later analyzed via primitive template matching. Atomic regions are formed by merging adjacent pairs of pixels that are within a small range of brightness values (unlike Brice and Fenema who required a zero range). Boundary and region differences are then used to merge large regions. A feature vector is constructed for each region and these are matched against models of the known objects. Success of this system is strongly dependent on the small number and simplicity of the objects that are used.

Kelly [KEL70] developed a specialist system for distinguishing pictures of people. The program uses "planning" to recognize faces in a hierarchical, goal-oriented fashion. Thus, obvious features such as the head are searched for first, then the eyes, mouth, etc. In addition, the image is reduced by averaging 8x8 non-overlapping windows across the image. The coarsened image allows the program to do searching and backup without a significant time penalty. The smoothed picture also eliminates digitization noise that might otherwise interfere with the recognition process.

Feldman and Yakimovsky [FEL73] utilized semantic information in a decision-theoretic approach to scene segmentation. The information includes properties of the boundaries between regions (e.g., how likely is the adjacency of two regions) and properties of the regions themselves (color, shape, etc.). After initial clustering of picture points to form regions, a decision-tree analysis is used to further join and then identify regions according to a maximum likelihood analysis based on these properties. For more complex environments, we feel that the a-priori conditional probability of a feature given a region cannot be reliably estimated (usually the number of samples is very small) and changes drastically with respect to a different context and over time. Thus, it is becoming apparent that the inclusion of more complex semantic information is necessary; furthermore, the nature of this information must be such that it can be utilized in a highly flexible manner.

Tenenbaum and Barrow [TEN76] demonstrated that the interactive human semantic labelling of regions can be used to block most erroneous merges made by nonsemantic rules. They interactively supply labels of identities to initial conservatively formed atomic regions whose size is greater than some threshold t. Then, an attempted merger of two regions with differing labels can be blocked, while the merger of an unlabelled region with a labeled 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 t, the human again supplies the proper label. For a simple office scene and outdoor scene, the final results are quite reasonable when is set so that about 20 regions are labeled during this process.

This approach led Tenebaum and Barrow to employ a generalization of Waltz's [WAL75] 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 predefined 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 Riseman and Arbib [RIS77].

Freuder [FRE76] provided an interesting 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 difficult questions concerning texture that remain to be solved if this is to be a viable approach.

Chen and Pavlidis [CHE78] used a split-and-merge algorithm and a co-occurrence matrix to segment based on textural differences of regions. Their system uses a layered, parallel cone structure [UHR73,HAN74] to store and manipulate images. The bottom level of the cone (or quadratic picture tree as they refer to it) corresponds to individual pixels, while the highest level (root node) represents the entire picture. Each node has four children corresponding to its four subsets. Regions identified at any level of the tree can be split into subregions or merged into super-regions depending on some criteria.

In the system designed by Chen and Pavlidis, the criteria for split/merge is a gray level co-occurrence matrix [HAR78, HAN75, RIS77]. The (i,j) entry in the matrix represents the probability that pixels with gray levels i and j co-occur at some distance d apart and some orientation theta with respect to each other. Thus, off-diagonal elements may represent local microtexture. The system merges the four subsets of a node if their matrices are not too different. Otherwise, the subsets are left alone and the node is split.

## II.2 Global Regions Analysis

This class of approaches 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 (or any other measurable featue), one would expect the intensity histogram to form major peaks (or clusters) about their respective means.

One of the earliest uses of histogram thresholding was by Prewitt and Mendelson [PRE66]. Their technique consists of finding valleys (antimodes) in histograms of white blood cells. Once the valleys are found, each pixel in the image can be labelled according to which one of the peaks that its intensity value belongs to. Then, simple region growing can be applied so that adjacent pixels with the same mode-label can be given the same region label. Tsuji and Tomita [TSU73] extended the mode selection idea to multiple features which are computed locally for the purpose of analyzing textures drawn on block surfaces.

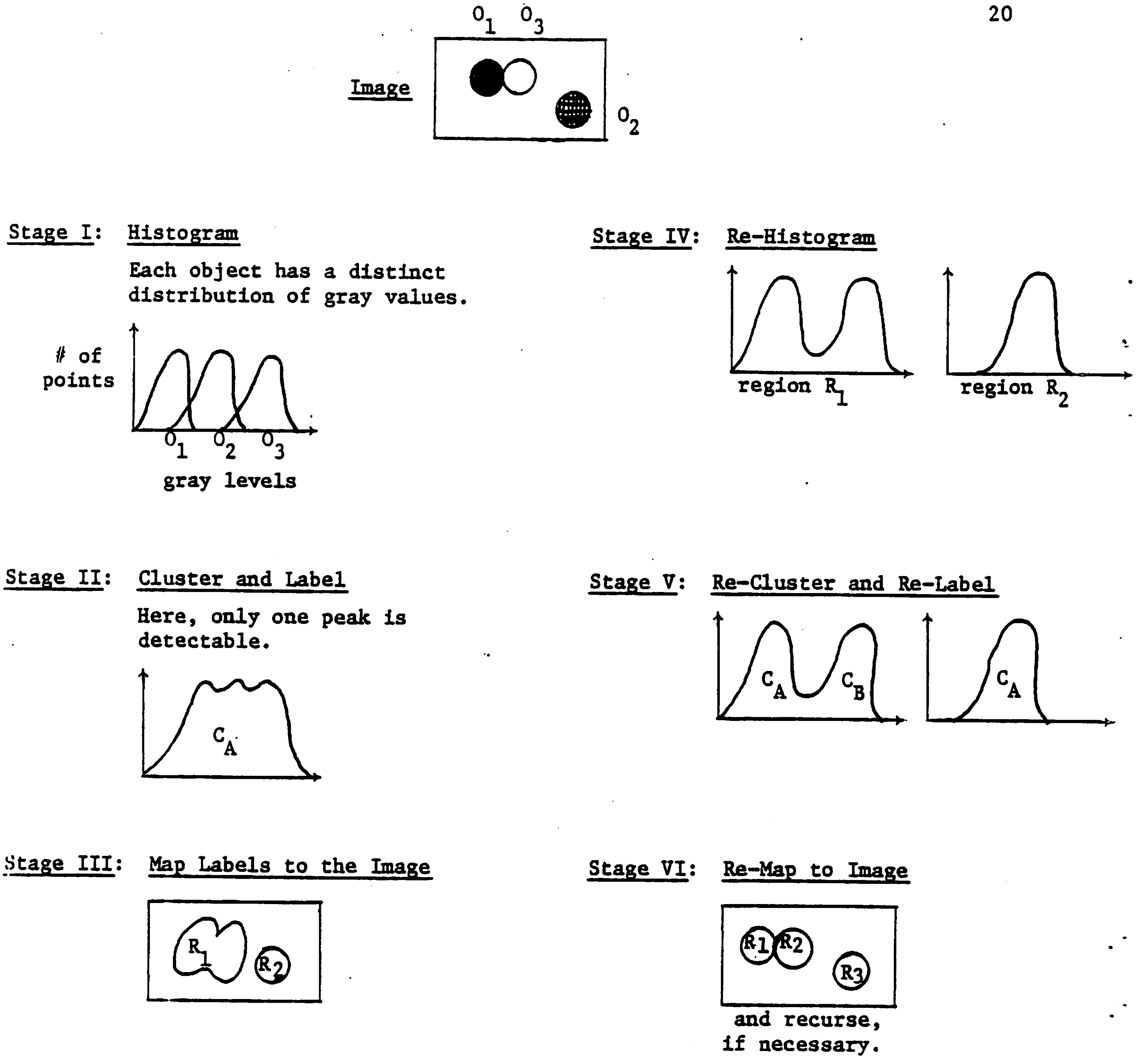
Ohlander [OHL75] developed a technique of recursively partitioning an image by setting thresholds at valleys of 1D histograms of various features. The first partition forms around the clearest peak in any histogram; then, the associated points in the image are flagged and adjacent points with the same label are merged into a region by growing on the symbolic labels. These regions are smoothed by blurring, and each of these distinct regions forms 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 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. Often the peaks and cluster widths of typical histograms are not so clear. A more serious problem, though, is that different objects can partially overlap distributions of other objects in one or all of the features. This can cause peaks and valleys to appear and disappear--and shift--if the particular combination of objects is varied, despite the possibility that all of the objects appear visually distinct to the human observer.

In general, one can hope that the sequential determination of the largest regions can be used to continually subtract away the data which obscures the presence of less noticeable peaks in the global feature histogram. Howver, the quality of this algorithm seems to be subject to an arbitrary condition, namely the particular mix of regions being examined. (See Figure II.1 and II.2 on recursive analysis.) This problem would probably be reduced if the image were broken into smaller aeas; this can be thought of as a foveal window where the system initially focuses 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 multidimensional histograms were employed (although then the detection of peaks and clusters is less straightforward).

But there is still a more significant drawback that must be overcome; that is the lack of information on the spatial relationships of the features being examined. On the basis of a global histogram analysis, one cannot determine the difference between a red area bordering a yellow area and red polka dots within a yellow area--they can produce identical histograms and the difference in structure is not seen.

Price [PRI77] improved upon the segmentation system of Ohlander and added new work on change detection. His techniques dramatically



Stage I: Histogram

Each object has a distinct distribution of gray values.

Stage II: Cluster and Label

Here, only one peak is detectable.

Stage III: Map Labels to the Image

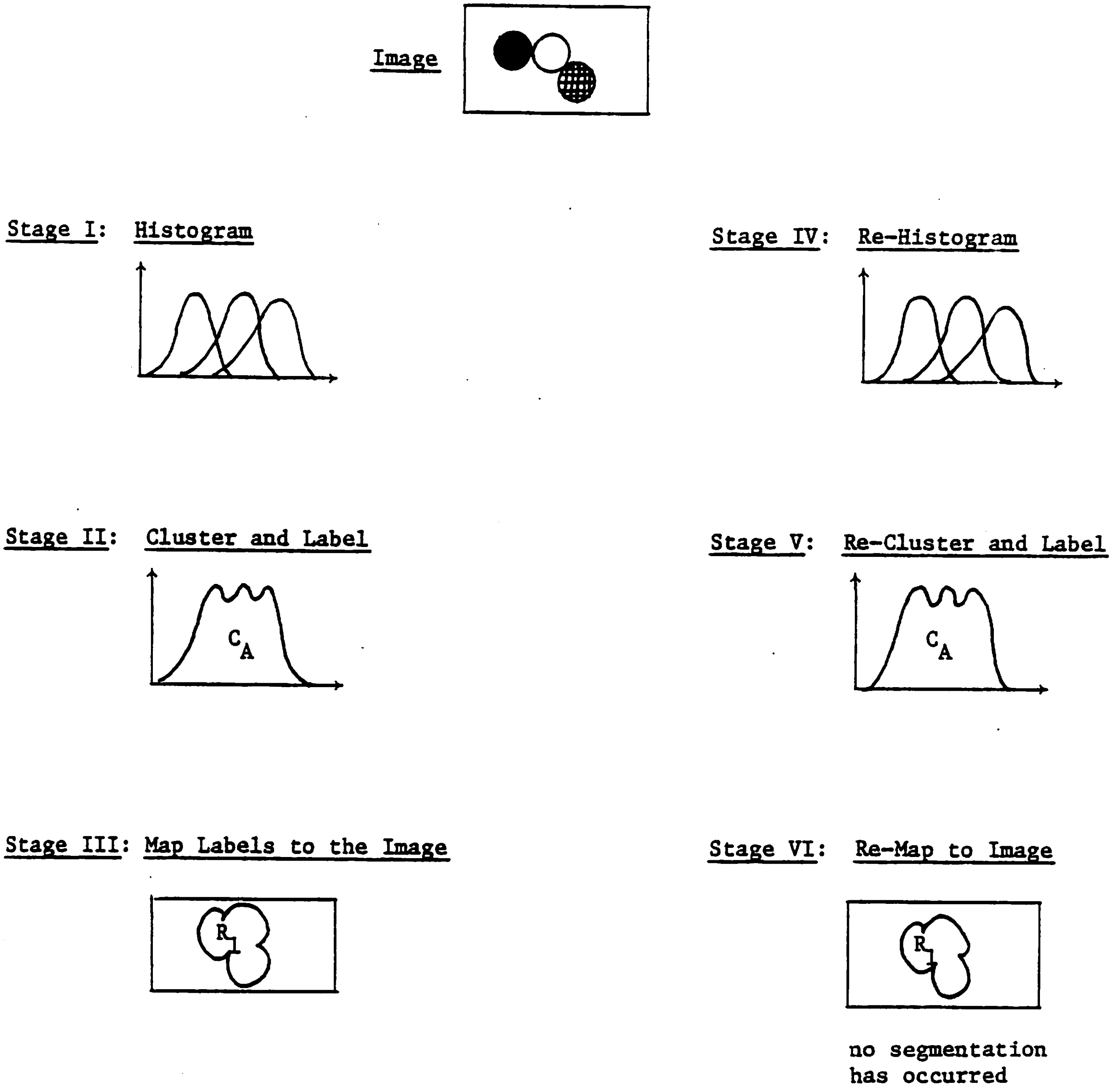
Stage IV: Re-Histogram

Stage V: Re-Cluster and Re-Label

Stage VI: Re-Map to Image

and recurse, if necessary.

Figure II.1 Recursive Segmentation - Successful: A Recursive analysis may be necessary when the distributions of objects overlap and individual peaks are observed.



Stage I: Histogram

Stage II: Cluster and Label

Stage III: Map Labels to the Image

Stage IV: Re-Histogram

Stage V: Re-Cluster and Label

Stage VI: Re-Map to Image

no segmentation has occurred

Figure II.2 Recursive Segmentation - Failure: The analysis may not succeed if objects with overlapping distributions are also adjacent in the image.

reduced segmentaion time mainly through the use of a "reduced" image. Thus, the input data which may have been scanned to a resolution of 512x512 pixels is transformed into a coarsened "plan" [KEL71, NAG77] that consists of perhaps 128x128 pixels. For instance, a plan may be obtained by computing the "average" pixel value in 2x2 or 4x4 non-overlapping windows computed across the image. Although this technique offers an obvious speedup for further segmentation processes, it has the disadvantage of blurring boundaries and creating hybrid values in textured areas.

Coleman [COL77] defined the problem of region segmentation as unsupervised clustering and feature selection in n-dimensional feature space. The best number of clusters in the feature space (i.e., the number of region types in the image to be partitioned) is defined as one that gives the maximum of the ratio of between-cluster scatter and within-cluster scatter. The usefulness of a feature is defined by the average of Bhatacharyya distances between pairs of clusters. The features which have the least power in discriminating clusters are discarded and the segmentation which yields the "best number of regions" is sought.

Several techniques developed in relation to the use of histograms should be briefly mentioned. When the sizes of the target regions are very different and their features are somewhat similar, the peak corrsponding to a bigger region tends to hide the peak corresponding to a smaller one. The resultant histogram may not demonstrate a clear bimodality. One useful technique to overcome this difficulty is to compute the histogram only for those pixels near the boundary of the regions [WES74]. First the spatial derivative (say, Laplacian) of the image intensity is computed. Then only those pixels which have a high derivative value are histogrammed. The differential histogram by Watanabe [WAT74] is another useful method to calculate an appropriate threshold level. A survey of threshold selection techniques can be found in [WES78,KOH79].

## II.3 Hybrid Systems

Local approaches to region analysis have the drawback that they tend to generate regions that are either "under-grown" or "over-merged." Textured areas tend to be broken into many fragments, while areas that are for the most part separable, but have a few weak (low contrast) boundary points, can leak together to form large regions.

The global approaches often obtain good initial segmentations but lack spatial sensitivity to generate highly accurate results. Thus, it seems natural to try to combine the two approaches so that local analysis can correct obvious mistakes of the global analysis.

The simplest remedy is to add a post-processing phase to the histogram-based segmentation. For example, if most of the neighbors of pixel P have been labeled as C, then P itself is reclassified as label C. This type of post-processing, called <u>plurality relaxation</u> (see Chapter 4), has been used in remote sensing applications. For example, the fact that a "wheat" pixel will not appear in the middle of the "corn" field seems to justify this technique. A slightly modified method involves the use of a "conservative" threshold [NAG77]. The classification of the pixels having feature values near the threshold (or boundary of the discriminant surface) is delayed, and those pixels are classified according to the labels of the neighbors.

A generalization of this approach is the use of relaxation labelling processes [ROS76, ZUC78, Chapter 4]. First, instead of assigning a single label to each pixel, the probability p that P belongs to class C is estimated via the distribution of image feature values. Then, these probabilities are adjusted using some relaxation formula:

$$p_i(n) \leftarrow F\ (p_i\ (n-1), \{q_i(n-1)\ |\ Q \text{ is a neighbor of } P\})$$

which means that $p_i$ is revised iteratively using the previous values of its own and of the neighboring pixels. Eklundh [EKL78] and Nagin [NAG78] have applied the relaxation technique to natural scenes and were able to show an improvement over the initial classification of the image pixels.

# CHAPTER III

## SEGMENTATION USING HISTOGRAMS

Feature histograms of an image have been well-established in pattern recognition and scene analysis [PRE66, OHL75, ROS76] as a fruitful representation to aid in region detection. A histogram of two visually discriminable objects should reveal two peaks (or clusters) that represent the distribution of gray levels for the objects. To generate a segmentation, it is simply a matter of isolating the clusters and then dividing the image pixels into classes according to the cluster with which they are associated.

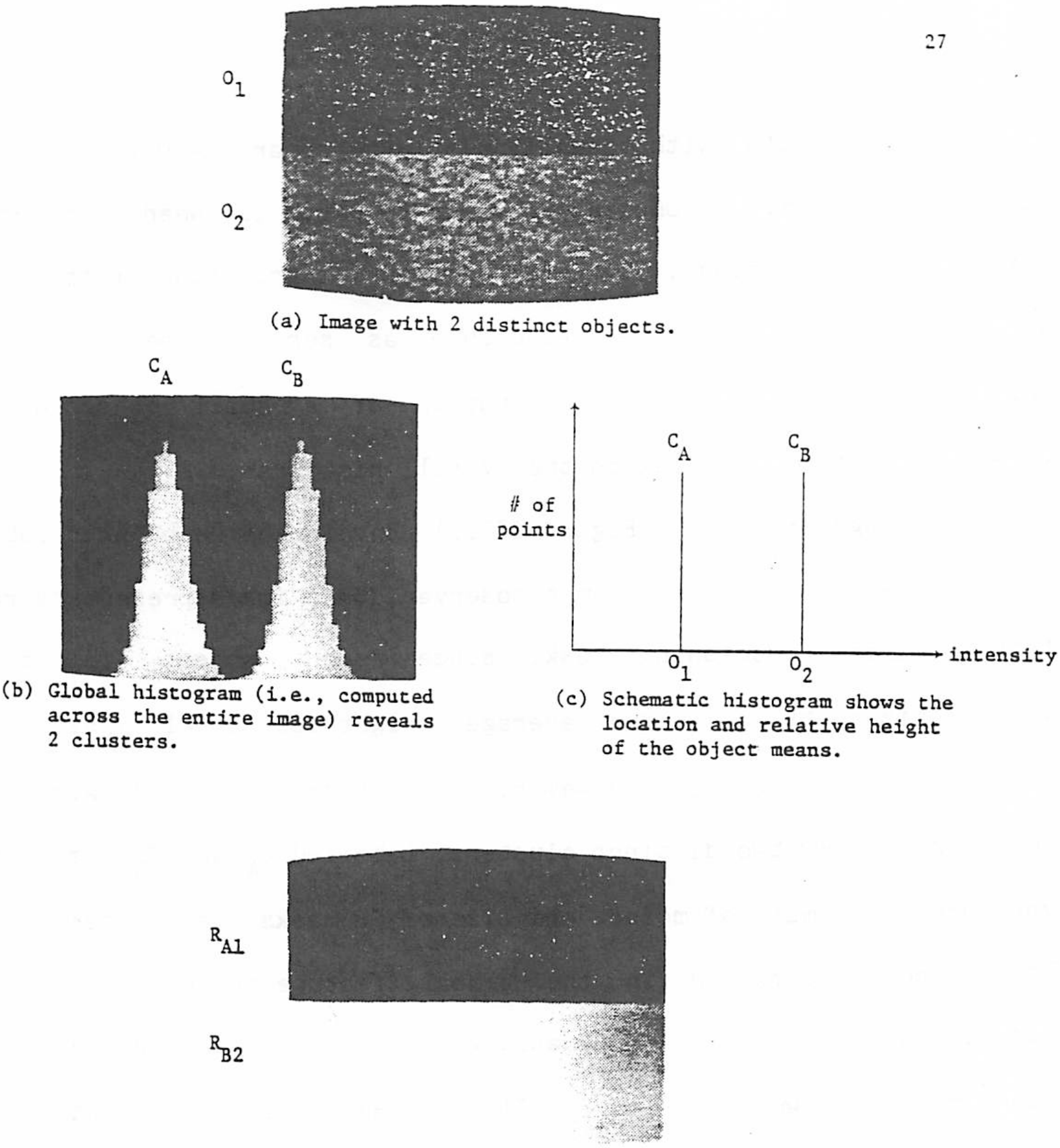
This technique -- feature clustering and pixel labelling -- will be referred to as a _global_ segmentation algorithm because the histogram is counted across the entire image without regard to the location of any particular pixel. This class of algorithms has been explored and developed by many researchers and, considering its simplicity, has met with reasonable success even when applied to complex natural images. There is, however, a serious fault with the technique, namely the difficulty of _completely_ isolating the distributions of objects via the histogram representation. This will be referred to as the "cluster overlap" problem. In the remainder of this section we discuss three kinds of segmentation errors caused by cluster overlap: _overmerging_, _fragmentation_, and _thin-object fragmentation_. The discussions are illustrated by a series of computer-generated test images ("cases") which demonstrate particular problems.

## III.1 Notation

The use of histograms for scene labelling is a two-step process that maps objects in scenes into histogram clusters and then maps the cluster labels to segmented regions (Figure III.0). As will be shown, there may not be a 1-1 correspondence between objects and regions -- a region may contain or be contained within an object. Thus, for example, region $R_2$ may not refer to object $O_2$. In the text that follows it would not be illuminating to refer to regions by number; rather they will be referred to by the label of the cluster that the pixels within the region map into. Moreover, regions may be further specified by a number that indicates the object space from which they have been generated. In Figure III.0, $R_{B2}$ is the label of the region generated by cluster $C_B$ and which is contained within the space occupied by object $O_2$. Unless otherwise stated, objects in the test images are assumed to have normal distributions with random spatial placement of the features values in the distribution. Sometimes the distribution of feature values will be correlated with the spatial position within the region representing the object.

## III.2 Overmerging

The first kind of segmentation error that will be discussed results whenever the histogram of an image does not reveal as many peaks as there are distinguishable objects. In this event, pixels



(a) Image with 2 distinct objects.

(b) Global histogram (i.e., computed across the entire image) reveals 2 clusters.

(c) Schematic histogram shows the location and relative height of the object means.

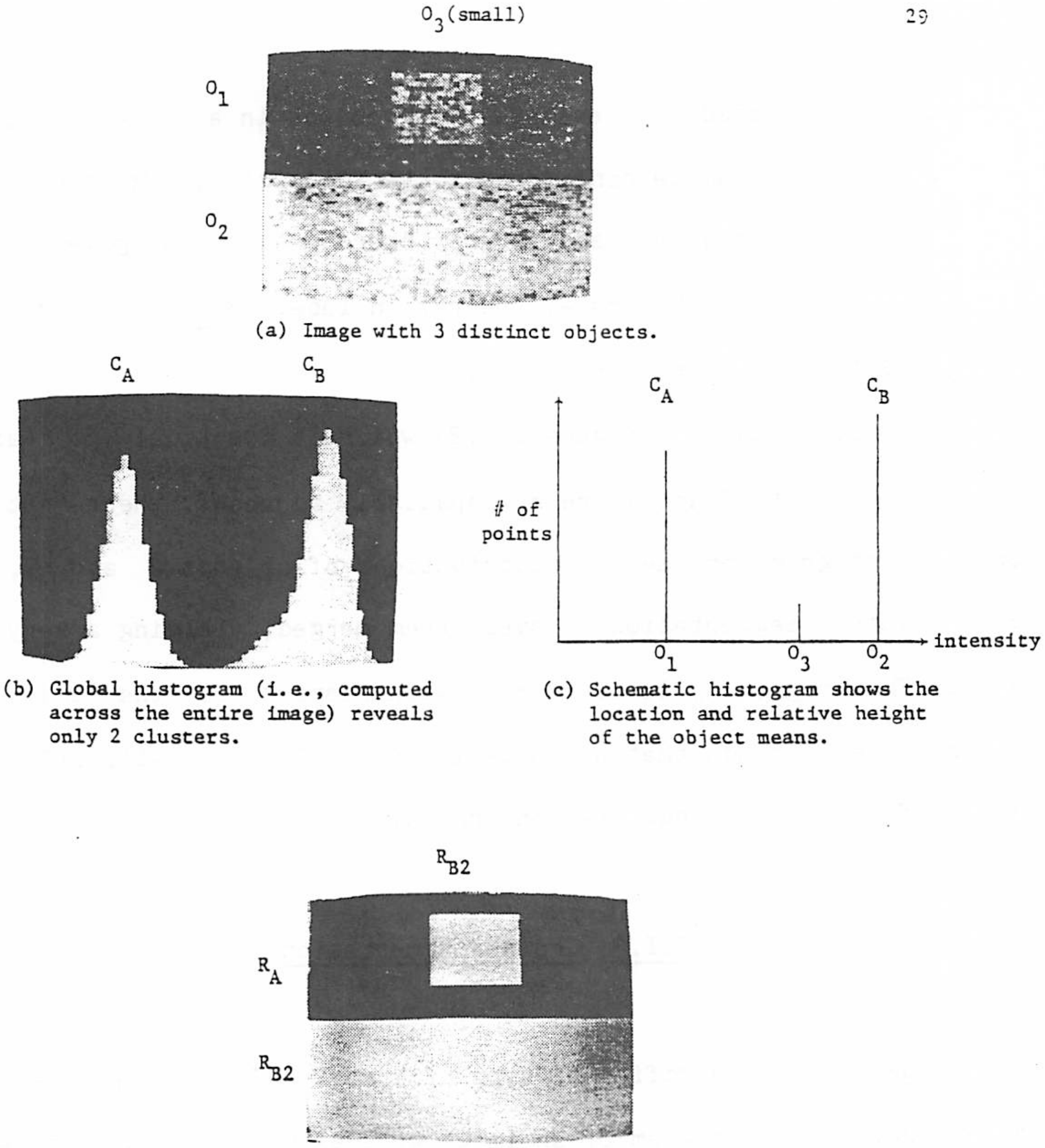
(d) Segmentation into two regions. The designation $R_{\alpha i}$ means "the region labelled by cluster $\alpha$, occupying the space covered by object i."

Figure III.0 A Simple Segmentation Algorithm Based on Feature Clustering.

comprising objects with visually distinct appearance will be labelled with the same symbol. One way for this to occur is when two objects have fairly close means. In this case, the distribution of the objects will sum and may not be detectable as separate peaks. Another possibility is when the distribution of a small object does not generate a detectable peak in the overall histogram.

The image in Case 1 (Figure III.1) contains three objects labelled $O_1$, $O_2$, and $O_3$. To a human observer, this image presents a rather trivial image processing task, since each object is clearly characterized by a unique average brightness level ($\mu_1$=20, $\mu_2$=40, $\mu_3$=50, $\sigma_1 = \sigma_2 = \sigma_3 = 3$). However, the histogram computed across the scene shows only two distinct clusters, labelled $C_A$ and $C_B$, because the variance, proximity of means, and size of $O_2$ masks the cluster of $O_3$. Thus, the information in the global feature space consists of two discriminable classes. A schematic view of the information in the diagram is shown in Figure III.1c. Here, it is evident that the histogram is actually composed of three distinct distributions; however only two distinct clusters are revealed due to the combined effects mentioned above.

Notice that the left side of cluster B has a slight shoulder, which is the only global indication that there is a third distribution. Let us assume that the shoulder is not detectable as a separate cluster. The region classification that results from only two clusters, $C_A$ and $C_B$, (Figure III.1d) contains only two distinct types of regions, $R_A$ and $R_B$. However, since object 2 and object 3 happen to



(a) Image with 3 distinct objects.

(b) Global histogram (i.e., computed across the entire image) reveals only 2 clusters.

(c) Schematic histogram shows the location and relative height of the object means.

(d) Segmentation into two cluster types, but 3 spatially distinct regions.

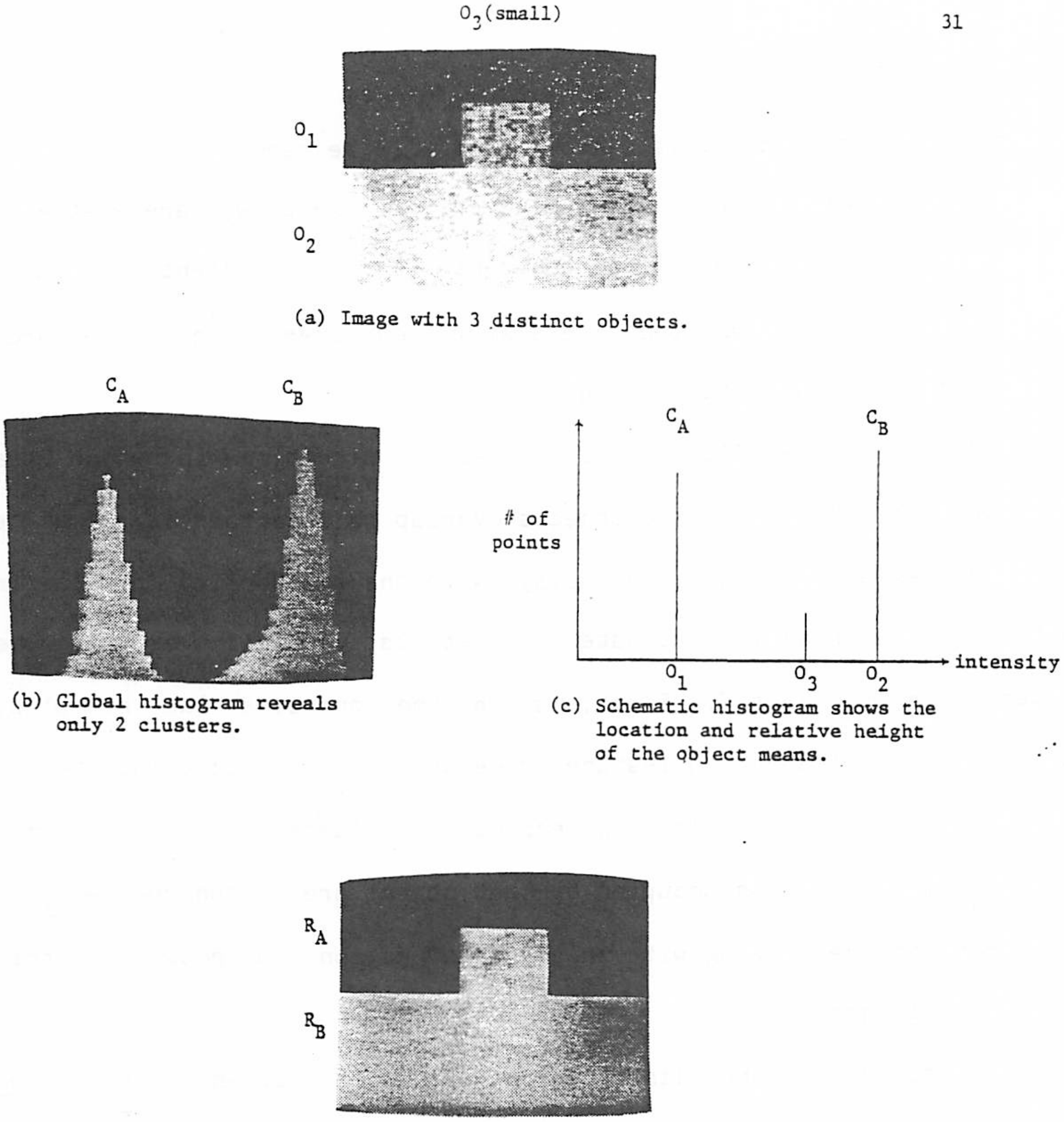
Figure III.1 Case 1, Overmerging - First Example: Distinct objects to not necessarily generate distinct clusters. However, the segmentation may still be successful depending on the spatial arrangements of the objects.

be spatially separated in the image by an object in a different class, the segmentation may be considered to be successful. The two regions labelled R are spatially disjoint and, therefore, can be given unique region labels. In this case, the region labels $R_{B2}$ and $R_{B3}$ refer to the objects to which they correspond.

Now consider Case 2 (Figure III.2) which is identical to Case 1 except that objects 2 and 3 are now spatially adjacent. Here, not only does the histogram confuse the distributions of objects 2 and 3, but the resulting segmentation leaves them merged, yielding a very poor result. Thus region $R_B$ is overmerged with respect to the underlying objects. We conclude that a change in object location will affect the quality of the global segmentation analysis.

## III.3 Gross Fragmentation

Let us consider a different type of error. It is incorrect to assume that the histogram of an image will either completely reveal a cluster or completely hide it. In fact, clusters can overlap to any degree. Fragmentation occurs whenever there is partial cluster overlap of distinct peaks and manifests itself as mislabelled pixels. The impact of fragmentation depends both on the degree of overlap of feature clusters as well as the spatial organization of the pixels involved. There is an obvious correlation of the percentage of overlap and the percentage of mislabelled pixels in an optimal Bayesian classifier [DUD73]. If the intensity values in an object are



(a) Image with 3 distinct objects.

(b) Global histogram reveals only 2 clusters.

(c) Schematic histogram shows the location and relative height of the object means.

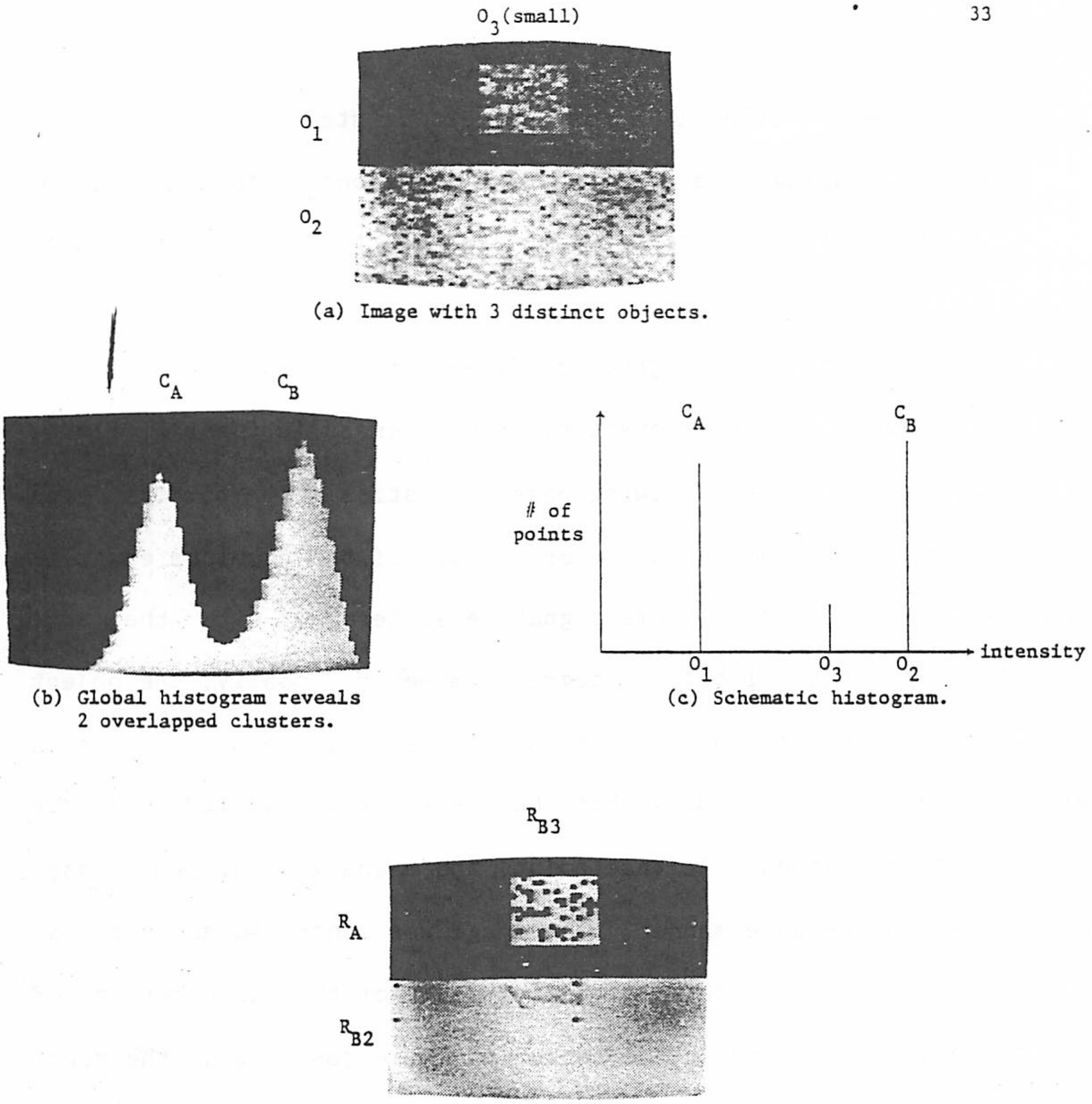
(d) Segmentation into 2 cluster types and 2 regions. Object 2 and object 3 have been segmented as 1 region.

Figure III.2 Case 2, Overmerging - Second Example: In the example, objects 2 and 3 are erroneously merged into a single region.

uncorrelated, then the mislabelled pixels will be randomly distributed across the regions involved. However, if the pixels are spatially correlated, as in the case of a non-zero feature gradient across a region, then the mislabelled pixels will themselves be correlated and, in fact, may form a viable region.

Case 3 (Figure III.3) illustrates a situation in which the distributions of all of the objects overlap to a certain degree. The resulting segmentation appears "noisy" with the mislabelled pixels in each region randomly located. Let us carefully examine the segmentation of object 3. First, as in the previous two examples, there is no cluster in feature space that uniquely discriminates it from the other objects. In this respect, the cluster labels that map onto the image location occupied by that object are not unique -- $O_3$ is thus prone to overmerging with an adjacent region (although in this example it is not).

Second, since the distribution of $O_3$ is hidden within two identified clusters, the classification of its pixels is guaranteed to consist of some mixture of two label types, neither of which provides a reasonable representation of the object. In this example, the mixture is such that 20% of the pixels are labelled by cluster A and 80% are labelled by cluster B. One might say therefore, that, at best, there is a 20% error rate in the labelling of this region. Fortunately, in this case, the fragmentation of $O_3$ into 2 cluster types has the desirable property that the minority cluster type ($C_A$) maps into random locations across the image space occupied by the object. For this



(a) Image with 3 distinct objects.

(b) Global histogram reveals 2 overlapped clusters.

(c) Schematic histogram.

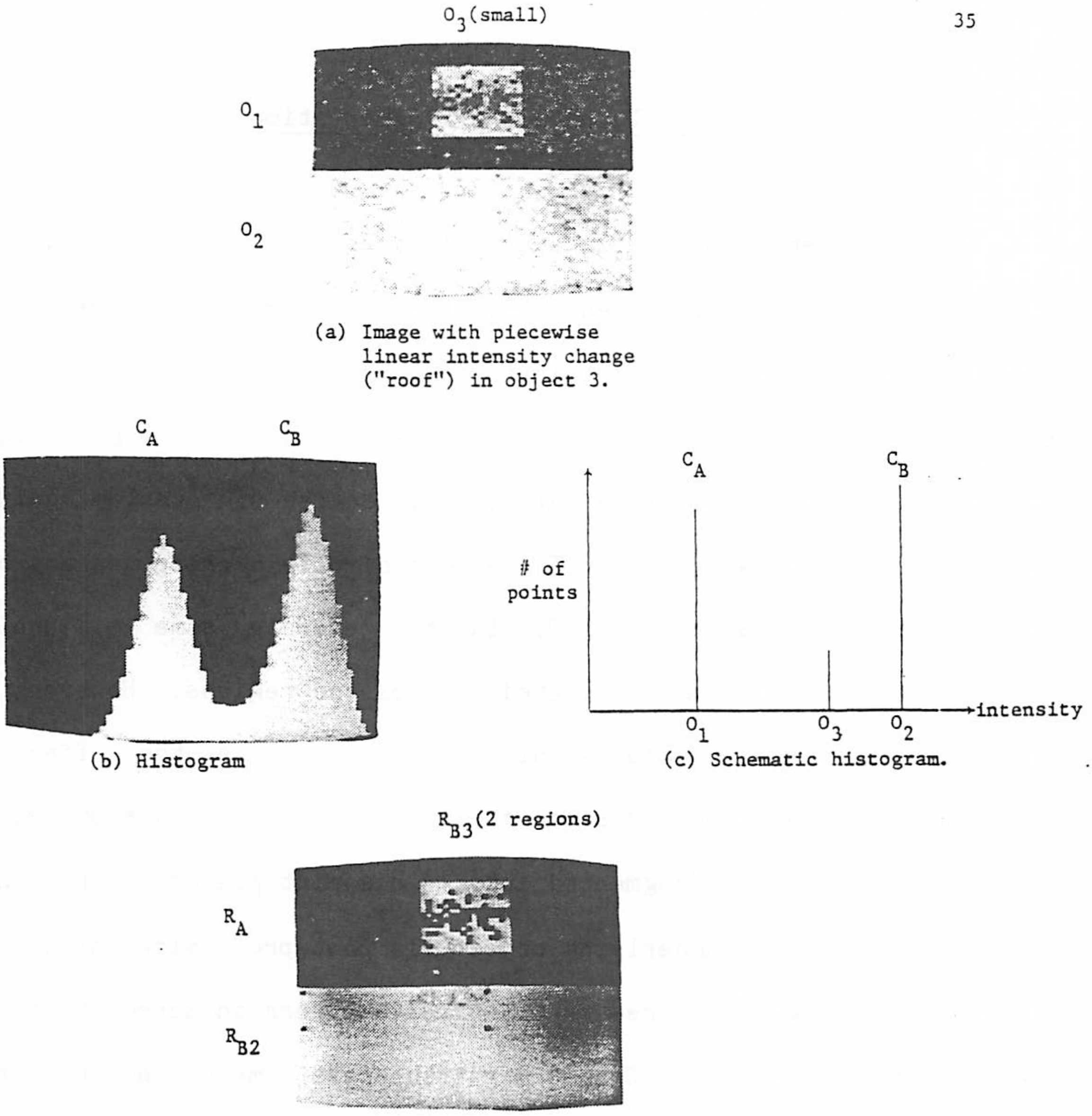
(d) All regions display erroneously labelled pixels. $R_{B3}$ is particularly fragmented since the mean of $O_3$ is hidden between two identified clusters.

Figure III.3 Case 3, Fragmentation - First Example: When clusters overlap the resulting segmentations appear "noisy" because the regions contain mislabelled pixels. Since they are labelled by the same cluster, $R_{B2}$ and $R_{B3}$ would have been overmerged if they had happened to be spatially adjacent as in Figure III.2.

reason, it may be possible, in a post-processing step, to recover the object by suppressing one and two pixel "regions" into the dominant region surrounding them, or to use some other sort of smoothing into large regions..

Now let us consider a slightly different image in which object 3 contains a piecewise linear intensity change across it (Case 4, Figure III.4). Object 3 has the following characteristics. First, its mean and variance are the same as they were in all of the previous examples -- thus, globally it has the same signature as before, with the same contribution to the global histogram. However, locally, the object consists of a series of bands: starting at the top, each row has a slightly lower mean intensity than the one above it, until the middle of the object is reached. At that point, the means gradually increase, row by row, in the same manner. The image was contrived so that most of the pixels in $O_3$ coincide with the left tail of the distribution of $O_2$. However, the pixels in the center band lie just inside the right tail of the distribution of $O_1$.

Once the clusters have been determined and the image pixels labelled, it is apparent that in addition to the randomly located errors, there is a connected set of errors at the center of object 3. This set of errors, in fact, forms a region that is just as viable, i.e. impervious to post-processing clean-up, as the two other major regions. Thus, object 3 has been fragmented into two regions even though the image does not contain an edge between those regions.



(a) Image with piecewise linear intensity change ("roof") in object 3.

(b) Histogram

(c) Schematic histogram.

(d) Most of the pixels in the center portion of the roof lie within cluster A. Therefore, object 3 is fragmented into 2 disjoint regions.

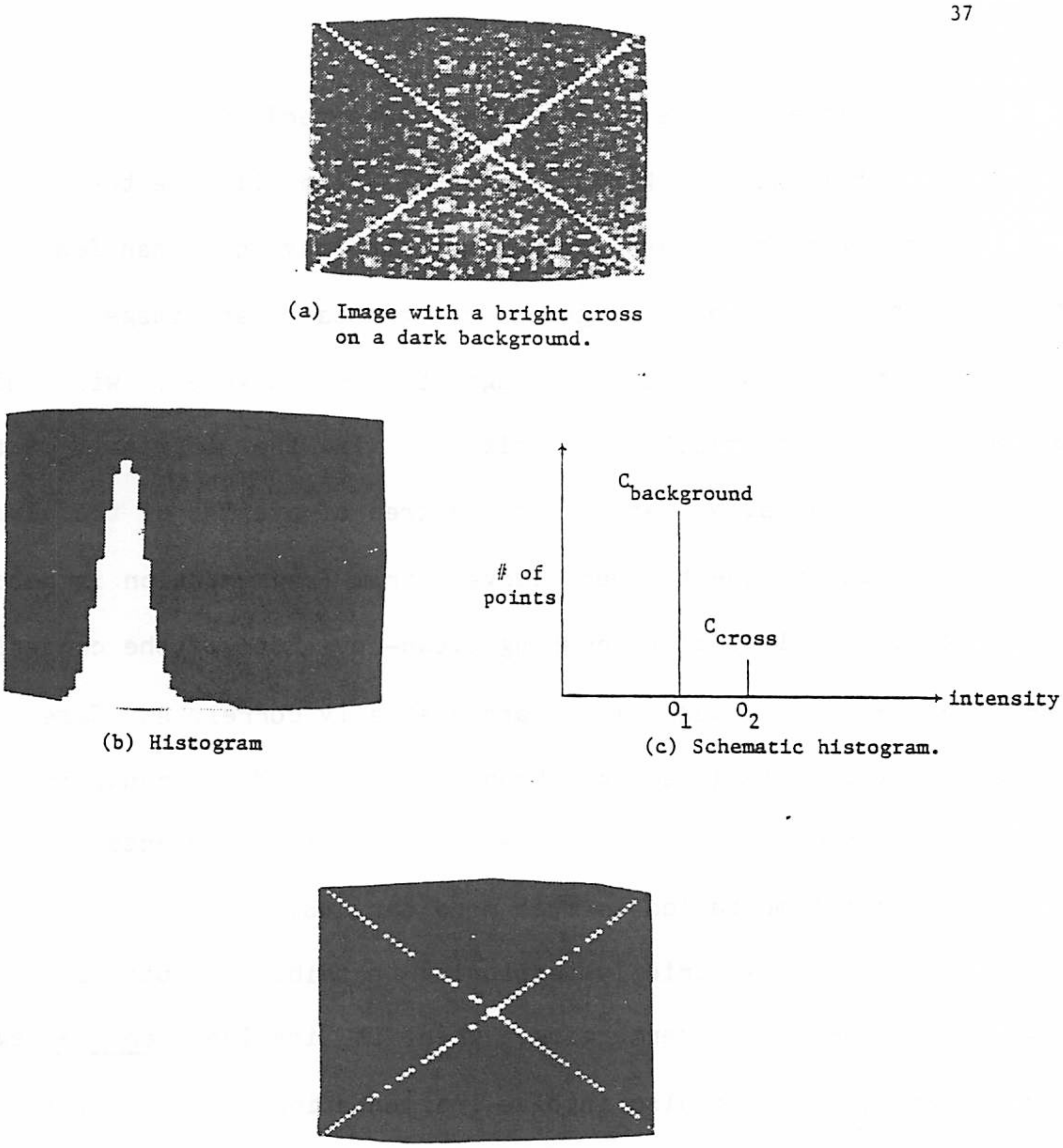
Figure III.4 Case 4, Fragmentation - Second Example: The effects of fragmentation depend both on the degree of cluster overlap as well as the spatial correlation of the feature values of the pixels involved.

## III.4 Thin Object Fragmentation

Next consider another instance of fragmentation, namely, when there are thin objects present in the scene. For our treatment here, structures will be considered thin if they are one or two pixels wide. Case 5 (Figure III.5) shows a cross-shaped object running through a background, and a histogram reveals a small degree of overlap between the object and the background. The effect of this overlap, as has been shown previously in section III.3, is to generate some mislabelled pixels that are randomly located across the regions. However, the mislabelled pixels that occur within the cross have the effect of breaking it into small disconnected pieces. In this example, the single cross object is fragmented into 18 disjoint pieces (18 regions), and recovery of the underlying object via post-processing may be very difficult. In fact, the one- or two-pixel suppression scheme mentioned above in the discussion of Case 3 might suppress some of the fragmented regions of the cross, thus making recovery even more difficult.

## III.5 Conclusion

This chapter has explored three types of segmentation errors that can result from cluster overlap in global feature histograms. These errors -- overmerging, fragmentation, and thin line fragmentation -- were shown to exist even in very simple, clearly structured test images. Their effect on noisy, multi-class natural images partly



(a) Image with a bright cross on a dark background.

(b) Histogram

(c) Schematic histogram.

(d) Cluster overlap fragments the cross into 18 disjoint regions.

Figure III.5 Case 5, Thin-line Fragmentation: Thin lines are particularly susceptible to fragmentation since even a single mislabelled pixel breaks the object into the disjoint regions.

explains why segmentations often appear ragged and unacceptable.

Overmerging and both kinds of fragmentation all have their origin in cluster overlap. Overmerging may or may not manifest itself depending on the spatial arrangement of objects in an image -- which is, in general, arbitrary. Fragmentation, however, will always manifest itself as mislabelled pixels. If the original data is uncorrelated spatially, and if the degree of overlap of the clusters involved is not too great, then recovery from fragmentation is possible by means of simple post-processing clean-up. But, if the converse is true, that is if the image pixels are spatially correlated (Case 4), or if the overlap is large enough, then the effect of fragmentation is to create large error regions. Moreover, if any of the objects are thin (Case 5) then fragmentation is much more serious.

We conclude by briefly exploring possible solutions to the problems discussed. Overmerging, when it involves <u>entire</u> regions (i.e., when it does not also involve fragmentation) is relatively easy, although costly to overcome. The solution proposed by Ohlander [OHL75] is to <u>recursively</u> decompose each region -- starting with the entire image -- until a stopping criterion is reached (refer to Figure II.1 and II.2). The criterion for region decomposition (splitting) is that a histogram of some feature is multi-modal. Thus, when all regions are unimodal in all features, the segmentation is complete. Fortunately, in practice, very few decomposition steps (usually only one or two) appear to be necessary for any given region.

The solution to the problem of fragmentation is complementary to the solution proposed for overmerging. Instead of attempting to decompose regions into sub-regions, it is desirable to merge regions that were erroneously split. Recall that the effect of fragmentation is to break an object into two or more regions even though no edge exists locally between the regions. One recovery technique is to examine the combined distribution for each pair of adjacent regions. If the distribution of some region pair is detectably multimodal, the boundary between them would remain intact. If, however, the combined distribution appears to be unimodal, then one may assume that the two regions are actually derived from one object and the boundary between them is removed. In a later chapter, a precise remerging statistic will be discussed.

Recovery from thin-line fragmentation is not only costly but extremely difficult. The remerging criterion above is inapplicable since the region fragments are not adjacent in the image. It seems unreasonably expensive and ill-defined to apply the remerging statistic to the rather large class of region pairs that are "almost adjacent." What is required is a local algorithm that can recognize what appears to be a significant local structure, e.g., a line, and which can induce pixels that are in the range of this structure to gravitate towards membership. It is in this spirit that the relaxation labelling process discussed in the next chapter has been defined.

# CHAPTER IV

# SEGMENTATION USING GLOBAL HISTOGRAMS AND ITERATIVE UPDATE

The previous chapter explored three kinds of errors that can arise from the global "feature-cluster/pixel-label" technique. These errors--overmerging, fragmentation, and thin line fragmentation-- can all be traced to the problem of cluster overlap. Cluster overlap, in turn, can be traced to an inadequancy of the histogram representation, that is, the lack of spatial information.

The focus of this chapter is on algorithms which can correct some of the errors that are introduced by the global technique. In particular, we will explore different forms of relaxation labelling processes (RLP's) that incorporate, with varying degrees of sophistication, contextual (i.e., spatial) information associated with each pixel. It will be shown that in many cases, given a first-approximation to pixel classification, neighborhood information can be manipulated to successfully correct errors. Figure IV.1 summarizes the segmentation algorithm that will be explored in this chapter.

## IV.1 Relaxation Labelling Processes

The general formulation of a probabilistic RLP requires the specification of a set of probabilities representing the degree of "class" membership to be associated with each "object" in some network.

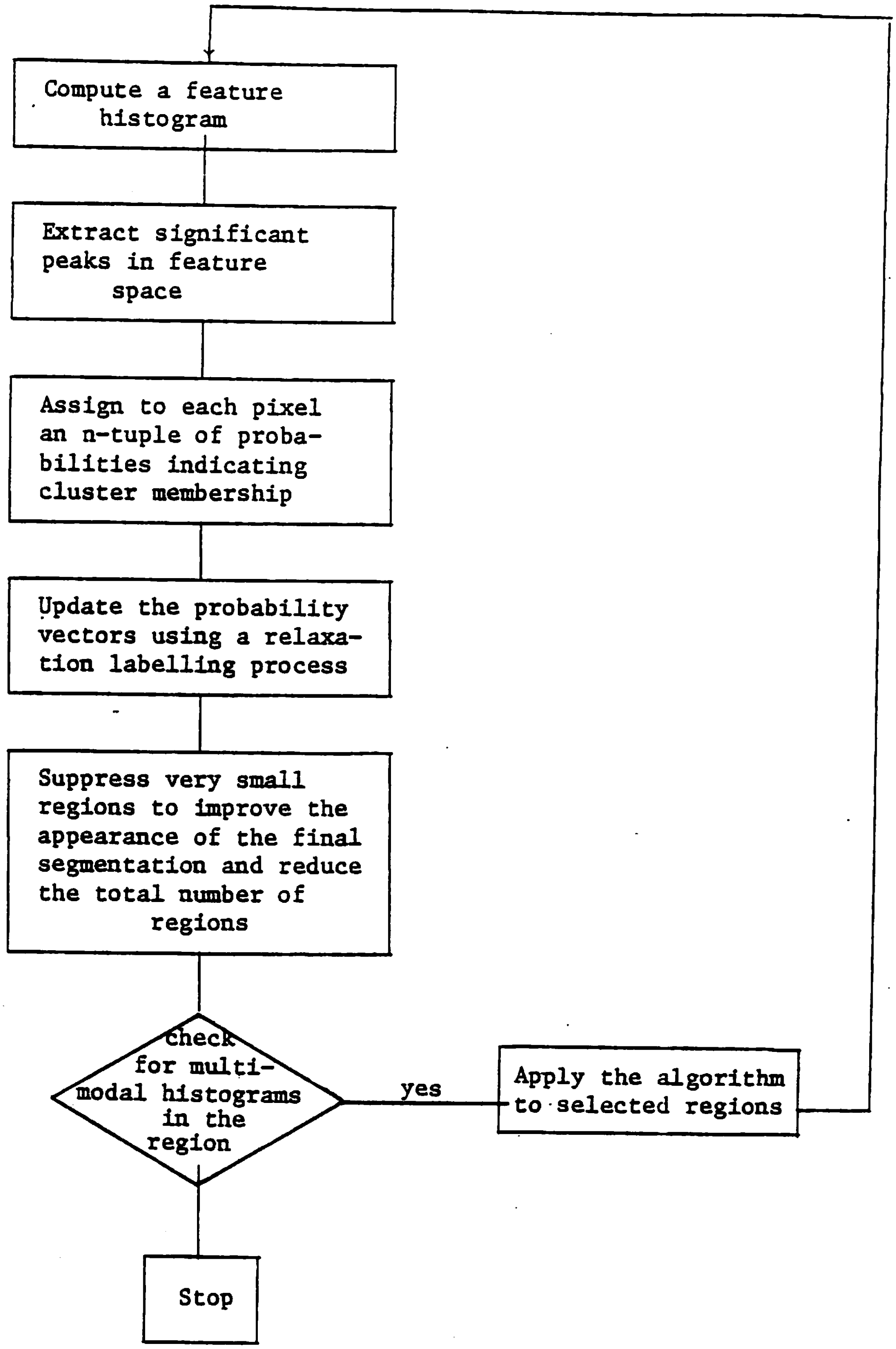

Compute a feature histogram
Extract significant peaks in feature space
Assign to each pixel an n-tuple of probabilities indicating cluster membership
Update the probability vectors using a relaxation labelling process
Suppress very small regions to improve the appearance of the final segmentation and reduce the total number of regions
check for multimodal histograms in the region
yes
Apply the algorithm to selected regions
Stop

Figure IV.1 Summary of the global segmentation algorithm with iterative update in a relaxation labelling process.

For our purposes, the classes correspond to clusters detected in feature space and the objects correspond to the pixels in the image. At each iteration, the probabilities of cluster membership associated with each pixel are adjusted according to the degree of support received from the probabilities at neighboring pixels. The adjustment or updating process is iterated with the expectation that there will be a marked reduction in the ambiguity of the initial classifications.

Let us examine two important characteristics of relaxation before giving the formal definitions of the process. First, there is the use of probabilities to indicate cluster affiliation. Recall that the histogram clustering technique explored in the previous chapter generated a discrete label indicating the cluster affiliation for each pixel. This label was the only indication of the location of the pixel in feature space. In the probabilistic formulation a precise "location vector" can be specified so that, for example, pixels in ambiguous locations in feature space (i.e., between clusters) can be encoded to reflect a lack of confidence of belonging to any particular cluster. The use of probabilities thus allows such pixels to defer their final labelling* until contextual information can be obtained.

The second important characteristic of the RLP lies in the use of compatibility coefficients which contain statistical information about the image. These coefficients are meant to reflect any detectable

*The term label refers to "class label" or, equivalently, "cluster label." The final labelling of a pixel is the distribution of the labels after the RLP has terminated.

spatial dependencies between labels. For example, wher appropriately specified, they can reflect directional tendencies cf objects in an image. Thus, an image that contains horizontal black bars on a white background might have compatibility coefficients such as:

$$\text{Compat}_{\text{VERTICAL}}(\text{black given white}) \quad = +1 \text{ (very likely)}$$

This can be interpreted as "the label indicating black is very likely to be vertically adjacent to the label indicating white." Similarly:

$$\text{Compat}_{\text{VERTICAL}}(\text{black given black}) \quad = -1 \text{ (very unlikely)}$$

Ideally, the compatibility coefficients should tend to anchor the iterative update of probabilities so that the final pixel labelling is not too far removed from the initial labeling. As will be shown in the section on results (IV.4.3), this property helps to inhibit the RLP from "eating away" thin structures. If only local information were used in the RLP, a thin structure (e.g., a one-pixel wide line) might be suppressed into the background as if it were uncorrelated "noise." However, if there is a sufficient sample of "line-like" objects, the compatibility coefficients will reflect this and bias the local update towards maintaining such lines.

## IV.2 Formal Definitions

Let us formally define the RLP as in [ROS76]. See also [ROS77,ZUC78] for a general discussion of RLP's. Let $A_1, A_2, \ldots, A_N$ be the pixels in the image and $C_A, C_B, \ldots, C_M$ be the labels associated with the clusters detected in feature space. Next, we must initially associate with each pixel $A_i$ an m-dimensional probability vector $(P_{iA}, P_{iB}, \ldots, P_{iM})$ whose component $P_{i\alpha}$ indicates the probability that $A_i \in C_\alpha$. Note that

$$0 \leq P_{i\alpha} \leq 1 \text{ and } \sum_{\alpha=1}^{M} P_{i\alpha} = 1.$$

The compatibility coefficients are specified as a mapping r from the set of quadruples (i, α, j, β) into [-1, +1]. One should interpret r in the following manner:

(a) if and are compatible for objects A and A , respectively, then $r(i,\alpha j,\beta) > 0$;

(b) if and are incompatible for A and A , respectively, then $r(i,\alpha,j,\beta) < 0$;

(c) if neither labelling is constrained by the other, then $r(i,\alpha,j,\beta) = 0$;

(d) the magnitude of r represents the strength of the compatibility.

It is apparent that one may interpret the coefficients as statistical correlation or mutual information [PEL78] since these functions behave in the manner of (a)-(d) above. Note that the compatibility coefficients are defined only for pairs of labels which are "adjacent" anywhere in the image, according to some local neighborhood adjacency relation. In our case, a canonical orientation dependent neighborhood $N_\theta$ will be defined;  will be quantized to 45 degree increments, e.g., N, NE, E, SE, S, SW, W, NW. Thus, relative to a given pixel, adjacent neighbors are those which are found in $N_\theta$ in the prescribed direction.

At each iteration t, we independently compute a new $P_{i\alpha}$ in the following heuristic manner:

$$P_{i\alpha}^{t+1} = P_{i\alpha}^{t}(1 + q_{i\alpha}^{t}) / \sum_{\alpha} P_{i\alpha}^{t}(1 + q_{i\alpha})$$

where

$$q_{i\alpha}^{t} = \sum_{j}\sum_{\beta} r(i, \alpha, j, \beta) * P_{j\beta}^{t}$$

and where j is an index over pixels in $N_\theta$. The denominator is a normalizing factor computed across the new probabilities of the m labels, so that the new values for $P_{i\alpha}$ will sum to one.

In practice it is useful to keep the probabilities of all labels non-zero because the updating of probabilities of each pixel label involves a multiplicative function. Once a label has probability zero, it would remain there during the iterative relaxation process. Therefore, the probability of each pixel label will not be allowed to drop below some small non-zero value. Notice that this heuristic equivalently implies that no label will ever reach probability 1. This will allow the probabilities of unlikely labels to grow if the context so demands, even for pixels whose current labelling includes a label with probability near one.

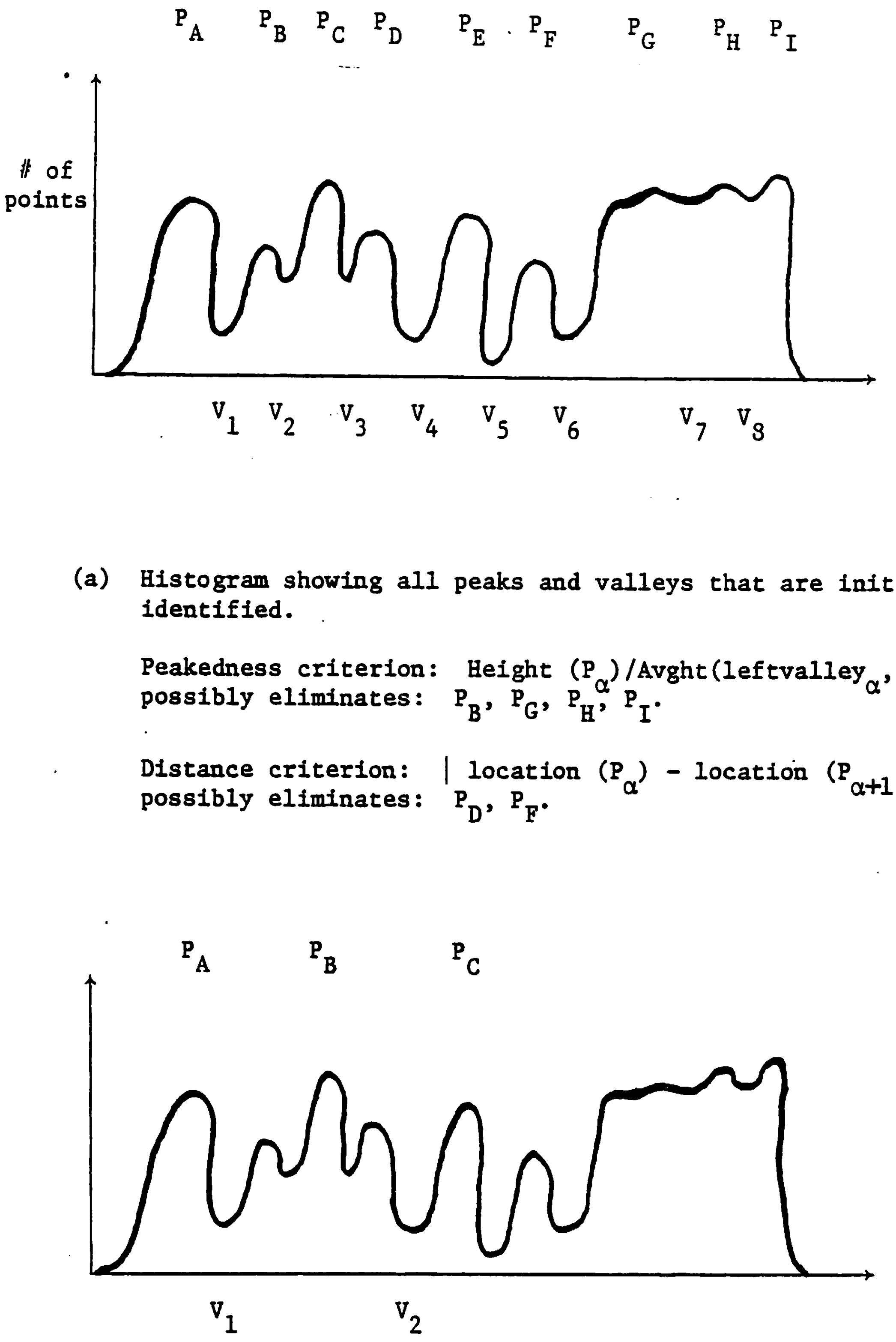
## IV.3 Initial Label Probabilities

The process of assigning an initial probability labelling to each pixel in the image will now be discussed. As previously mentioned, the label set and the label probabilities at each pixel are derived from the feature space selected to represent the image. The algorithm that will be discussed below assigns cluster-membership probabilities to each pixel as a function of the distance of a pixel to each of the cluster peaks. Thus, the algorithm is broken into two steps: detection of prominent cluster peaks and estimation of pixel-to-cluster relatedness.

### IV.3.1 Selection of Cluster Peaks

After looking at even a few histograms, it becomes obvious that the set of useful cluster peaks is a subset of the set of local peaks. Distributions of natural scenes tend to be extremely jagged and do not have clearly defined cluster locations. It is frequently a very difficult task, even for a human, to parse a histogram into its cluster components. Indeed the problem of automatic detection of clusters, although traditionally in the realm of statistical pattern recognition [DUD73] can also be approached as an image processing task. The following discussion will be limited to one-dimensional histograms. Later, in section IV.6.2, cluster detection will be explored for two-dimensional histograms.

What criteria are necessary for a cluster isolation mechanism? Consider the problem of clustering the histogram shown in Figure IV.2. First, it seems important that cluster centers be strongly peaked, i.e., each true peak should have a significantly greater "height" than



(a) Histogram showing all peaks and valleys that are initially identified.

Peakedness criterion: Height $(P_\alpha)$/Avght(leftvalley$_\alpha$,rightvalley$_\alpha$)
possibly eliminates: $P_B$, $P_G$, $P_H$, $P_I$.

Distance criterion: | location $(P_\alpha)$ - location $(P_{\alpha+1})$ |
possibly eliminates: $P_D$, $P_F$.

(b) Final peak and valley labelling misses the "plateau" of $P_G + P_H + P_I$.

Figure IV.2 An Example of Peak Selection.

the height of its surrounding valleys, and the valleys should probably not be too far apart. Second, it is reasonable to require that peaks should be somewhat separated from each other; otherwise they probably represent the same cluster. Both of these criteria will help to eliminate "false" peaks and thereby lead to a more useful cluster analysis.

A possible peakedness criterion is simply the ratio of the peak height to its surrounding valleys. That measure is sufficient for the first peak ($P_A$) in the figure, but is ill-defined for $P_B$ since the valleys in the latter case are rather unequal in height. The peakedness ratio can be modified so that a peak is compared to the average height (avght) of its valleys, or perhaps a better comparison is the larger of the two valleys (maxht). Formally, let us express two functions of peakedness, $F_1$ and $F_2$, computed for the $\alpha$th peak and its valleys as follows:

(1) $F_1\ (\text{Peak}_\alpha,\ \text{leftvalley}_\alpha,\ \text{rightvalley}_\alpha) = \text{height}(P_\alpha)/\text{avght}(\text{leftvalley}_\alpha,\ \text{rightvalley}_\alpha)$

or

(2) $F_2\ (\text{Peak}_\alpha,\ \text{leftvalley}_\alpha,\ \text{rightvalley}_\alpha) = \text{height}(P_\alpha)/\text{maxht}(\text{leftvalley}_\alpha,\ \text{rightvalley}_\alpha)$

For any peak $\alpha$ and some threshold $\theta$, if $F < \theta$, the peak will be discarded and considered to be a false cluster center. In the figure, $P_B$ is probably discardable (by a reasonable setting of $\theta$), while $P_D$ is somewhat ambiguous, but probably can be extracted. However, $P_G$, $P_H$, and $P_I$ are problematic because they are not individually peaked in any reasonable sense, even though they do seem to form a meaningful cluster. The structure that these three peaks form—called a plateau—must be detected by other means. In particular, the algorithm must detect a "run" of relatively flat peaks. A central location can then be selected to represent the plateau. Plateau detection was not included in the current system because it was difficult to obtain a satisfactory implementation. Fortunately, in practice, "pure" plateaus do not seem to arise very often.

Next let us consider the second criterion for peak selection which is based on inter-cluster distances. It seems reasonable that peaks that are very close to each other are not truly indicative of distinct object classes, and may be considered part of the same cluster. Thus, for each pair of peaks, we define

$$D_i(\text{Peak}_\alpha, \text{Peak}_{\alpha+1}) = |\,\text{loc}(P_\alpha) - \text{loc}(P_{\alpha+1})\,|$$

When $D_i < \theta'$ for some given $\theta'$, then the smaller peak will be discarded as a false peak. In the example of Figure IV.2, peaks $P_D$ and $P_F$ are potentially eliminated, since they are relatively close to the larger peaks $P_C$ and $P_E$, respectively.

A reasonable peak and valley labelling is given in IV.2(b). The result is somewhat unsatisfying, since

(1) the plateau is completely missed, and

(2) $P_F$ probably should not have been eliminated.

The reason that one might argue that $P_F$ should be kept—even though it is very close to $P_E$—is that it is extremely peaked. Thus, we conclude that the two criteria for peak selection cannot be applied independently. One solution is to apply the distance measure conditionally, e.g. only if the peakedness measure is less than some amount. Another possibility is to compute the product of the two measures so that when either is high, the peak will be kept. These questions will be left for future empirical investigation; in the following it is assumed that a reasonable and representative histogram parsing has been obtained.

### IV.3.2 Assignment of Probabilities of Peak/Label Membership

After identifying the prominent peaks, the next step is to link this information with the spatial distribution of information in the image. We want to recode each pixel so that it reflects its location in feature space relative to the peaks. In this manner, groups of pixels which are near each other both in feature space and in image space can be merged and labelled as belonging to the same region.

Given a set of peaks, $P_A$, $P_B$, ...,$P_M$ and a set of pixels, $A_1$, $A_2$, ..., $A_N$, we compute for each pixel $A_i$ the following probabilities:

$$P(A_i \text{ is labelled } \alpha) = \frac{\frac{1}{d_{i\alpha}}}{\sum\limits_{\alpha} \frac{1}{d_{i\alpha}}}$$

where $d_{i\alpha}$ is the Euclidean distance in feature space from $A_i$ to $\alpha$. This measure is a monotonically-decreasing nonlinear function of the Euclidean distance of a point in feature space to the $\alpha$th cluster center. For the special case of $d_{i\alpha} = 0$, the distance is reset to some small number $\epsilon > 0$. This guarantees that no label will have either probability 1 or probability 0.

### IV.3.3 The Neighborhood of a Pixel

Once the initial labelling has been computed, it is necessary to define a standard neighborhood around each pixel. Some examples are given in Figure IV.3. Note that for simplicity the choices have been limited to the 3x3 area surrounding a pixel, although this can easily be enlarged with subsequent impact on the speed of label updates and the results obtained.

Each of the neighborhood representations uniquely affects the performance of the local algorithms; for example, a diagonal line that cuts through a uniform background may be missed in a 4-adjacent neighborhood or relatively under-represented in an 8-adjacent neighborhood. Moreover, it will be shown that the weighting of the central pixel strongly influences the rate at which it can be changed from its current value.

### IV.3.4 The Compatability Coefficients as Conditional Probabilities

Now that the parameters of the local environment have been defined formally; the discussion now concentrates on the global information that is to be gathered for the RLP, namely, the compatability coefficients. The compatibility coefficient between each pair of labels defines whether labels of neighboring pixels support each other or compete with each other.

| | | |
|---|---|---|
| | X | |
| X | | X |
| | X | |

(a) "4-adjacency neighborhood"

| | | |
|---|---|---|
| | X | |
| X | X | X |
| | X | |

(b) "5-adjacency neighborhood" is composed of 4-adjacency and center pixel

| | | |
|---|---|---|
| X | X | X |
| X | | X |
| X | X | X |

(c) "8-adjacency neighborhood"

| | | |
|---|---|---|
| X | X | X |
| X | X | X |
| X | X | X |

(d) "9-adjacency neighborhood" is composed of 8-adjacency and center pixel

| | | |
|---|---|---|
| $W_1$ | $W_2$ | $W_3$ |
| $W_4$ | $W_5$ | $W_6$ |
| $W_7$ | $W_8$ | $W_9$ |

(e) A "weighted neighborhood" is a generalization of cases (a)-(d) where the $W_i$ are weights selected for some purpose.

Figure IV.3 Neighborhood Definitions.

The coefficients may be defined to be positive for identical labels and negative for differing labels. This is reasonable in images composed of large blobby regions with relatively few boundary points. Here, the typical interaction is between labels of the same type (positive correlation of label vs label , while interactions between labels of different types are relatively infrequent. The simplest specification of compatibility coefficients is to restrict them to signed unary values:

$$r(i,\alpha,j,\beta) = +1 \quad \text{if } \alpha = \beta \qquad (1)$$
$$r(i,\alpha,j,\beta) = -1 \quad \text{if } \alpha \neq \beta$$

This arrangement works reasonably well in areas lacking fine structure, but, in general, it is more desirable to have the coefficients reflect the way pairs of labels spatially co-occur in the image. In this way structured objects that display directionality, e.g. thin diagonal lines, can be given increased weighting in the probability update on the basis of their statistical significance in the image. Peleg and Rosenfeld [PEL78] have suggested the use of the mutual information of the two labels to capture the way labels co-vary. However, we suggest a simpler variation using conditional probabilities [ZUC78] because they also reflect the desired label dependencies (a-d in section IV.2). Note that Zucker rules out the use of statistical correlation for compatibilities since it is a symmetic measure of dependence, whereas asymmetric configurations of labels often arise (e.g., when one object is above another).

Let $P_i(\alpha)$ denote the initial estimate of the probability that pixel i is labelled α. Then

$$\overline{P}(\alpha) = \frac{1}{N} \sum_{i=1}^{N} P_i(\alpha)$$

is a global estimate of the a priori probability of α across the entire image. It is directly related to the average distance in feature space of pixel values to the αth cluster center*. Therefore, it is related to the density of pixels around the αth cluster in the histogram. (However, the reader should note that the situation is somewhat more complex because the values of $P_i(\alpha)$ are a function of the position of other clusters in feature space and the density of pixel values around them also).

The joint probability of a pair of points having labels α and β at some orientation, say east(e), can be estimated by

$$\overline{P}_{ie}(\alpha,\beta) = \frac{1}{N} \sum_{i=1}^{N} P_i(\alpha) * P_e(\beta)$$

We can now estimate the conditional probability that i is labelled α given that the east pixel e is labelled β, by

$$P_{ie}(\alpha|\beta) = \frac{\overline{P}_{ie}(\alpha,\beta)}{\overline{P}(\beta)} = \frac{\sum_{i=1}^{N} P_i(\alpha) * P_e(\alpha)}{\sum_{i=1}^{N} P_i(\beta)}$$

Two labels are independent in direction θ if the pair of labels co-occur with the same probability as the product of their individual

*More precisely, it is the average distance to $C_\alpha$ divided by the sum of the average distance to each cluster $C_\alpha$, α=1, M.

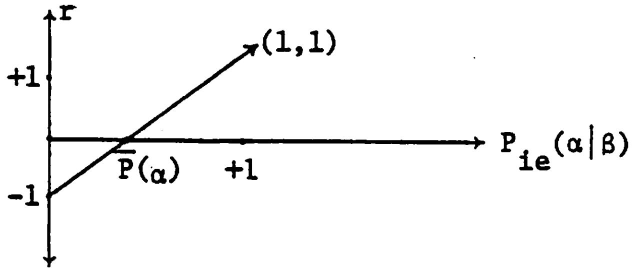
probabilities. Then in our example of pixels at the relative spatial orientation of east, their independence implies that

$$\overline{P}_{ie}(\alpha , \beta) = \overline{P}(\alpha)\overline{P}(\beta).$$

and in terms of conditional probabilities,

$$P_{ie}(\alpha \mid \beta) = \overline{P}(\alpha)$$

This latter case is the situation where β at a pixel oriented at east gives no information about α . Thus, the point at which r(i,α,j,β) = 0 can be defined to be the prior of α . If r is to range between -1 and +1, then it can be defined in terms of a piecewise linear interpolation function:



Note that the coefficients are no longer symmetric: r(i,α,j,β)   r(i,β,j,α). It is also worth mentioning that viewing compatability coefficients directly as conditional probabilities leads to an updating scheme which can be formulated in Bayesian probability theory [RIS79]. Here, we have used a heuristic formulation to derive the coefficients from the joint probabilities.

Note that the above formulation is ill-defined if we wish to include a pixel as its own neighbor. However it has been empirically shown to be desirable to inhibit the RLP from straying too far from the initial labelling on any given iteration. Consequently, in this case, we extend our definition of a local neighborhood and compatibility coefficients:

$$r(i,\alpha,i,\beta) = +1 \quad \text{if } \alpha = \beta$$

$$r(i,\alpha,i,\beta) = -1 \quad \text{if } \alpha \neq \beta$$

This means that for neighborhoods in which the center pixel is included, all labels at the central pixel are +1 compatible with themselves and -1 compatible with all other labels.

### IV.3.5 Problems with Compatability Coefficients

The definition of compatability coefficients, either as mutual information or in terms of the scheme just presented, has two possible weaknesses. First, since the measurements are computed across the entire image, they reflect the "average" image structure. Infrequently occuring spatial structures might not make any significant contributions to the overall accumulation of compatibility statistics. This can be dealt with by localizing the compatability coefficients to smaller sections of the image, where local structures will occur with a higher relative frequency. Of course, this would increase the amount of storage required to maintain the coefficients, as well as creating problems with pixels lying along section boundaries.

A second problem with our definition of compatability coefficients involves the geometry of regions and the different kinds of information which combine into the joint probability of neighborhoods of labels. If a region is large and compact (i.e., its ratio of area to perimeter is relatively large), then there are many more interior pairs of adjacent pixels than boundary pairs of pixels. Thus, the coefficients can be dominated by large contributions in all the orientations from pixels which lie internal to the region. In this case, the smaller amounts of information associated with the region boundary, which may in fact be highly orientation sensitive, may be lost. It may be possible to remedy this situation by maintaining two sets of compatability coefficients: one set for those pixels that are estimated to lie inside of objects and another set for those pixels that are estimated to lie along boundaries. Of course this will only work if there is some means of determining which pixels are likely to be on boundaries. In the case where differences within a region are not expected to be great, the determination of edge given interior can be based upon differences in pairs of pixel values.

## IV.4 Three Variants of Relaxation for Empirical Analysis

This section will show the results of applying the algorithm explored in the previous sections to an artificially generated image. The results will demonstrate the behavior of three variants of the relaxation algorithm. The first is probabilistic relaxation with "simple" compatibility coefficients, namely, for all neighbors j:

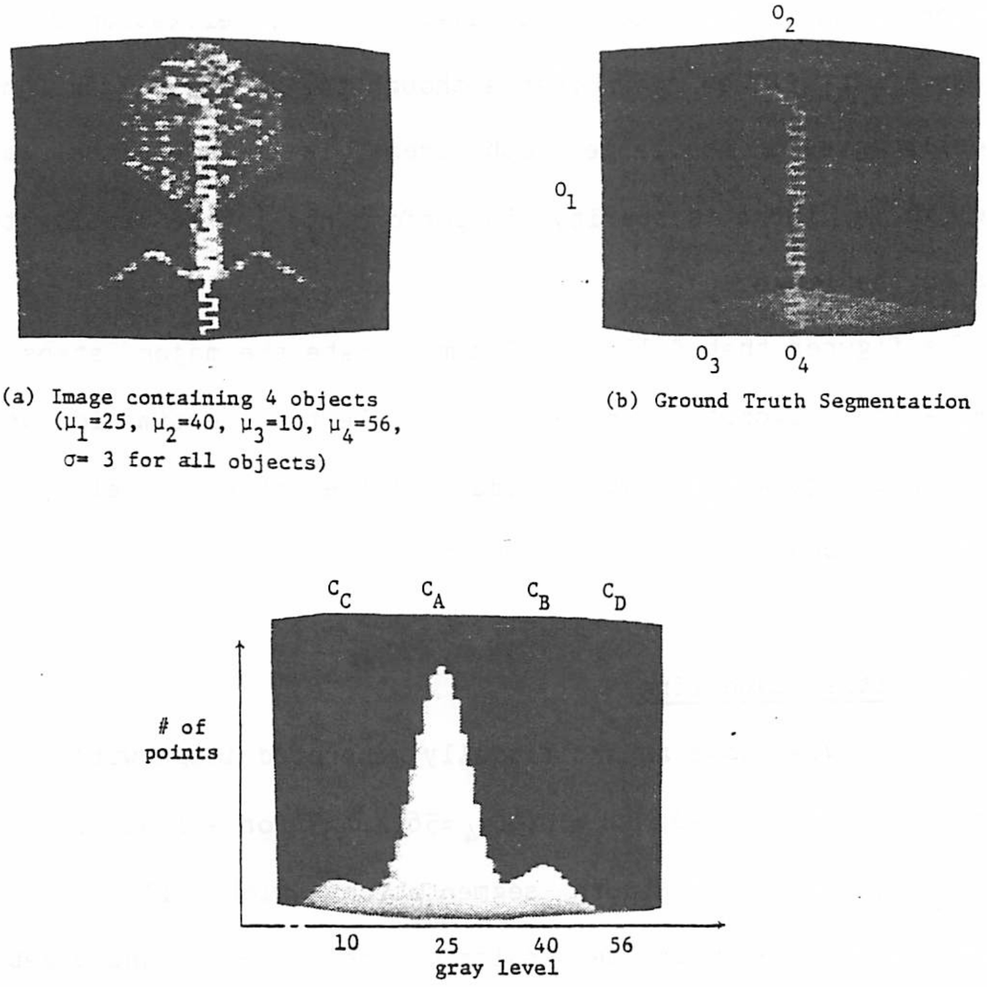
$$r(i,\alpha,j,\beta) = +1 \quad \text{if} \quad \alpha = \beta$$

$$r(i,\alpha,j,\beta) = -1 \quad \text{if} \quad \alpha \neq \beta$$

The second variant uses probabilistic relaxation with conditional probabilities for coefficients (as defined in IV.3.4). Finally a degenerate form of discrete relaxation will be presented, called *plurality* update. In this scheme, both the label probabilities and the label compatibilities are discarded. The algorithm initially assigns the most likely label to each pixel, e.g. via a minimum distance classifier. Next, an update rule is applied which consists simply of selecting the most frequently occurring label in the neighborhood of each pixel. This is equivalent to a mode filter [COL78] except that it is applied to labels instead of pixel intensities.

As will be shown later in this chapter, it is extremely difficult to clearly evaluate the effects of different segmentation algorithms when these algorithms are applied to natural scenes. This difficulty is due to the high degree of noise, edge blurring, irregular texturing, etc. typically present in non-trivial natural scenes. The presence of these anomalies implies that accurate ground truth segmentations are difficult or impossible to obtain. Hand-drawn segmentations are inevitably prone to errors at the object boundaries and tend to reflect implicit and explicit biases of the human perceiver.

In response to this, we have designed a series of artificial scenes in which each region has well-defined boundaries. The feature data for each object i are distributed normally $(N(\mu,\alpha))$ and then placed at random locations across the object (e.g., the distribution is spatially uncorrelated). The image in Figure IV.4a was designed so that the distributions of the individual regions would have a reasonable degree of overlap. In addition, attention was given to the creation of thin, varying spatial structures that might test the



(a) Image containing 4 objects ($\mu_1$=25, $\mu_2$=40, $\mu_3$=10, $\mu_4$=56, $\sigma$= 3 for all objects)

(b) Ground Truth Segmentation

(c) Overall histogram reveals 4 partially overlapped clusters

Figure IV.4 Artificial image used in the next set of experiments. All objects have normal distributions and the pixels have a spatially random distribution within each object. Notice that there is a 1-1 correspondence between the numeric ordering of the objects and the alphabetic ordering of the clusters.

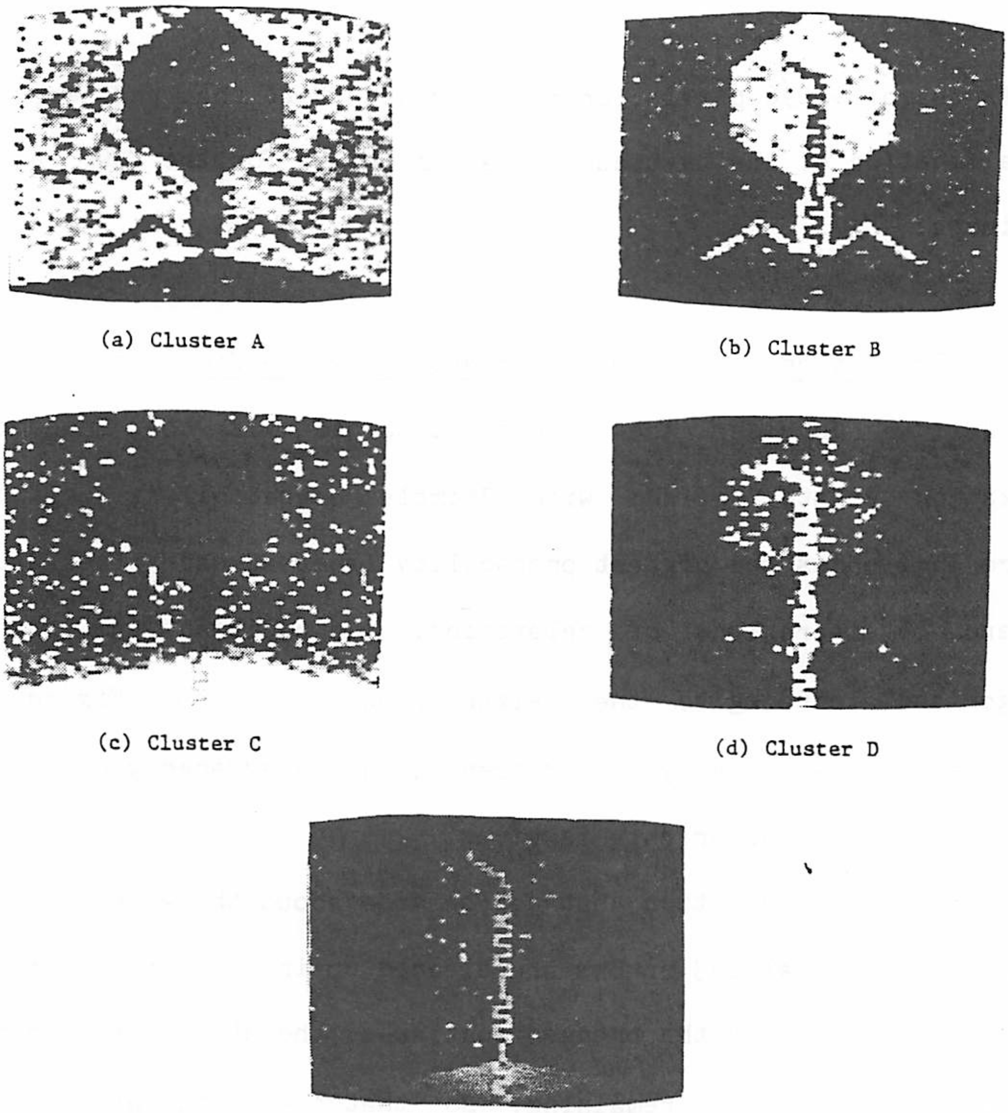
behavior of our iterative, spatially-sensitive, relaxation segmentation process. It will be shown that although the various algorithms tested basically agree in the large blobby areas, it is indeed the case that there is a large disparity in performance in the finely structured areas in the image.

The figures that follow will demonstrate the major steps of the segmentation algorithm, namely, peak selection, estimation of initial pixel labelling, and iterative update of the pixel labels using the three variations of relaxation specified above.

### IV.4.1 Initial Labelling

Figure IV.4 shows an artificially generated image with 4 labelled objects ($\mu_1$=25, $\mu_2$=40, $\mu_3$=10, $\mu_4$=56, $\sigma$=3 for all objects). Figure IV.4b shows the ground truth segmentation which will be used for comparison purposes with the results of the three variant update rules. The histogram of the scene shows four clusters which are identified and labelled by the peak detection algorithm explored in Section IV.3.1.

Figure IV.5 shows the result of assigning cluster probabilities to each pixel. Each of the four probability-images is displayed with probabilities in terms of gray levels. Black is interpreted as a very low probability of belonging to a cluster, while white implies a very high probability of belonging to a cluster. In addition, there is an image that indicates the highest probability label at each pixel. These labels may be compared pixel-by-pixel with the labels in the ground truth segmentation to establish a base-line error rate of 233



(a) Cluster A

(b) Cluster B

(c) Cluster C

(d) Cluster D

(e) Minimum distance classification of pixels into cluster types. There are 233 mislabelled pixels. (5.7%)

Figure IV.5 Initial pixel labelling based on the 4 peaks detected in the intensity histogram of the image.

pixels out of 4,096 pixels, or 5.7 percent. Note that this error rate is a function of the particular $\mu$ and $\sigma$ chosen for the four regions in the image.

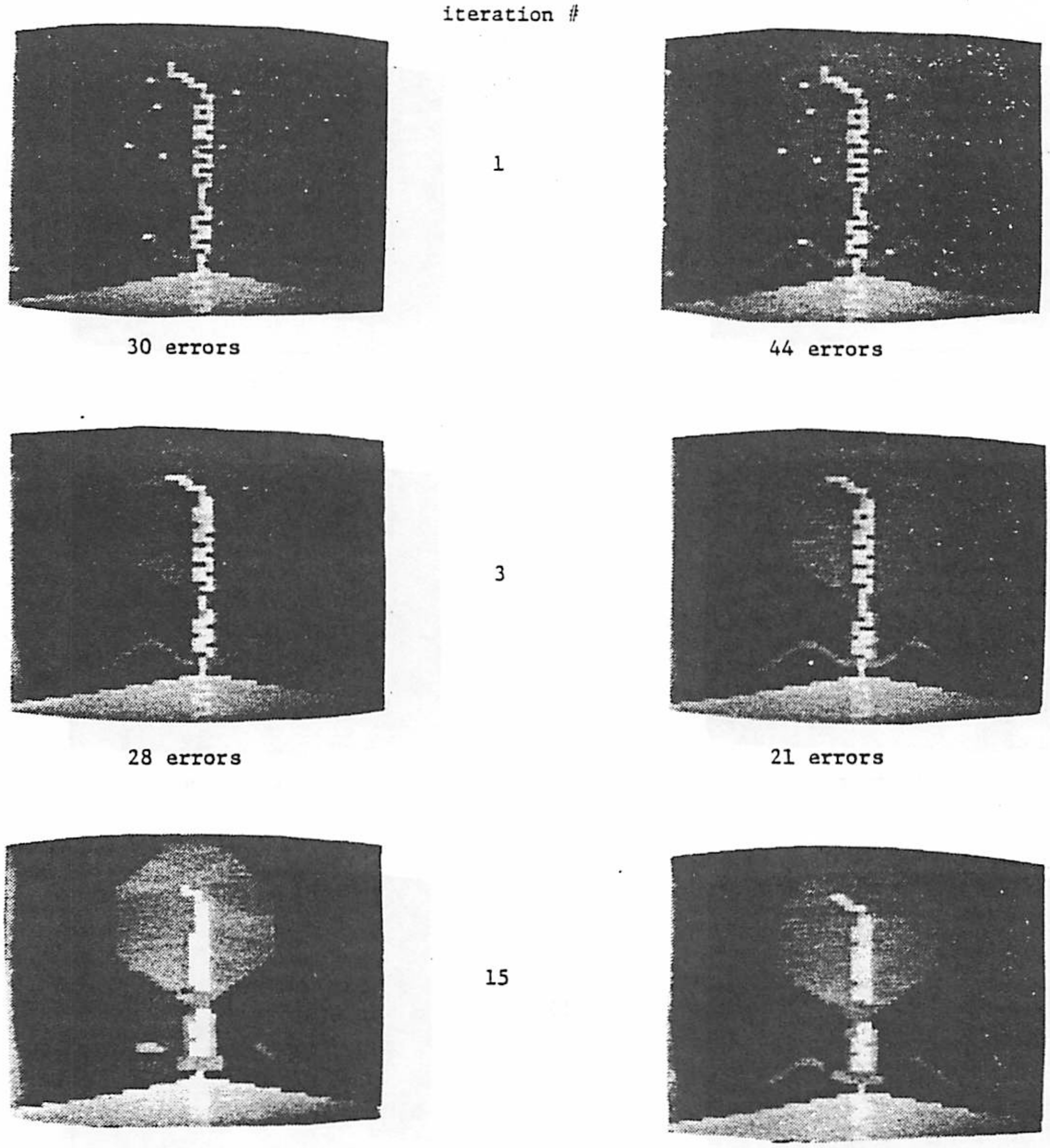
### IV.4.2 Relaxation Using Simple Compatibility Coefficients

Next we consider the results obtained via the probabilistic relaxation update defined with "simple" compatibility coefficients. Figure IV.6 shows the highest probability label at each pixel after 1, 3, and 15 iterations of relaxation. In addition, the results are categorized according to the neighborhood used (see Figure IV.3): 4-adjacency, 5-adjacency, 8-adjacency, and 9-adjacency neighborhood. In each case, the error rate is given.

The first observation that can be made about these results is that most of the initial 233 errors are cleaned up in the first iteration of the RLP. However, as the process continues, the RLP clearly introduces new errors to those remaining, so that the segmentations after 15 iterations* are much worse than the results after 1 iteration. Moreover, all of the errors that are introduced occur in the thinly structured areas (e.g. $O_4$ and the diagonal appendages of $O_2$) of the image: just those areas of the image that are desirable to preserve!

The explanation is that the pixels that are interior to objects are being strengthened at a rate much faster than those along the

*Note that in all cases, the RLP converged by iteration 15; that is, all pixels had a single label with probability 1.



30 errors

44 errors

28 errors

21 errors

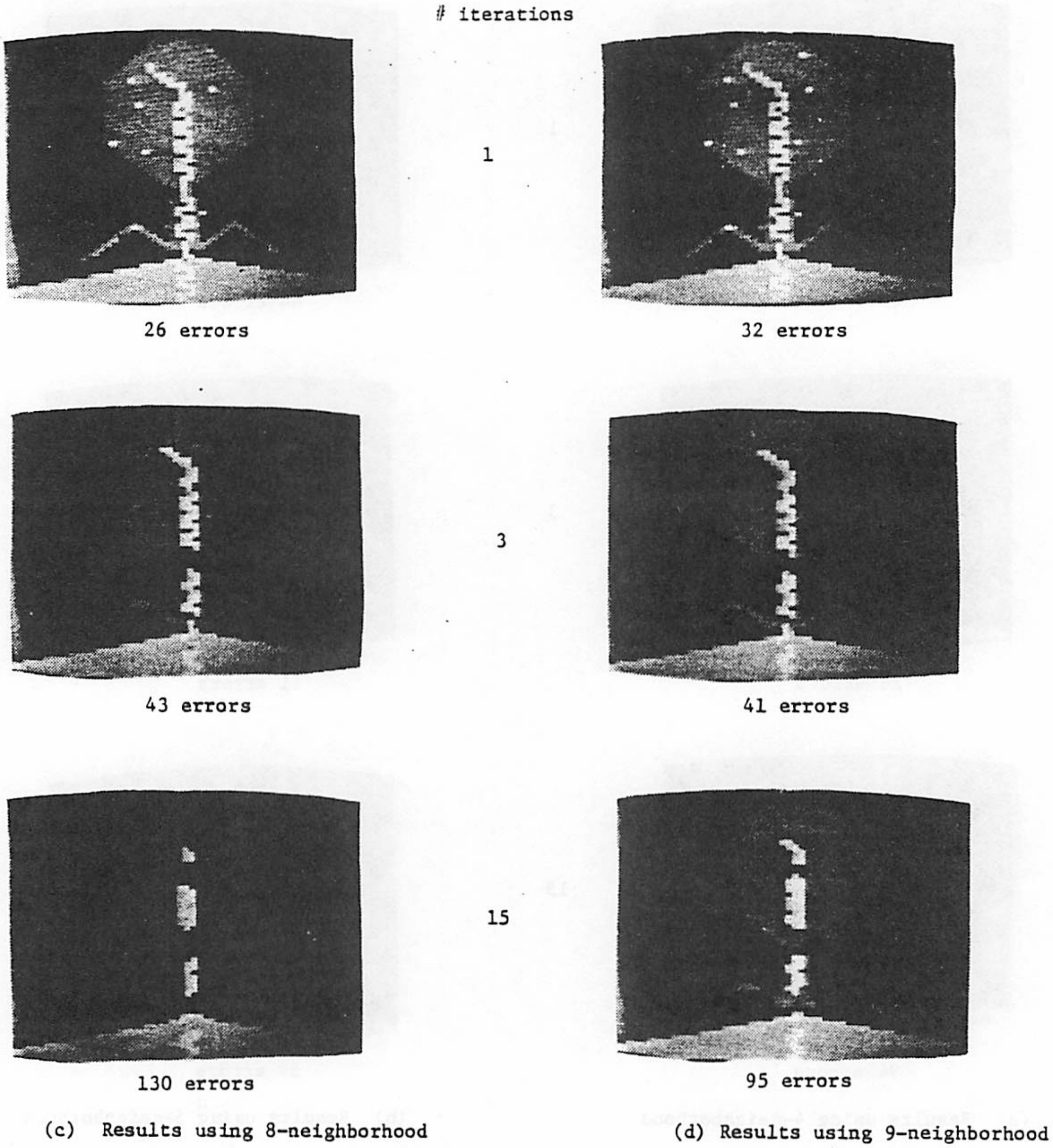
94 errors

59 errors

(a) Results using 4-neighborhood

(b) Results using 5-neighborhood

Figure IV.6 Probabilistic relaxation using simple compatibility coefficients across indicated neighborhoods. (Continued)



(c) Results using 8-neighborhood

(d) Results using 9-neighborhood

Figure IV.6 Relaxation using simple compatibility coefficients across indicated neighborhoods.

boundaries of objects. The boundary pixels therefore are relatively weakened and, in the case of the unstable (thin) configurations, the once dominant label is ultimately suppressed by the competing label. This behavior can be traced to the action of the compatibility coefficients.

Let us consider a pixel in the interior of an object and refer to it as $P_{INT}$. It often has one label that is dominant (highly probable); the remaining labels all have low probabilities. More importantly, however, is the fact that all of the neighbors of $P_{INT}$ are, more or less, specified in the same manner. Thus, during the update process, the dominant label gets positive support (+1) from the <u>high</u> probability label at each pixel in its neighborhood and negative support (-1) from the remaining <u>low</u> probability labels at each pixel in its neighborhood. In general, the positive support for $P_{INT}$ greatly outweighs the negative support.

Now consider a pixel, say $P_{BORD}$, lying along the border of an object. The dominant label associated with such a pixel receives strong positive support (highly compatible and highly probable) from roughly <u>half</u> of its neighbors and, more importantly, it receives strong <u>negative</u> support (highly incompatible) from the remaining neighbors (which are also highly probable). Thus, the dominant label of $P_{BORD}$ tends to receive much less total support than the dominant label of $P_{INT}$. This situation is worse in the case of a thin object in which there may be few if any interior pixels to support its existence.

Let us turn our attention to a relative comparison of the different results. How do we account for the widely differing error rates associated with the various neighborhood formations? Notice that upon convergence the two results using limited neighborhoods, 4- and 5-adjacency, are better than those using larger neighborhoods of 8- and 9-adjacency. Again, when one considers that the errors are associated with thin objects that are embedded within large blobby objects, the explanation becomes clear. As shown in Figure IV.7, the 4-adjacent neighborhood allows the dominant label which represents a thin area (call it label A) to compete in equal numbers against an opposing label (call it label B) in a surrounding area. On the other hand, the 8-adjacent neighborhood is biased in favor of the competing label in the surround.

| | | |
|---|---|---|
| | B | |
| A | (A) | A |
| | B | |

4-adjacent neighborhood
can help preserve thin
lines

| | | |
|---|---|---|
| B | B | B |
| A | (A) | A |
| B | B | B |

8-adjacent neighborhood
favors the surrounding
label

Figure IV.7: The Impact of Neighborhood Geometry

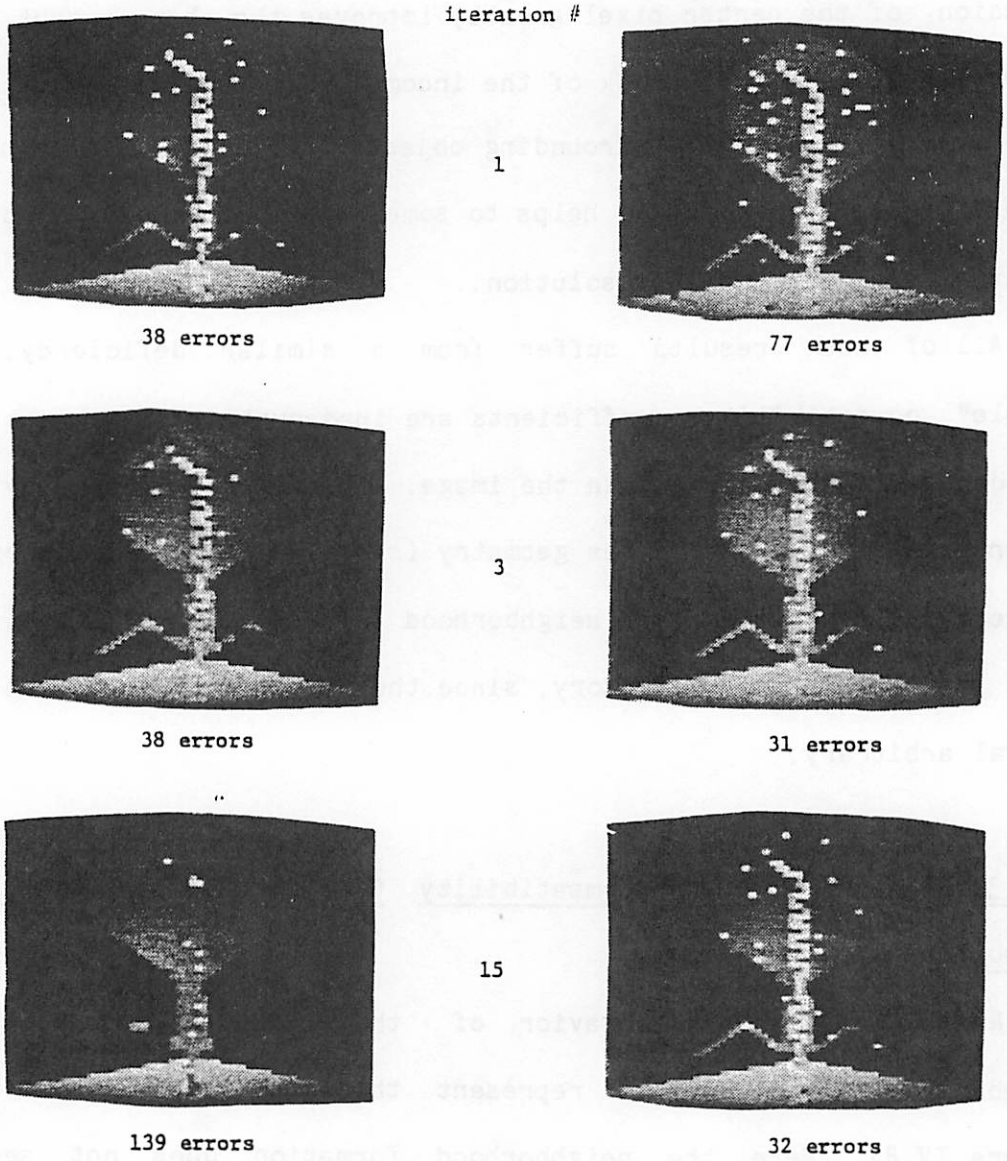
Next, consider the effect of the inclusion/exclusion of the center pixel in its own neighborhood. Clearly, the results are tremendously improved when it is included. The same argument applies since inclusion of the center pixel greatly improves the chances that a thin object can survive the attack of the incompatible label associated with the many pixels of the surrounding object. It is a form of "inertia" of resting probabilities and helps to some degree, but unfortunately it is not a very sound general solution.

All of these results suffer from a similar deficiency. The "simple" compatibility coefficients are inadequate to represent label dependencies that occur within the image. Therefore, the quality of a segmentation is driven by the geometry (shape) of objects with respect to the geometry of the pixel neighborhood that is defined for the RLP. This is clearly unsatisfactory, since the geometry of an object is in general arbitrary.

### IV.4.3 Relaxation Using Compatibility Coefficients as Conditional Probabilities

Next consider the behavior of the RLP when conditional probabilities are used to represent the compatibility coefficients (Figure IV.8). Here, the neighborhood formation does not seem to matter very much compared to the inclusion/exclusion of the center pixel in determining the overall error rate. When it is included, the RLP behaves in a desirable manner. The uncorrelated, mislabelled pixels are suppressed into the background and the finely structured areas are generally preserved.

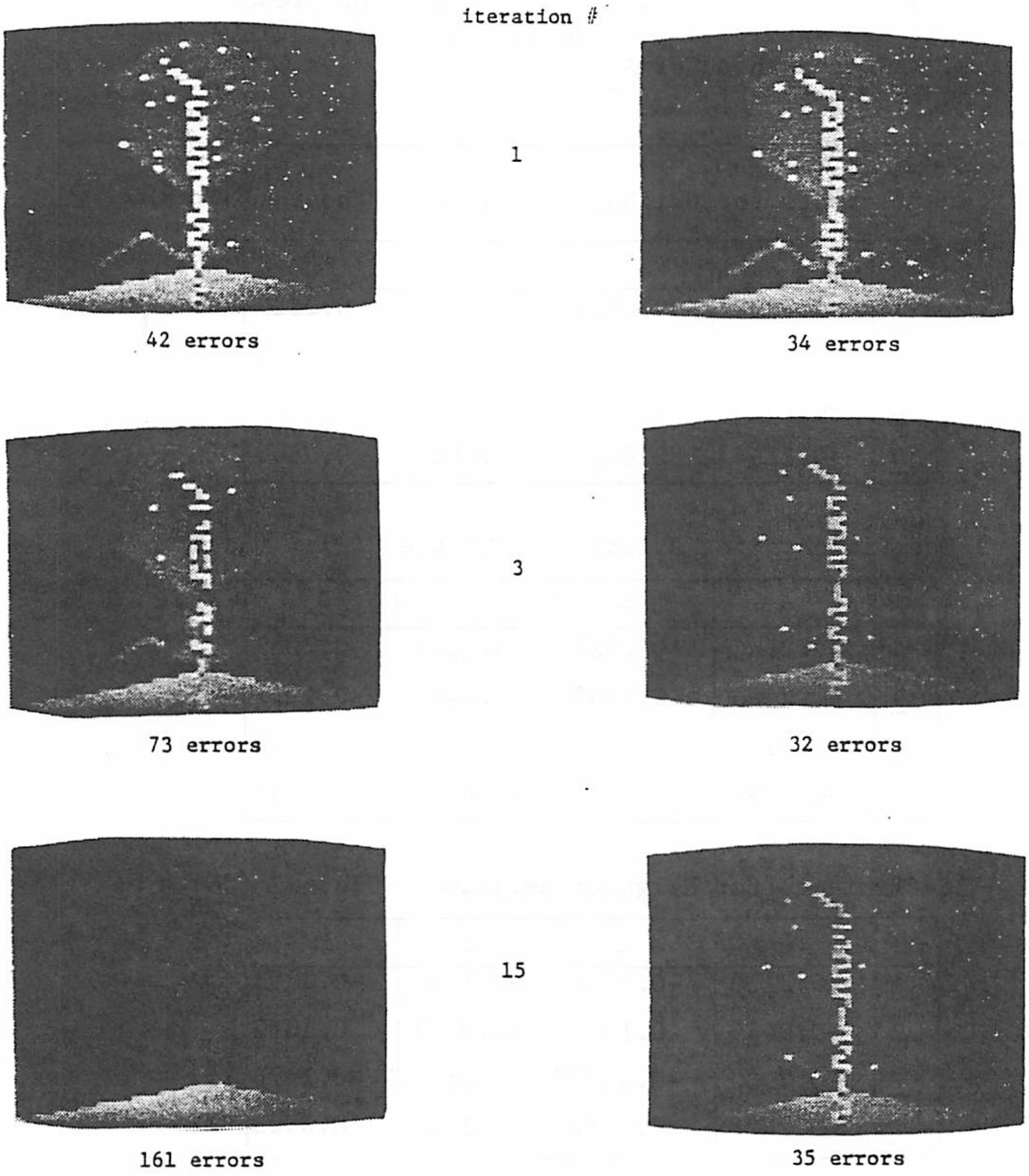
Let us carefully examine the compatibility coefficients for this image (Table IV.1). There are five arrays corresponding to the 5



(a) Results using 4-neighborhood

(b) Results using 5-neighborhood

<u>Figure IV.8</u> <u>Relaxation using conditional probabilities for compatibility coefficients.</u> Neighborhood sizes as shown. (Continued)



(c) Results using 8-neighborhood

(d) Results using )-neighborhood

Figure IV.8 Relaxation using conditional probabilities for compatibility coefficients. Neighborhood sizes as shown.

## TABLE IV.1

Compatibility coefficients for the image shown in Figure IV.4. Coefficients are a function of the conditional probability, $P(\alpha|\beta)$ and are specified between all pairs of pixels $A_i$ and $A_j$, where $A_j$ ε Neighborhood of $A_i$.

Coefficients for Neighbor Relation: "Above" (North)

| I \ J | A | B | C | D |
|---|---|---|---|---|
| A | 0.2440 | -0.2557 | -0.2349 | -0.2329 |
| B | -0.2459 | 0.1999 | -0.3793 | 0.1038 |
| C | -0.1553 | -0.3374 | 0.2095 | -0.2334 |
| D | -0.2194 | 0.0423 | -0.2665 | 0.0860 |

Coefficients for Relation: "To the Right" (East)

| I \ J | A | B | C | D |
|---|---|---|---|---|
| A | 0.2367 | -0.2522 | -0.2220 | -0.2289 |
| B | -0.2466 | 0.1895 | -0.3469 | 0.1111 |
| C | -0.2157 | -0.3534 | 0.2496 | -0.2143 |
| D | -0.2294 | 0.0471 | -0.2287 | 0.0735 |

Coefficients for Neighbor Relation: "Below" (South)

| I \ J | A | B | C | D |
|---|---|---|---|---|
| A | 0.2120 | -0.2552 | -0.1536 | -0.2293 |
| B | -0.2649 | 0.1989 | -0.3263 | 0.1055 |
| C | -0.2332 | -0.3682 | 0.2682 | -0.2476 |
| D | -0.2428 | 0.0416 | -0.2144 | 0.0903 |

Coefficients for Neighbor Relation: "To the Left" (West)

| I \ J | A | B | C | D |
|---|---|---|---|---|
| A | 0.2298 | -0.2454 | -0.2122 | -0.2278 |
| B | -0.2510 | 0.1899 | -0.3503 | 0.1180 |
| C | -0.2185 | -0.3437 | 0.2499 | -0.2245 |
| D | -0.2272 | 0.0447 | -0.2101 | 0.0738 |

Coefficients for Neighbor Relation: "Center"

| I \ J | A | B | C | D |
|---|---|---|---|---|
| A | 1.0000 | -1.0000 | -1.0000 | -1.0000 |
| B | -1.0000 | 1.0000 | -1.0000 | -1.0000 |
| C | -1.0000 | -1.0000 | 1.0000 | -1.0000 |
| D | -1.0000 | -1.0000 | -1.0000 | 1.0000 |

neighbor relations: "above," "to the right," "below," "to the left," and "center." Each is a 4x4 array corresponding to the 4 labels (i.e., the 4 cluster centers). For example, the compatibility between label B at the center pixel and label C at the pixel below (e.g., south) is -.3263.

Notice first that the on-diagonal entries of the arrays ($\alpha$ given $\alpha$) are all positive and the off-diagonal entries are generally negative. This is expected since in large blobby objects such as $O_1$, $O_2$, and $O_3$ the dominant label of the center pixel is most likely the same as the dominant label of the neighboring pixels. It is for this reason that the compatibility of label D given label D is the least positive on-diagonal entry; that is to say, $O_4$, which is represented by label D, has very few interior points and therefore label D given label D is a relatively infrequent event.

Because objects are oftentimes blobby, one might be tempted to use the "simple" compatibility coefficients -- they are an extreme example of the on-diagonal, off-diagonal (positive, negative) dichotomy. However, upon careful inspection of the tables one finds some important exceptions to this observation. Consider the compatibilities between label B ($O_2$) and label D ($O_4$). In all orientations the tables show a positive compatibility between these two labels which is the largest off-diagonal entry.

The compatibility between label D and label B is also positively compatible. It is because of these statistical relations that this version of the RLP does not suppress the thin object, $O_4$, into the background object $O_2$. This also explains the persistence of the 1-pixel "regions" -- labelled D -- inside of $R_2$; the compatibilities tend to support label D given label B wherever they co-occur. In contrast to this behavior, notice that the 1-pixel regions initially associated with the labelling of $O_1$ are (correctly) suppressed after 1 or 2 iterations. This is because no label other than A itself is positively compatible with label A, and therefore the mislabelled pixels are unsupported.

Let us now consider directionality information contained in the coefficients. Generally, the objects in this image do not display any strong directional dependency. However, the compatibilities do reflect a slight directional relationship between label A and label C. Notice that $O_1$ (label A), the upper background, is above $O_3$ (label C), the lower background. Thus, when C is associated with the neighbor "below," the compatibility between A and C (-.1536) is much less negative than for any other neighbor relation between A and C (-.2349, -.2220, -.2122). Similarly, when A is associated with the neighbor "above," the compatibility between C and A (-.1553)is much less negative than for any other neighbor relation between C and A (-.2157, -.2332, -.2185).

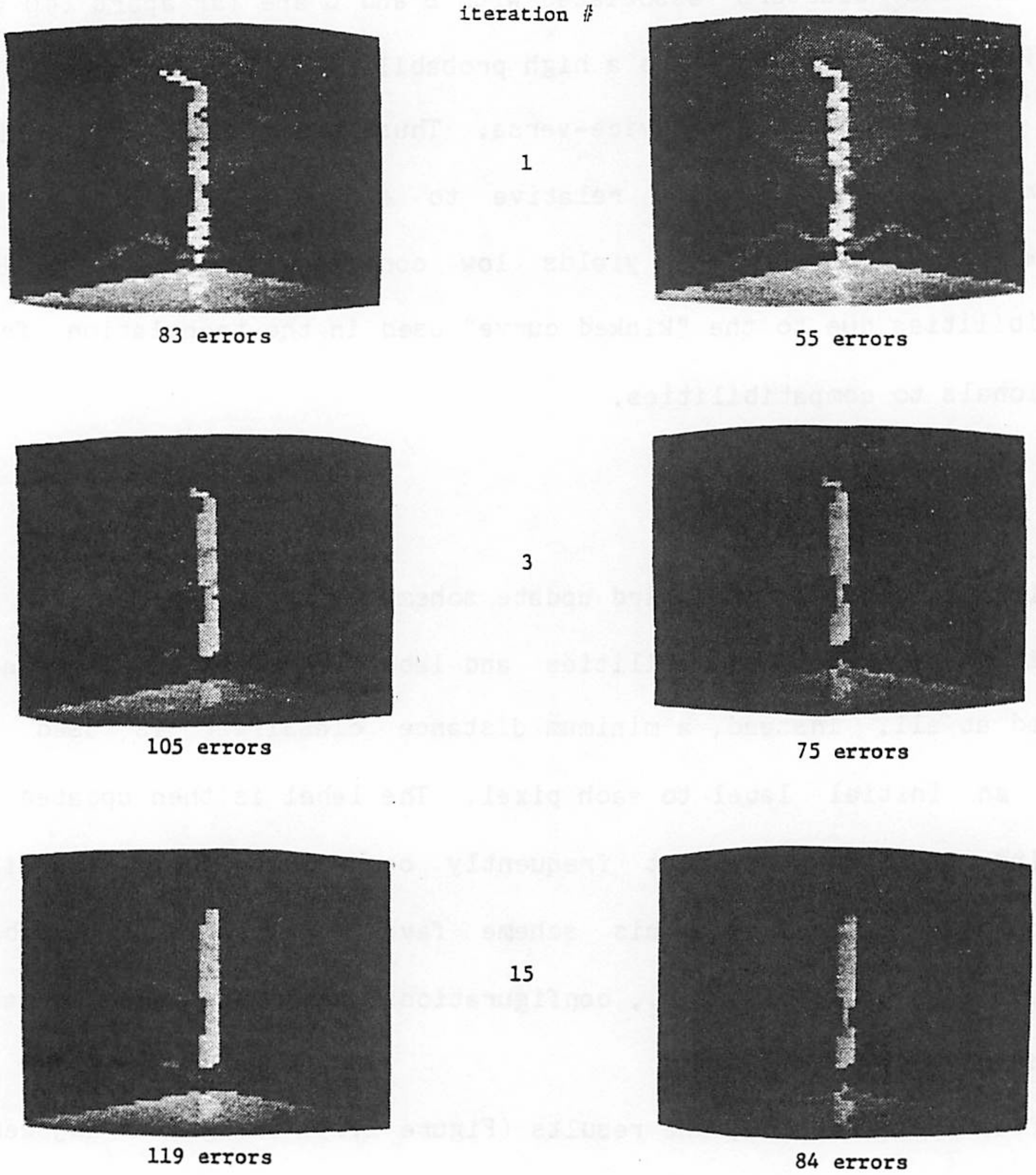
Finally, notice the relationship between labels B and C (or C and B). In all directions, this relationship is the most negative. This is the result of two effects. First, the objects that correspond to B and C, namely $O_2$ and $O_3$, have no common boundary and thus there is no significant spatial dependency between these labels.* Moreover, the means of the clusters associated with B and C are far apart (40 and 10). Therefore whenever B has a high probability (i.e., inside $O_2$), C has a low probability, and vice-versa. Thus, their joint probability (approximately .02) is low relative to either of their priors (approximately .2). This yields low conditionals and very low compatibilities due to the "kinked curve" used in the translation from conditionals to compatibilities.

### IV.4.4 Plurality Relaxation

Finally, consider the third update scheme -- the plurality rule -- in which the label probabilities and label compatibilities are not employed at all. Instead, a minimum distance classifier is used to assign an initial label to each pixel. The label is then updated by replacing it with the most frequently occurring label in its neighborhood. Therefore, this scheme favors geometrically stable configurations of labels, e.g., configurations that are rounded and contain interior points.

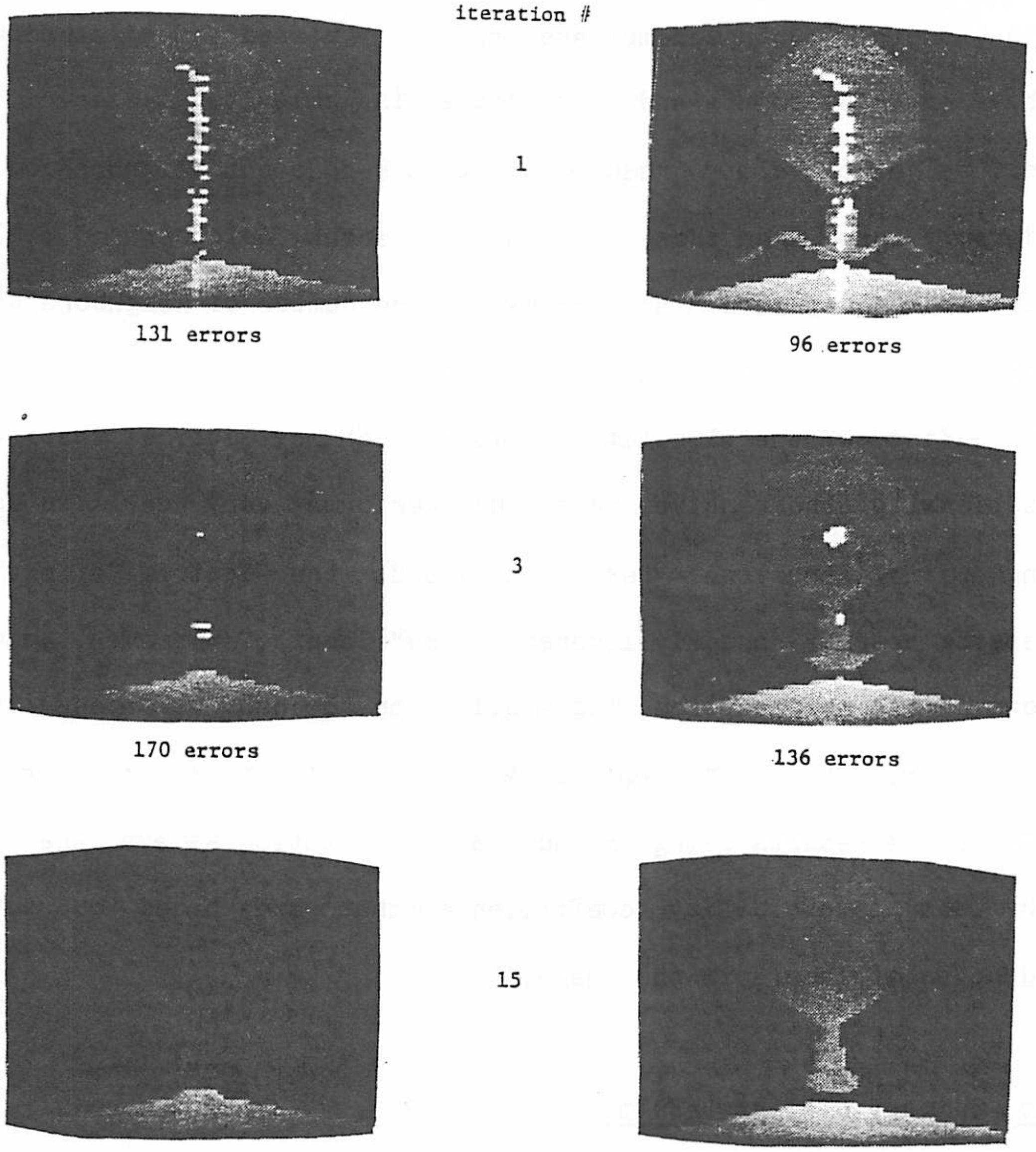
After 15 iterations, the results (Figure IV.9) using a 4-adjacent neighborhood are not much worse than the results obtained via the probabilistic relaxation update using simple compatibility coefficients. This is not surprising since neither technique

*Although, one should recall that all pixels have some finite non-zero probability of both B and C. However, since the corresponding objects O and O do not touch, it is very unlikely that the joint probability of B and C will be high in any neighborhood.



(a) Results using 4-neighborhood (b) Results using 5-neighborhood

Figure IV.9 Plurality update across indicated neighborhoods. (Continued)



131 errors

96 errors

170 errors

136 errors

172 errors

156 errors

(c) Results using 8-neighborhood

(d) Results using 9-neighborhood

Figure IV.9 Plurality update across indicated neighborhoods.

incorporates information that is based on structural dependencies between labels. Both schemes are implicitly biased toward structures that have interior points and thus neither is able to preserve thin regions. The plurality results using an 8-adjacent neighborhood are considerably worse than those with the 4-adjacent neighborhood. This is also to be expected since increasing the number of neighbors works against maintaining the fragile structures that we have been examining.

In defense of the plurality relaxation scheme, notice that this computationally inexpensive technique performs very well in areas lacking spatial structure. Here, it yields the desired effect of suppressing sparse, randomly located "noise" labels. Moreover, as will be shown in the next section, its application to natural scenes that mostly contain "blobby" regions yields results that are remarkably similar to the results using probabilistic relaxation -- even when the latter uses compatibility coefficients that are based on spatial dependencies of labels in the image.

### IV.4.5 Summary of Test Results

Before leaving this set of images, it is worth commenting on the error rates (see Tables IV.2 and IV.3). According to Table IV.2 the "simple" relaxation scheme gives the best results in the short run. However, the converged results (Table IV.3) show a significant degradation of performance. On the other hand, relaxation with conditional probabilities has only slightly worse peak results than the simple scheme and importantly, it does not degrade at all over time.

TABLE IV.2 SUMMARY OF TEST-IMAGE RESULTS

Minimum number of errors tabulated for each Relaxation Method and neighborhood size. Entries indicate total number of mislabelled pixels at the given iteration.

| Neighborhood<br>Method | 4 | 5 | 8 | 9 |
|---|---|---|---|---|
| Simple (+1,-1) | 28<br>(Iter 3) | 21<br>(Iter 3) | 26<br>(Iter 1) | 32<br>(Iter 1) |
| Conditional Probabilities | 38<br>(Iter 1) | 31<br>(Iter 3) | 42<br>(Iter 1) | 32<br>(Iter 3) |
| Plurality | 83<br>(Iter 1) | 55<br>(Iter 1) | 131<br>(Iter 1) | 96<br>(Iter 1) |

(233 errors initially)

TABLE IV.3 SUMMARY OF TEST-IMAGE RESULTS

Number of errors at convergence tabulated for each relaxation method and neighborhood size. In each case, the relaxation process was run until all pixels had a single label with probability 1 (approximately 15 iterations). Entries indicate the total number of mislabelled pixels. The number of errors in parentheses was obtained after 1 and 2 pixel "regions" were suppressed. No other cases produced reductions in error rates by such processing.

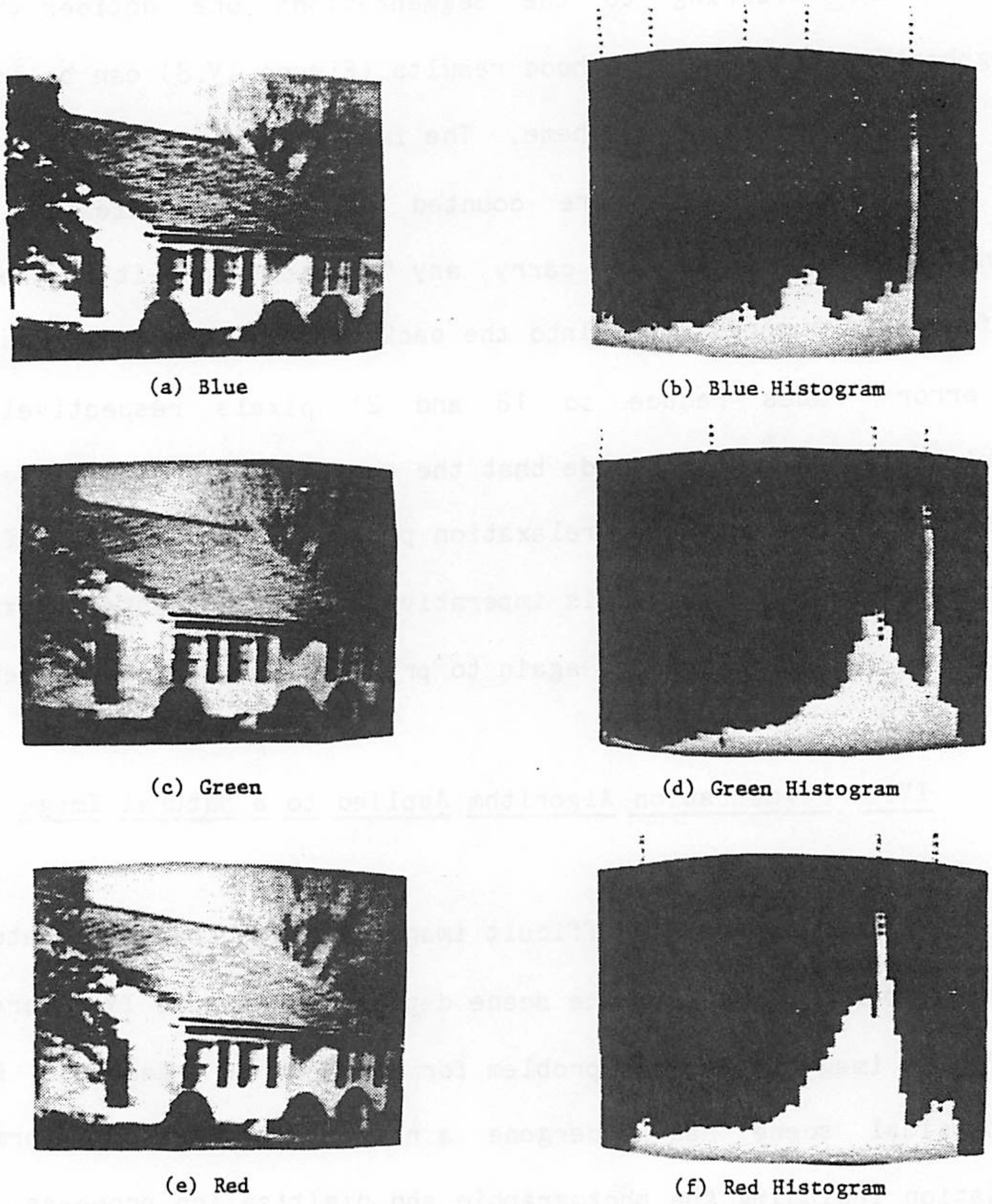
| Neighborhood<br>Method | 4 | 5 | 8 | 9 |
|---|---|---|---|---|
| • Simple (+1,-1) | 94 | 59 | 130 | 95 |
| Conditional Probabilities | 139<br>(149) | 32<br>(18) | 161 | 35<br>(21) |
| Plurality | 119 | 84 | 172 | 156 |

(233 errors initially)

Moreover, by referring to the segmentations one notices that the 5-neighborhood and 9-neighborhood results (Figure IV.8) can be improved by a very simple clean-up scheme. The images show a large number of 1 and 2 pixel "regions" that are counted as errors. Clearly, these regions are too small to carry any "meaning", and it is therefore justifiable to suppress them into the background. When this is done the error rates reduce to 18 and 21 pixels respectively, or approximately .4%. We conclude that the conditional probabilities are necessary to prevent the relaxation process from destroying fragile structures. In addition, it is imperative that the center pixel be included as its own neighbor, again to preserve fragile structures.

## IV.5 Segmentation Algorithm Applied to a Natural Image

We now turn to a more difficult image domain, that of naturally occurring outdoor scenes. The scene depicted in Figure IV.10 presents a difficult image processing problem for a number of reasons. First, the physical scene has undergone a number of stages of information degradation including the photographic and digitization processs, and a spatial averaging (blurring) process to reduce the amount of data to managable levels. The effect of these processes (refer to Section I.1) is to introduce noise, blur edges, and to create hybrid pixel values -- mixed pixels -- which are not easily classifiable. Moreover, the image displays inherent visual complexities such as irregular texturing, object occlusion, and irregular changes in gradients. Finally, the



(a) Blue

(b) Blue Histogram

(c) Green

(d) Green Histogram

(e) Red

(f) Red Histogram

Figure IV.10 Blue, green, and red components of our outdoor scene. The indicated peaks were automatically detected.

image is complex because in 3-color space there is a large, unknown number of object classes to be discriminated, most of which overlap to varying degrees.

In the next section we will show the results of applying the algorithm using probabilistic relaxation and conditional probabilities to our example outdoor scene. Later, the algorithm will be expanded to include feature transformations of an opponent-color system that improves color contrast. In addition, multidimensional clustering to increase the sensitivity of the segmentation will be explored.

### IV.5.1 The Data

The natural outdoor image used in these experiments consists of a 512x512 array of pixels in which each pixel is represented as a triple of six-bit numbers. The components of a pixel correspond to its light intensity when scanned through red, gree, and blue (RGB) filters. The original data has been transformed by independently blurring each component via a 2x2 spatial averaging process, yielding a resolution of 256x256 pixels. The data reduction steps were performed so that the resulting image would contain a manageable amount of information that could be processed in a reasonable time period on a PDP-15 minicomputer.

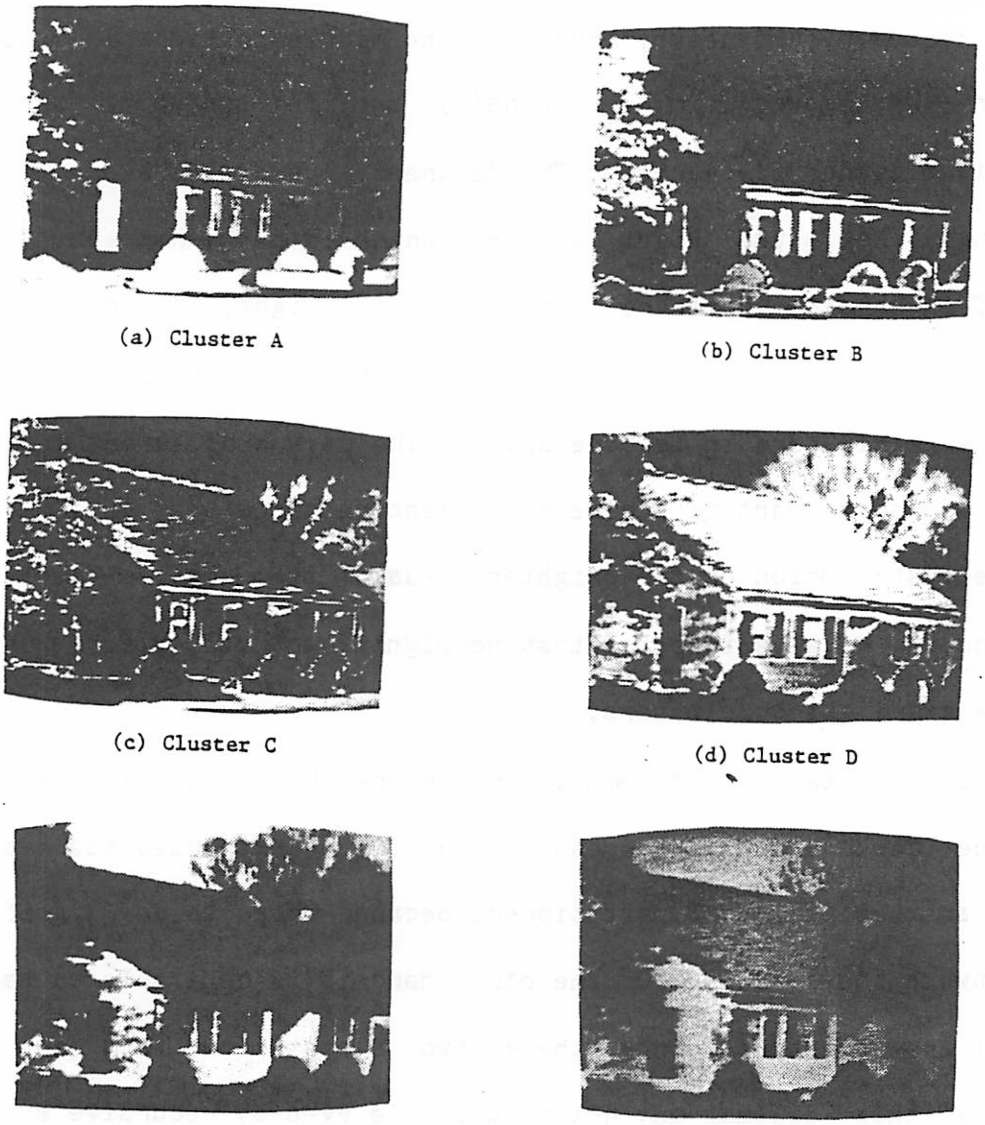
### IV.5.2 Initial Labelling

The peaks for the RGB distributions were detected by the process discussed in Section IV.3.1 and are marked on the histograms shown in

Figure IV.10. The following segmentation experiments were performed using the blue component of the image, since its histogram had the highest number (5) of detectable peaks. Figure IV.11(a-e) shows the initial labelling of each pixel with respect to the 5 peaks in the blue component histogram. As before, for each cluster, the probability of a pixel is displayed as a gray level with black representing low probability and white representing high probability. In addition, Figure IV.11(f) shows the highest probability label at each pixel, in effect a minimum distance classification, with each of 5 distinct gray levels representing a cluster label. Note that the 5 clusters correspond to distinct gray level ranges in the blue intensity image in ascending brightness. Thus, cluster A represents the darkest areas of the image (e.g. shadowed bushes) and cluster E represents the brightest areas (e.g. sky and sunlit house walls).

How can the initial segmentation be evaluated? Since there is no ground truth data available with which to generate an error rate, the evaluation must be subjective. First, notice that the roof of the house and the tree crown on the right are overmerged. Cluster D, which is relatively wide, apparently contains the distribution of both of these objects. Since they happen to lie adjacent to each other in the image, they receive the same region label and appear as a single region in the segmentation.

This situation is curious since there seems to be an edge (or part of an edge) between the two objects in the original image. The explanation is that the roof is actually a slowly varying piecewise



(a) Cluster A

(b) Cluster B

(c) Cluster C

(d) Cluster D

(e) Cluster E

(f) Minimum distance classification into 5 cluster types.

Figure IV.11 Initial pixel labelling based on the 5 peaks in the blue-component histogram. The highest probability label at each pixel is shown in (f).

linear gradient. That is, the upper left portion is dark (from the shadow created by the nearby tree) and the middle portion is slightly brighter (unshadowed). The intensity profile then drops until the lower right corner is reached. The darkness there is again due to shadowing. Now, the right hand tree which is dark, happens to touch the roof in an area where the latter is still bright, thus creating an edge. The difficulty of segmenting these two objects is a problem of undetectable clusters in feature space. The dark roof areas and the dark tree form part of a cluster in feature space that is indistinct from the distribution of the brighter areas of the roof. The latter have enough of a variance so that no significant valley forms between the dark and light subclusters.

This is certainly a dilemma. On the one hand, it is desirable that the dark and light areas of the roof be extracted via a single cluster so that it is not partitioned, because there is no significant edge between these areas. On the other hand it is desirable to segment the roof from the tree since these two ojbects do form an edge. Resolving this dilemma may not be possible even by recursive analysis of the overmerged roof/tree region, since a histogram localized to this region may still appear unimodal. It would require a model of the spatial changes in feature values. Recent work by Haralick [HAR78] may prove useful here.

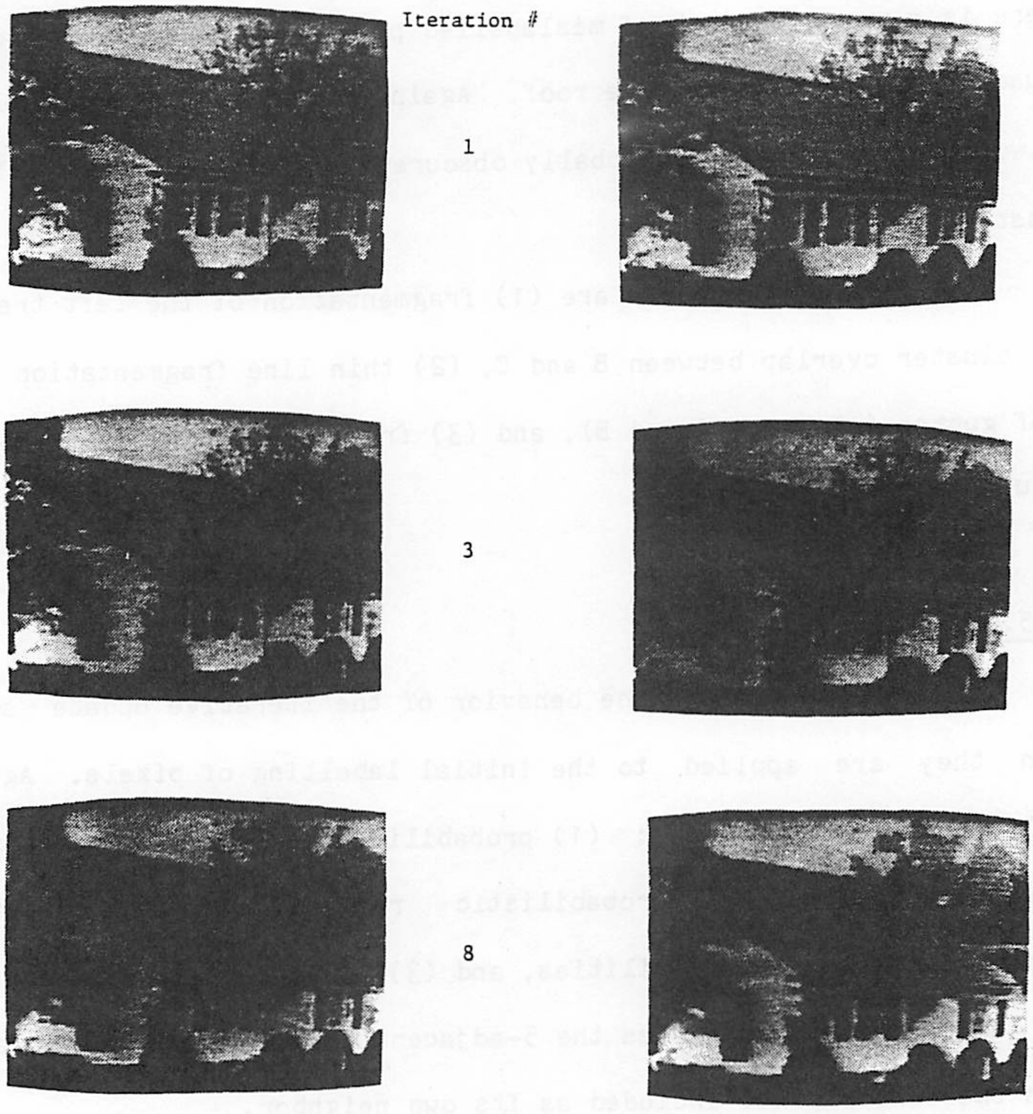
Consider some other problem areas in the initial segmentation. Notice for instance, the appearance of fragmentation in the roof. The left side of the roof contains many mislabelled pixels which are scattered and result from the overlap of clusters C and D. Further, there is a connected set of mislabelled pixels that extend from the house roof into the garage roof. Again, the edge that exists locally between the two roofs is globally obscured in the form of overlapping clusters.

Other areas of concern are (1) fragmentation of the left tree due to cluster overlap between B and C, (2) thin line fragmentation of the roof gutter (clusters A and B), and (3) fragmentation of the shutters (clusters A and B).

### IV.5.3 Iterative Update

Let us next consider the behavior of the iterative update schemes when they are applied to the initial labelling of pixels. Again we apply the three techniques: (1) probabilistic relaxation with simple compatibilities, (2) probabilistic relaxation with conditional probabilities for compatibilities, and (3) plurality relaxation. All of these are applied across the 5-adjacent neighborhood configuration, with the center pixel included as its own neighbor.

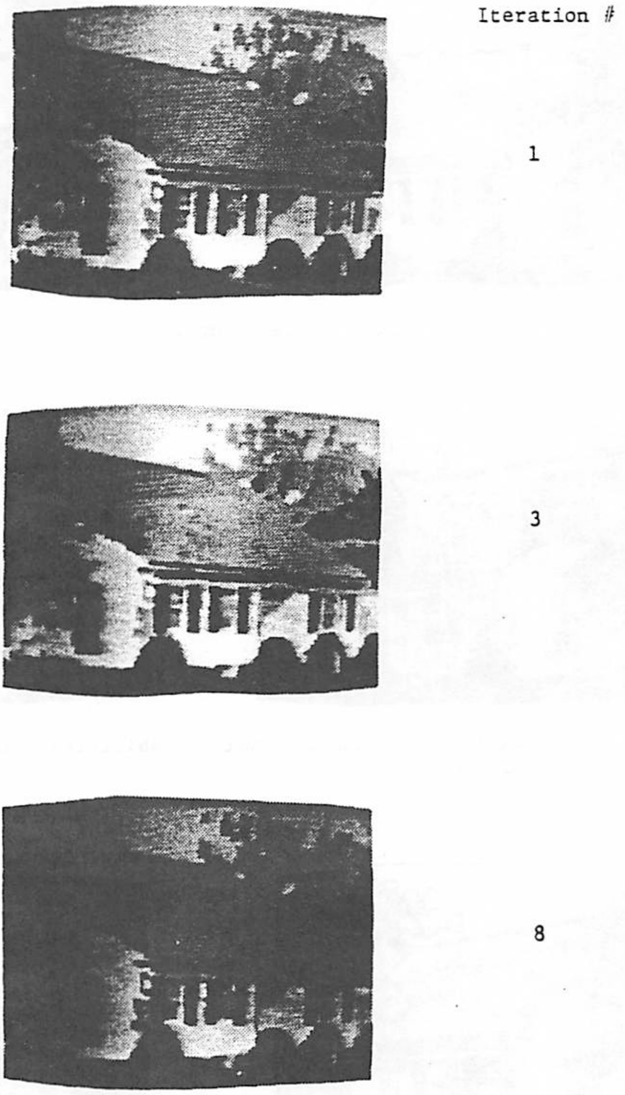
Figure IV.12 shows the results after 1, 3, and 8 iterations of each scheme. In addition, Figure IV.13 shows the results displayed as edges between regions. One is struck by the similarity of all of these results. The only significant differences are in the roof gutter, the left tree, and around the front windows. The probabilistic relaxation technique using simple compatibilities gives th: cleanest looking result in both areas. The other two schemes ge1erate many small



(a) Compatibilities as Conditional Probabilities

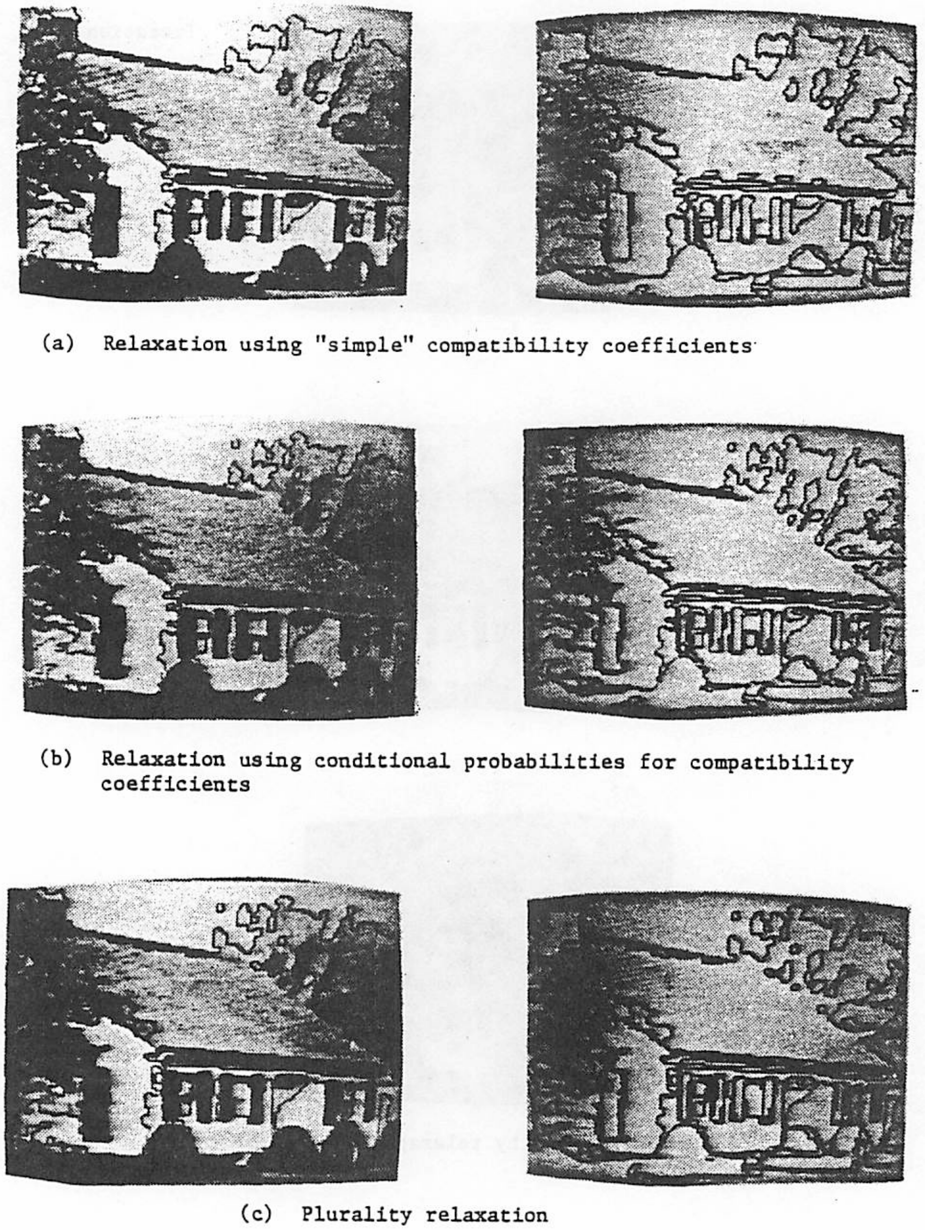
(b) Simple (+1,-1) Compatibilities

Figure IV.12 Three variations of the relaxation scheme applied to a 5-neighborhood. (Continued)



(c) Plurality relaxation

Figure IV.12 Three variations of the relaxation scheme applied to a 5-neighborhood.



(a) Relaxation using "simple" compatibility coefficients

(b) Relaxation using conditional probabilities for compatibility coefficients

(c) Plurality relaxation

Figure IV.13 Segmentations as Edges.

regions in these areas and it is not clear what to do with them: they are too small (i.e., <5 pixels) to be "meaningful" and they are too densely packed to justify simply suppressing them into the surrounding regions.

Apparently, the use of conditional probabilities is preserving too much detail. This leaves another dilemma, because the preservation of detail is clearly desirable in some areas. However, this should not be considered to be a fault of the (conditional probability) compatibility coefficients, for they are simply doing their job. Rather, in the case of the tree and gutter areas, the fault lies partially with the global clustering process, which fragmented the objects in the first place and partially with the data itself, which is particularly noisy in those areas.

It is interesting to speculate on how to recover the left tree as a single region. The segmentation has left a group of small regions that are densely packed and mostly of cluster types B and C. One might consider the use of a spatial adjacency matrix [HAR76] which would measure the frequency of pairs of labels over the set of region pairs [NAG77]. This NxN matrix (where N is the number of labels or clusters) would show, for instance, a high off-diagonal entry that would indicate the frequency of label B adjacent to label C. If large enough, this entry could be interpreted as a "cluster" and the corresponding region pairs could be relabelled as belonging to that cluster. Thus, all of the small regions with label B or C would be super-labelled into a new category. Notice that this is a kind of texture analysis in which a micro-texture pattern [RIS76] consisting of two elements is detected and labelled.

## IV.6 Multidimensional Feature Analysis

Let us now turn to two methods of augmenting the segmentation process developed to this point. First, we will dicuss the use of color spaces other than RGB, in which color information is enhanced. The enhancement often leads to better discrimination of objects in the scene. Second, we will consider the use of higher dimensional feature spaces in which it is possible to obtain finer cluster discrimination.

### IV.6.1 Opponent Color Features

The segmentation techniques depend on the measurement of some feature(s) of the image pixels, possibly including the raw sensory data originally used to represent the scene. For color images, the usual measurements are the red, green, and blue components (RGB) of the intensity level at each pixel in the scene. From this information, a variety of other representations, such as normalized RGB, or hue, saturation, and intensity (HSI), may be derived [TEN74,RIS77]. However, because many of these transformations are nonlinear, they give rise to distributions with unavoidable singularities [KEN76]. The presence of these singularities may severely complicate analysis of the resulting histogram. In order to avoid these difficulties, it has been suggested that analysis be restricted to linear transformations of RGB, such as the YIQ [BIN57] representation used in the television industry.

More recently, Sloan and Bajcsy [SLO75] have argued for the use of an opponent-color representation which has been proposed as underlying the color mechanisms in human vision [COR70]. Simply stated, the effect of this transformation is to parameterize the RGB color data into an equivalent set of features which have pairs of complementary colors at the extremes of their scales; for example, a feature whose opponents are blue and yellow would provide information on the relative amounts of blue and yellow present. The "zero" point in the scale, where equal amounts of each hue are present, is white.

For a precise formulation of opponent color spaces one can turn to the work of several researchers in colorimetry. See Pratt [PRA78] for an excellent review of systems such as (U,V,W), (U*,V*,W*), and (L,a,b). Unfortunately, as Pratt points out, there is no clear mechanism for selecting one system over another. We have selected the opponent system (U*,V*,W*), an extension of (U,V,W) for the current work. The opponent axes may be interpreted as follows:

U* ≡ red vs. green

V* ≡ yellow vs. blue

W* ≡ white vs. black

The (U*,V*,W*) system has the important property that chromaticity and brightness changes are more or less uniformly noticeable [PRA78]. Thus, in this space, our perception of color differences in an image that is displayed on a color monitor will be roughly uniformly proportional to the digital representation of those differences. This property which should aid in the subjective evaluation of image segmentation results, is absent in (R,G,B) space.

The computation of (U*,V*,W*) from (R,G,B) is defined by:

$$U^* = (217.358*R - 130.319*G - 24.558*B + k_1)*M_1$$

$$V^* = (-35.461*R - 79.703*G + 90.508*B + k_2)*M_2$$

$$W^* = (60.594*R + 80.160*G + 39.265*B)*M_3$$

where $k_1$ and $k_2$ are selected to insure that their respective components are strictly in a positive range and

$M_1$,$M_2$, and $M_3$ are selected so as to scale the components to n bits; n = 6 in our experiments.

We should mention that we have multiplied each term of V* by -1 before scaling. This changes V* so that it effectively measures blue-yellow. This allows a blue object to appear closer to a natural color when displayed on an RGB color monitor.

Figure IV.14 shows the results of transforming the R,G,B components into U*,V*,W*. Notice that many of the objects appear to be much more strongly contrasted in U*,V*,W* than they were in RGB, (e.g. the left-hand tree and the bushes). Notice also that the U* histogram has much clearer peaks than any of the RGB components.

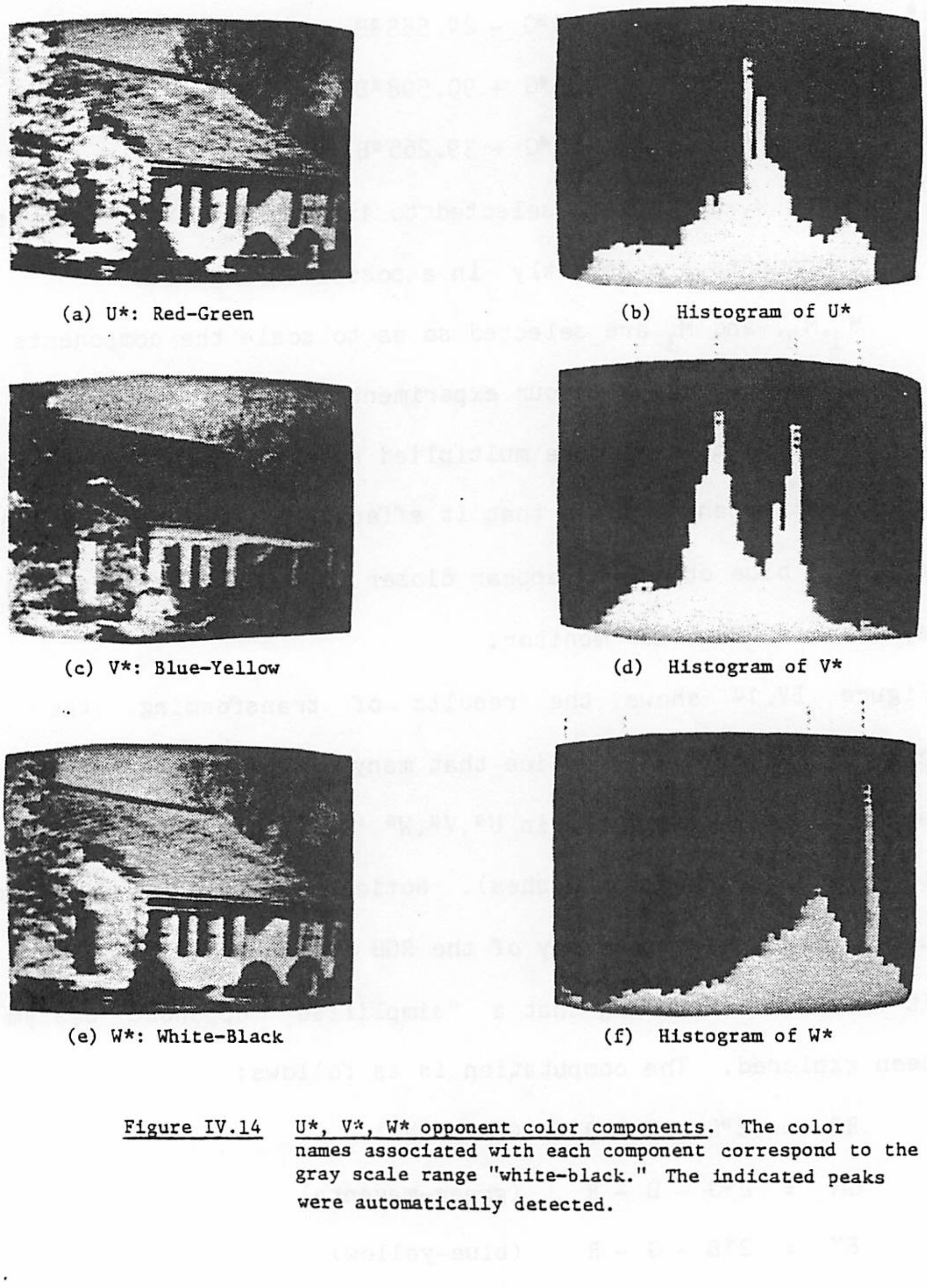
It is worth mentioning that a "simplified" opponent system has also been explored. The computation is as follows:

$$RC = 2*R - G - B \quad \text{(red-cyan)}$$

$$GM = 2*G - B - R \quad \text{(green-magenta)}$$

$$BY = 2*B - G - R \quad \text{(blue-yellow)}$$

The computation is simpler for the obvious reason that the coefficients are all integer and thus no scaling is necessary (except a linear shift to insure positive values). The result of subjective evaluation is that this system yields images that are contrast



(a) U*: Red-Green

(b) Histogram of U*

(c) V*: Blue-Yellow

(d) Histogram of V*

(e) W*: White-Black

(f) Histogram of W*

Figure IV.14 U*, V*, W* opponent color components. The color names associated with each component correspond to the gray scale range "white-black." The indicated peaks were automatically detected.

enhanced. Moreover, the histograms appear to have greater cluster separation than in RGB, thus allowing improved cluster detection.

### IV.6.2 Two-Dimensional Peak Detection

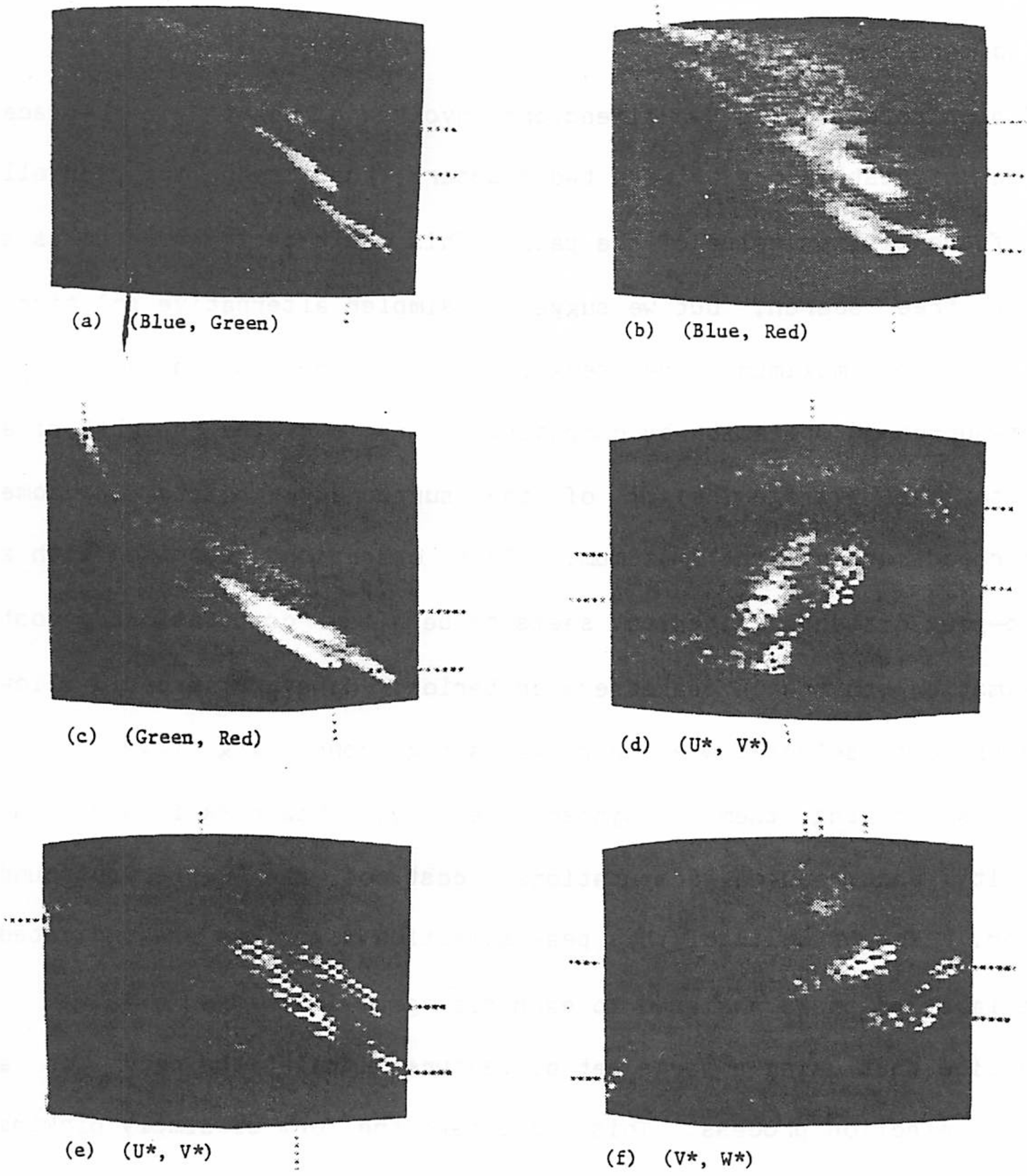
Looking again at Figure IV.14, one notices that there are objects that are clearly distinguishable in one feature that are not distinguishable in another. For instance, there is clearly an edge in U* between the right tree and the roof, while these same objects are much less distinguishable in V*. On the other hand, in V* the left bushes (shadowed) are clearly distinct from the right bushes (unshadowed), but they have about the same apparent intensity level in U*. Of course this is not necessarily a positive characteristic of V* because one would like the bushes to be labelled the same.

These observations lead one to conclude that classification of pixels into clusters would be improved if more features were used. This technique was used by Ohlander [OHL76]. In his algorithm, regions were liable to be recursively segmented if they were multimodal in <u>any</u> feature from among a set of nine feature histograms. Another approach to multi-feature analysis is to compute higher dimensional feature spaces. In this manner, not only can the segmentation algorithm exploit distinctions in many features simultaneously, but in addition subtle feature dependencies often appear which may yield cluster centers that are better representatives of the underlying data. Moreover, all of this can be accomplished in one step instead of many recursive steps.

One problem with multi-dimensional feature spaces is that clustering becomes a very non-intuitive, abstract process. This means that it is difficult to evaluate whether the clustering process is behaving in a desirable manner. For this reason, we have limited the application of the segmentation algorithm to 1-D and 2-D histograms since they can readily be displayed and understood.

Consider the set of 2-D histograms shown in Figure IV.15. In each case the axes are labelled with some pair of color feature components from (R,G,B) and (U,V,W). The frequency of values of a feature pair is displayed as a gray level (white = very high frequency). Notice that the RG, GB, and RG histograms all have the appearance of being very highly correlated. This is confirmed by looking back at the corresponding images (Figure IV.10), which all have a similar visual appearance. The U*V*, V*W*, and U*W* histograms have different characteristics. Here we see a wide spread of off-diagonal clusters. The detection of these additional clusters leads to a clear computational savings by reducing the number of recursive region decomposition steps necessary to accurately locate the underlying regions.

The peak selection algorithm explored in Section IV.3.1 can easily be modified to handle 2-D feature clusters. Recall that the 1-D peak selection algorithm used the minima between each maximum to determine peakedness and that this was an important criterion for peak acceptability. Detection of local maxima in two dimensions is straightforward, but detection of the corresponding valleys between



(a) (Blue, Green)

(b) (Blue, Red)

(c) (Green, Red)

(d) (U*, V*)

(e) (U*, V*)

(f) (V*, W*)

Figure IV.15 Two-dimensional histograms obtained from (blue,green, red) and (U*,V*,W*). The indicated peaks were automatically detected.

peaks is more complex.

Minima selection in two dimensions involves a search in 2-space for the highest ridge between two clusters, that is the max over all paths of the minimum value of the path. This could be implemented as a parallel tree search, but we suggest a simpler alternative solution. For each local maximum, the peakedness will be estimated via a "center-surround" operation by computing the ratio of the "height" of a local to the average height of the surrounding points in some neighborhood around the maximum. This operation, combined with a peak-to-peak distance criterion, seems to be a low computational cost approximation to the 1D peakedness criterion. However, it could allow two peaks to be selected when there was a high connecting ridge between the peaks, making them one syntactic entity. This risk is worth the simplicity and reduced computational cost of the center-surround operator. The results of this peak selection algorithm are indicated by the labelled peaks assigned to each histogram in Figure IV.15.

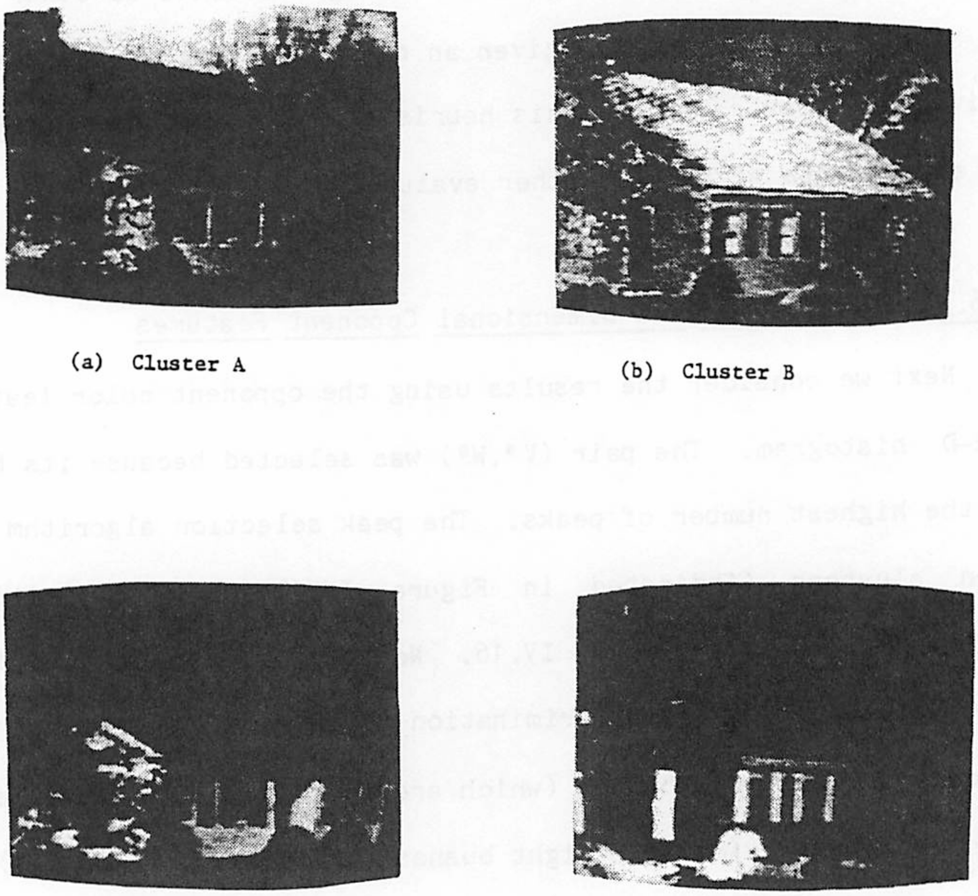
Notice that using a large set of features implies the need for a feature selection process. This might take the form of simply picking the histogram with the greatest number of peaks. Another possibility is to compute the entropy of the histogram. A high entropy value implies that the data in the histogram is widely spread. This could be interpreted as indicating greater numbers of clusters.

However, both of these measures could be improved by considering the "quality" of the peaks as well, where quality is a function of peakedness and separability. Thus, each peak could be rated by the product of its peakedness and its average distance to other clusters. The histogram could then be given an overall rating as the sum of the individual peak ratings. This heuristic has worked reasonably well in our experiments although further evaluation is required.

### IV.6.3 Results with Two-Dimensional Opponent Features

Next we consider the results using the opponent color features in a 2-D histogram. The pair (V*,W*) was selected because its histogram had the highest number of peaks. The peak selection algorithm selected seven clusters (indicated in Figure IV.15) and the initial pixel labelling is shown in Figure IV.16. Notice that there is a finer -- although not perfect -- discrimination of the roof from the right tree. In addition, the left bushes (which are in shadow) are labelled in a different cluster from the right bushes (which are more sunlit).

The results using relaxation are shown in Figure IV.17. The algorithm used probabilistic relaxation with conditional probabilities and was applied to a 4-adjacent neighborhood with the center pixel included. Compatibility coefficients were computed from a 2D (V*,W*) histogram. These results can be compared with those in Figure IV.12. There seems to be a general improvement in the quality of this segmentation over the results using the 1-dimensional histogram clusters obtained from the raw blue feature. Major components of the right tree appear as separate regions, the left tree is more or less in one piece, the roof gutter is not quite as fragmented, and in general all regions are much less noisy.



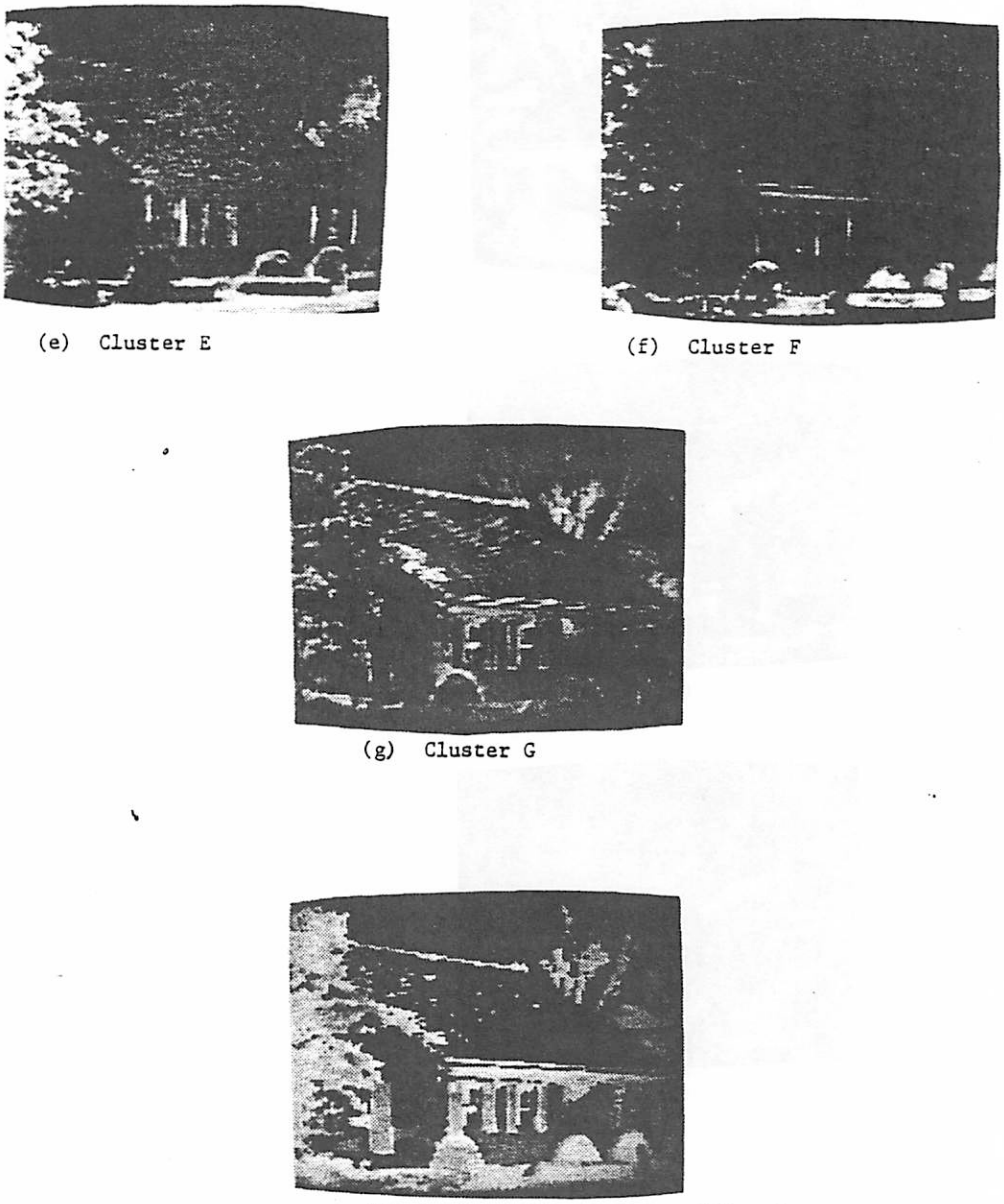
(a) Cluster A

(b) Cluster B

(c) Cluster C

(d) Cluster D

<u>Figure IV.16</u> <u>Initial pixel labelling based on the 7 peaks in the (V*,W*) histogram</u>. (Continued)



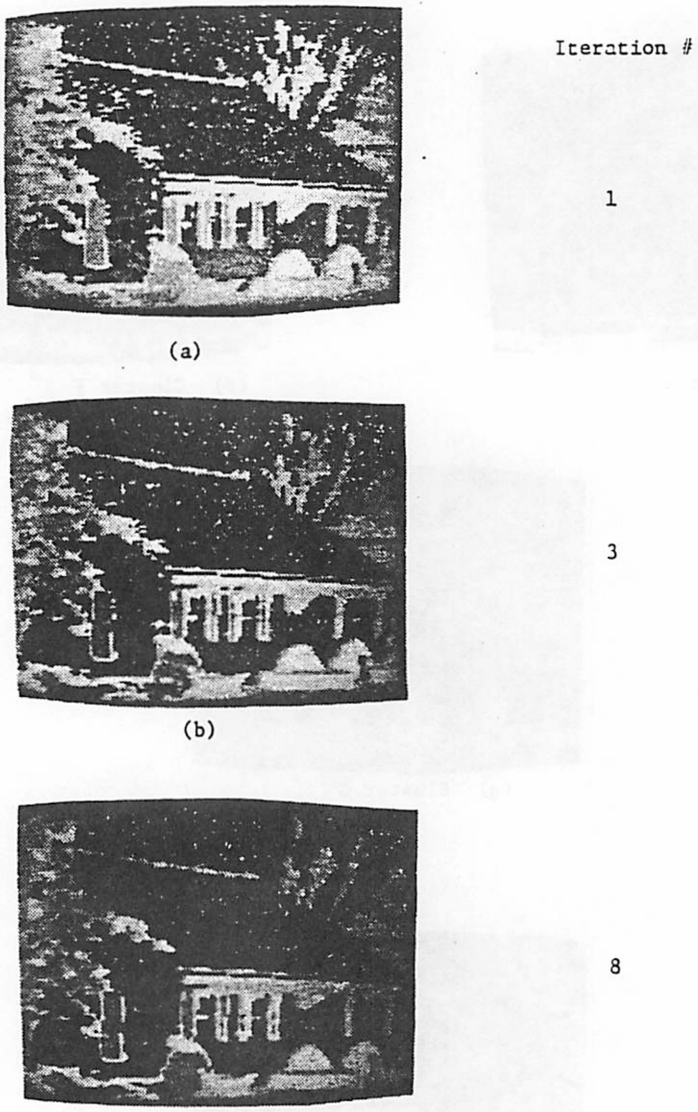
(e) Cluster E

(f) Cluster F

(g) Cluster G

(h) Maximum distance classification into 7 cluster types.

Figure IV.10 Initial Pixel Labelling.



(a)

(b)

(c)

<u>Figure IV.17</u> <u>Probabilistic relaxation applied to the initial labelling with 7 clusters from (V*,W*).</u> (5-neighborhood)

## IV.7 Conclusions

This chapter has covered a wide range of topics in image processing. Relaxation labelling processes were defined and their behavior was explored with applications to artificial and natural images. Compatibility coefficients for RLPs were explored and were shown to critically effect the performance of the algorithms. Finally, multidimensional color spaces were introduced and shown to improve the sensitivity and quality of the results.

Let us now make some recommendations and evaluations based on the work presented.

1. Post-processing via RLP of histogram-based pixel labelling clearly improves the <u>overall</u> quality of a segmentation. However, one must pay attention to the specific behavior of the RLP in certain areas of images. This can best be done with the help of test images that have been designed to highlight expected problems in image analysis, such as noise and unstable spatial structures. Other features such as gradients, blurring, and complex texturing should also be tested.
2. When appropriately specified, compatibility coefficients can help prevent the RLP from destroying fine detail in an image. This was clearly shown by comparative studies with image-dependent coefficients, image-independent coefficients, and no coefficients. The latter two experiments yielded much

worse results than the experiments with coefficients based on conditional frequencies of labels in an image.

3. During relaxation, the center pixel in a neighborhood should be allowed to contribute to the label update function as if it were a member of its own neighborhood. This allows spatially fragile structures to obtain more self-support and thus helps preserve fine image detail.

4. Plurality relaxation is useful for noise suppression but is damaging to image details. However, it is computationally much less expensive than other relaxation schemes, and therefore it may be of use in certain domains. In fact, its behavior in a complex natural scene domain did not appear to be much worse than the more complex probabilistic relaxation schemes.

5. Clusters that are hidden in one-dimensional histograms (due to overlapping distributions) may be revealed in multi-dimensional feature spaces. The extra clusters that are revealed may (a) lead to finer discrimination of image regions, and (b) reduce the number of overmerged regions, thereby reducing the need for recursive segmentation.

6. Opponent color spaces seem to enhance object boundaries and give clearer cluster separation in histograms.

# CHAPTER V

## FURTHER CASE STUDIES IN GLOBAL SEGMENTATION PROBLEMS

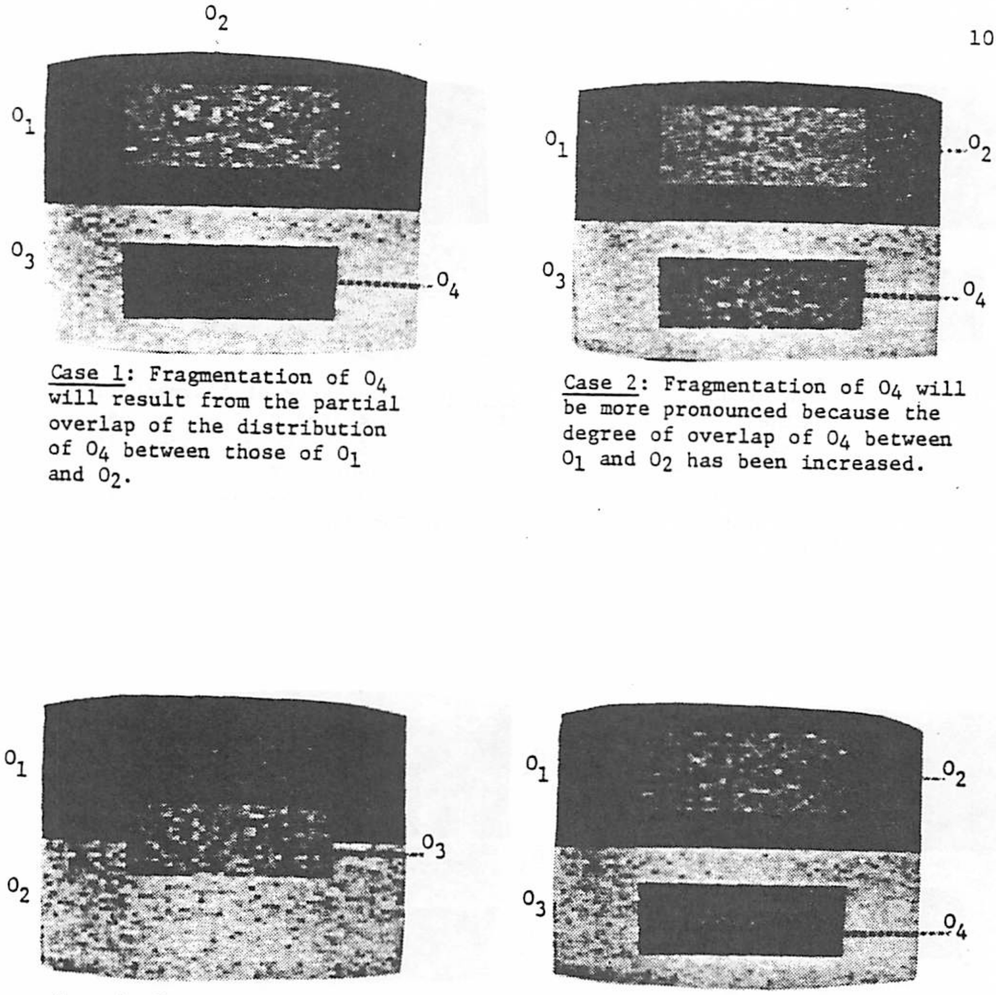
In the previous chapter, we discussed the details of the design and behavior of a segmentation algorithm based upon global statistics and a local update process. The algorithm was shown to yield reasonably accurate segmentations in noisy images with thin structures. The bulk of this chapter will be devoted to exploring two weaknesses of the algorithm whose effects were somewhat hidden in the previous discussion. These weaknesses stem from the global nature of the algorithm and can be demonstrated to yield disastrous results in images with certain characteristics. We will again make use of test images to explore problem situations. However, the analysis here will be much more comprehensive than that of Chapter III since, in addition to cluster overlap, the effects of relaxation will be accounted for.

The first weakness was explored in Chapter III and lies with the use of feature histograms computed globally across the entire image. In our algorithm, the peaks in the feature histogram are used to compute the initial probabilities associated with the label set of each pixel. It will be shown below that the global distribution is often a very poor reflection of the actual distributions of local objects. For example, clusters with relatively close means may not have distinguishable peaks and therefore the label set will not be representative of all the informatiion in the image.

The second weakness of the global algorithm can be seen in the computation of the structure-preserving compatibility coefficients used in the probability updating process. Here, there is a two-fold use of global information. First, the coefficients are computed as a function of the prior probabilities of each label, which are themselves reflective of possibly inaccurate global cluster information. More significantly, however, is the problem that the coefficients are computed across the entire image structure. This may prove to be a very poor reflection of the actual local information that will be encountered in any particular area. Thus the global compatibility coefficients may drive the local updating of probabilities toward the "average" structure which may be quite inaccurate.

The cases that will be presented in this chapter form an analysis of why the global algorithm converges to an incorrect segmentation in simple images in which the objects are locally discriminable. A figure is included with each case showing: (1) the test image containing numbered objects, (2) the global histogram indicating cluster labels, (3) the initial pixel classification into region labels, and finally (4) the converged results after application of two variant forms of the relaxation update rule. The two variations are plurality relaxation and probabilistic relaxation using conditional probabilities for the compatibility coefficients. Both of these algorithms were discussed in the previous chapter. Note that unless otherwise specified, the update rules are ıpplied using a 5-adjacency neighborhood (with the center pixel included as its own neighbor).

Figure V.0 is a compilation of the images that will be tested. In



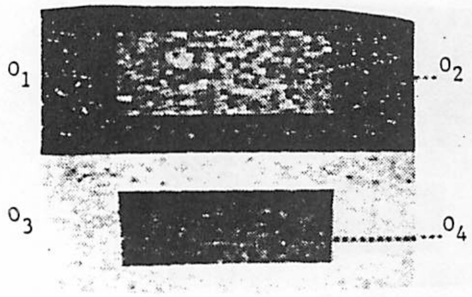
Case 1: Fragmentation of $O_4$ will result from the partial overlap of the distribution of $O_4$ between those of $O_1$ and $O_2$.

Case 2: Fragmentation of $O_4$ will be more pronounced because the degree of overlap of $O_4$ between $O_1$ and $O_2$ has been increased.

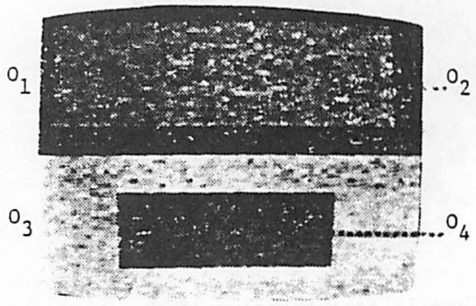
Case 3: Fragmentation and overmerging will occur because (1) $O_3$ is adjacent to $O_1$ and $O_2$ and (2) its distribution is entirely hidden within those of $O_1$ and $O_2$.

Case 4: Fragmentation of an object ($O_4$) with a piecewise-linear intensity profile. Relatively large error regions will form in the middle band of $O_4$ because the mislabelled pixels there are spatially correlated

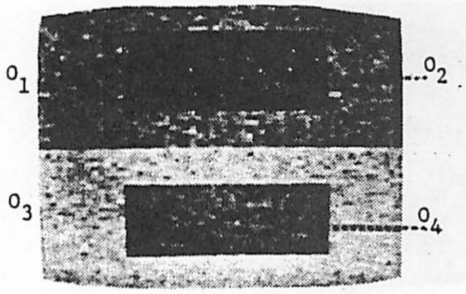
Figure V.0 Images to be used in the case studies. Brief captions are included which indicate the segmentation problems that will be encountered. (Continued)



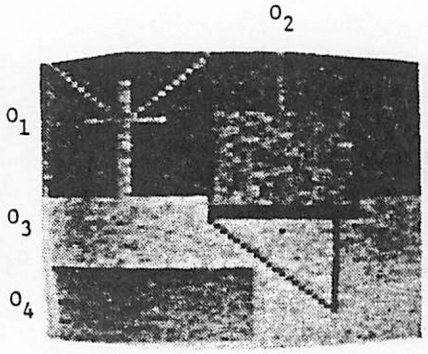
Case 5: Fragmentation of an object ($O_4$) with a linear intensity profile. An error region will form in the bottom band and will resist suppression because the competing label can attack only from above.



Case 6: The degree of fragmentation of $O_4$ will be greater than in Case 5 by decreasing the prior probability of the dominant label in $O_4$. This has been done by increasing the size of $O_2$ at the expense of $O_1$.



Case 7: The degree of fragmentation of $O_4$ will be greater than in Case 5 by decreasing the prior probability of the dominant label in $O_4$. This has been done simply by switching the means of $O_1$ and $O_2$.



Case 8: Thin-line fragmentation will occur because the lines have a relatively low frequency across the image and are therefore poorly represented in the compatibility coefficients.

Figure V.0 Images to be used in the case studies

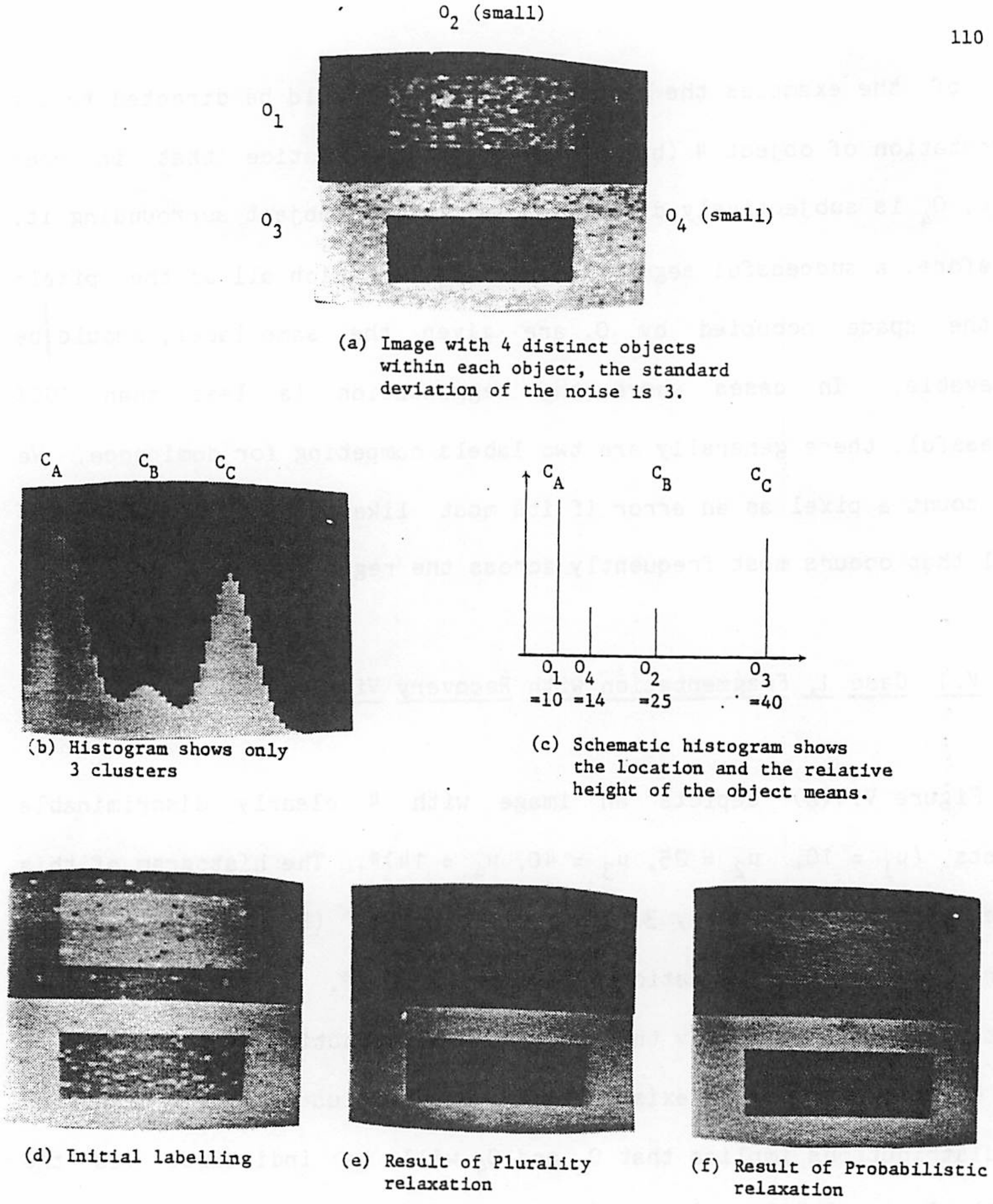
most of the examples the reader's attention should be directed to the segmentation of object 4 (but $O_3$ in case 3). Notice that in each image, $O_4$ is subjectively discriminable from the object surrounding it. Therefore, a successful segmentation of $O_4$, in which all of the pixels in the space occupied by $O_4$ are given the same label, should be achievable. In cases where the segmentation is less than 100% successful, there generally are two labels competing for dominance. We will count a pixel as an error if its most likely label is not the label that occurs most frequently across the region.

## V.1 Case 1, Fragmentation with Recovery Via Iterative Update

Figure V.1(a) depicts an image with 4 clearly discriminable objects ($\mu_1 = 10$, $\mu_2 = 25$, $\mu_3 = 40$, $\mu_4 = 14$)*. The histogram of this image, however, shows only 3 distinct clusters (Figure V.1(b). By referring to the schematic histogram IV.1c**, it can be seen that cluster A ($C_A$) is actually the sum of the distributions of objects 1 and 4 ($O_1$ and $O_4$). The existence of a single cluster to represent the two distributions implies that $O_1$ and $O_4$ will be indistinct via the global cluster labels. As we have seen before (see Case 2, Chapter 3), this situation potentially leads to overmerging: if $O_1$ and $O_4$ happen to be spatially adjacent, as well as having identical global labels,

*Note that $\sigma = 3$ in all objects in this chapter.

**The schematic histograms are obtained from an "ideal" (noise-free) image.



(a) Image with 4 distinct objects within each object, the standard deviation of the noise is 3.

(b) Histogram shows only 3 clusters

(c) Schematic histogram shows the location and the relative height of the object means.

(d) Initial labelling

(e) Result of Plurality relaxation

(f) Result of Probabilistic relaxation

Figure V.1 Case 1, Fragmentation - First Example: The iterative update rules can successfully recover from fragmentation due to cluster overlap when the density of mislabelled pixels is relatively sparse.

then only one region label will ultimately represent these two objects.

The situation is further complicated because the distribution of $O_4$, while mostly subsumed by that of $O_1$, is also partially overlapped with the distribution of $O_2$. Therefore fragmentation will result; that is, regardless of the spatial arrangement of $O_1$, $O_2$, and $O_4$, $O_4$ must initially be represented by two cluster labels ($C_A$ and $C_B$) in some mixture.

Figure V.1d shows the initial labelling obtained by minimum distance classification of the pixels into three clusters which are displayed as three distinct gray levels. Overmerging does not result because $O_3$ by chance spatially separates $O_4$ from $O_1$ and $O_2$. Fragmentation in the initial classification occurs in each region with relatively low frequency. Significantly, the mislabelled pixels in each region are randomly distributed, because the distribution of gray levels across each object was Gaussian; in particular there is a spatially invariant mean and the noise was spatially uncorrelated. The net effect is that the mislabelled pixels are spread randomly about the target regions and tend not to be spatially contiguous.

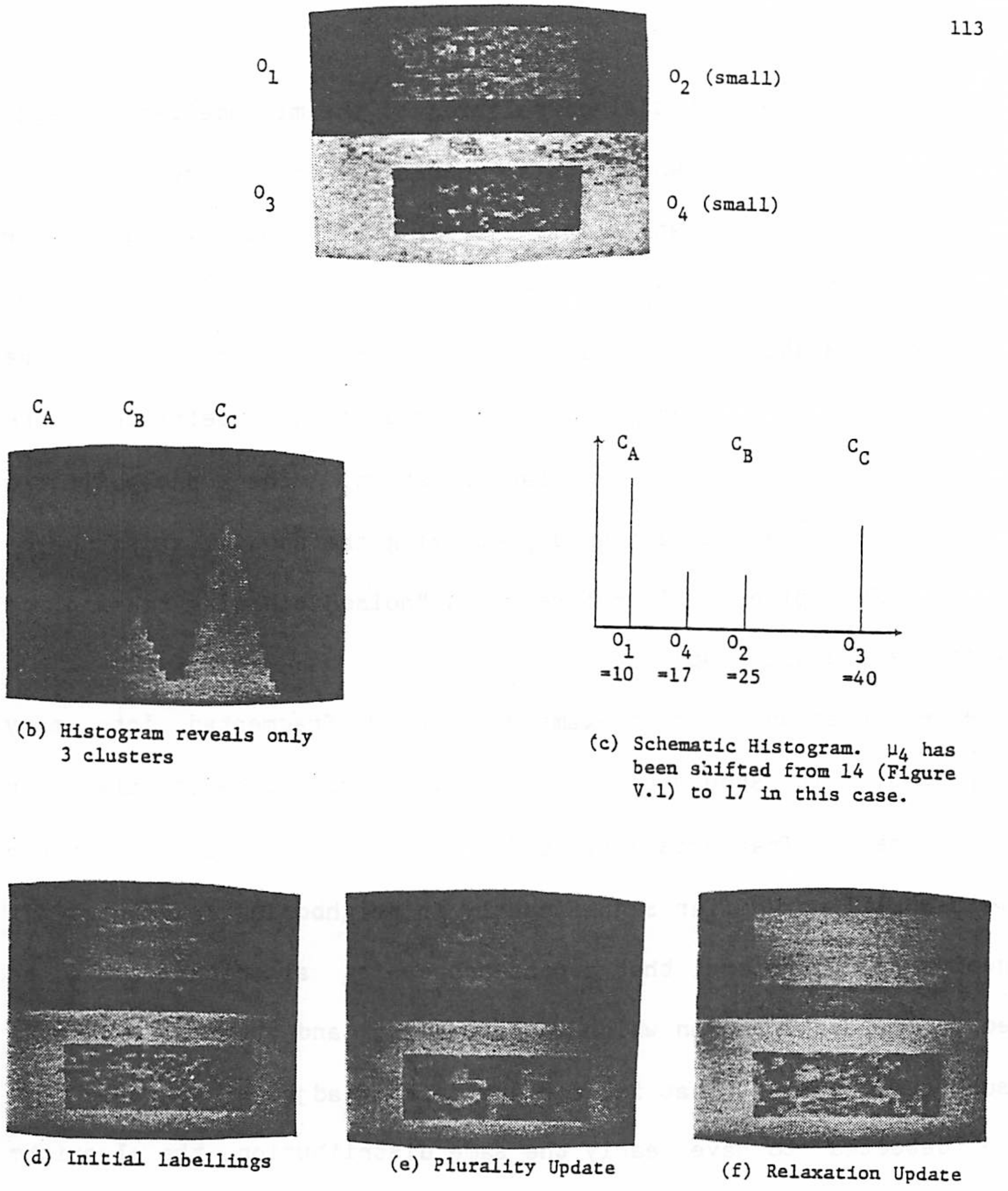
Figures V.1e and V.1f show the effect of two relaxation schemes applied to the initial probabilistic labelling. The figures show the highest probability label at each pixel. However, the probabilities themselves are not shown. Clearly, both update rules yield the desired effect of suppressing almost all of the 1 and 2 pixel region fragments into the dominant surrounding regions. In this example, neither the use of label probabilities nor the use of label compatibilities were necessary to correct the mislabelled pixels. Rather, the plurality rule simply takes advantage of the sparseness of the errors and the lack of any spatial correlation in the errors in order to succeed. However, the probabilistic relaxation scheme also operates effectively.

## V.2 Case 2, Unrecoverable Fragmentation

Figure V.2 shows the same image as in Case 1 except that the mean of $O_4$ has been shifted slightly, from 14 to 17. Locally, the contrast of the average edge between $O_3$ and $O_4$ has been only slightly weakened and is perceptually still clear to the human viewer. Globally, however, the distribution of $O_4$ is now completely ambiguous -- its mean is halfway between the means of $O_1$ and $O_2$ (10 and 25 respectively).

The initial labelling of this image (Figure V.2d) reveals the ambiguity in a striking manner. $O_4$ has been grossly fragmented into two label types, A and B. Consider what has occurred. A slight, linear shift in the global statistics of $O_4$ has created a tremendous change in the initial segmentation of the object. The problem is that the mean of $O_4$ is on the hyperplane boundary between $C_A$ and $C_B$ of a minimum distance classifier; small amounts of noise can vary the initial classification.

Next consider the behavior of the relaxation processes. In contrast to Case 1, the mislabelled pixels in this image are very densely populated. Consequently, a likely side effect is that some of the mislabelled pixels will be spatially adjacent. Note that the



(b) Histogram reveals only 3 clusters

(c) Schematic Histogram. $\mu_4$ has been shifted from 14 (Figure V.1) to 17 in this case.

(d) Initial labellings

(e) Plurality Update

(f) Relaxation Update

Figure V.2 Case 2, Fragmentation - Second Example: The hidden cluster corresponding to O4 is halfway between two detectable clusters. The update schemes organize the dense collection of mislabelled pixels in O4 into viable regions. The presence of these fragments clearly makes the segmentations unacceptable. However, since all of these region fragments have nearly the same feature value, O4 may yet be "recoverable" via a region merging process.

effect is equivalent to spatial correlation of the mislabelled pixels, although the locations where this occurs within the area of $O_4$ are random. The plurality update simply "fills in" areas wherever one label happens to be slightly dominant over another. This process continues until stable (but randomly configured) region shapes are attained. On the other hand, since the compatibility coefficients are representative of the initial classification, the probabilistic relaxation process is biased toward preserving the spatial structure of the misclassified pixels. Therefore, less "noise" cleaning takes place than with the plurality update.

The segmentations of this example leave $O_4$ fragmented into many small pieces. It should be noted, however, that since the pixels in each of the region fragments derived from the same population, their gray values will not differ significantly in neighboring fragments. It is conceivable, therefore, that a post-processing algorithm could be applied to the segmentation which would look for and attempt to recover from such a situation. That is, for any pair of adjacent regions that can be detected to have nearly the same distriibution, the algorithm could relabel all of the pixels involved with the same region label. This would (hopefully) merge all of the pieces of $O_4$ into a single unit. This technique will be explored in detail in Chapter VI.

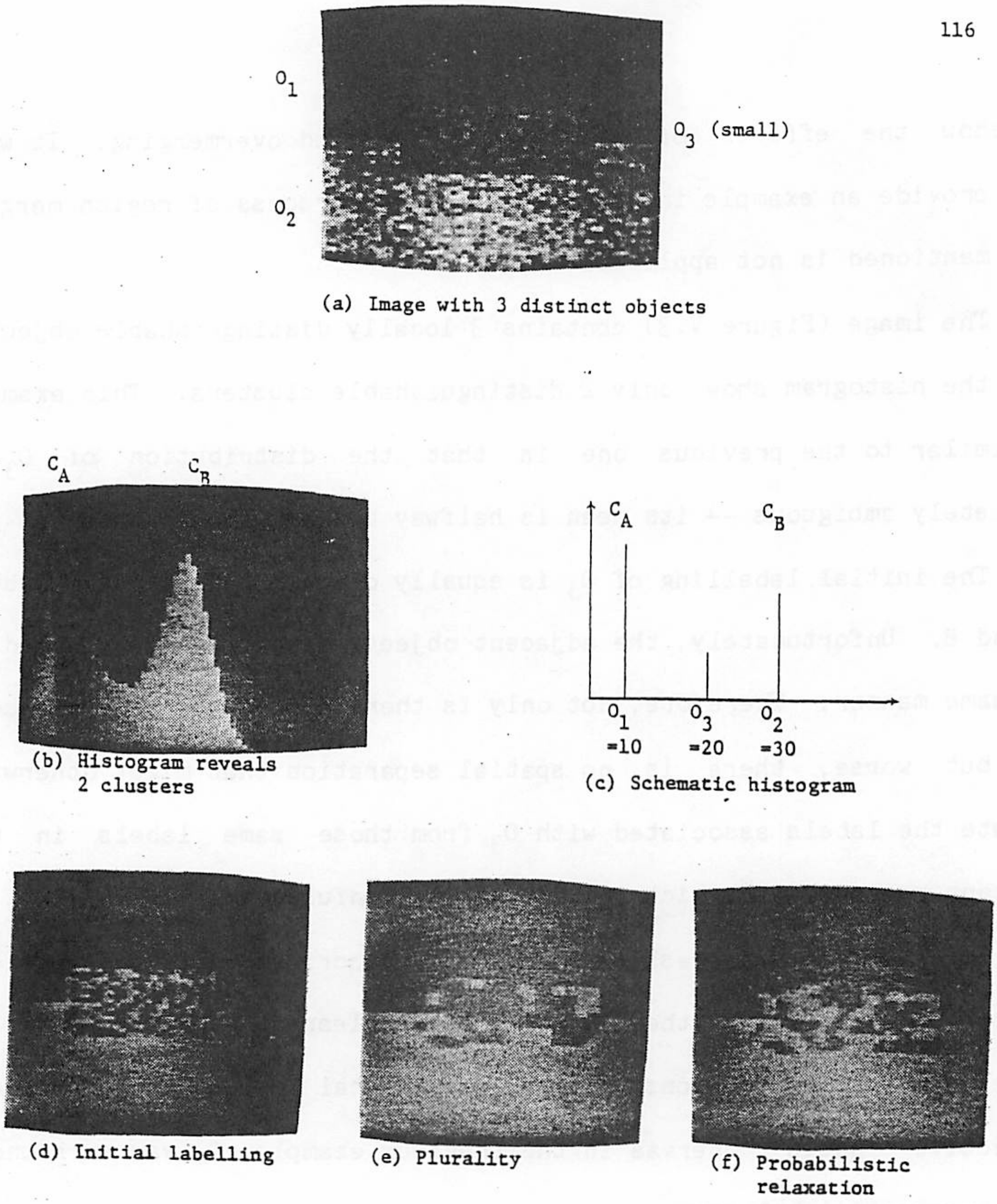
## V.3 Case 3, Fragmentation and Overmerging

This case is an extension of the previous example and is designed to show the effects of both fragmentation and overmerging. It will also provide an example in which the recovery process of region merging just mentioned is not applicable.

The image (Figure V.3) contains 3 locally distinguishable objects, but the histogram shows only 2 distinguishable clusters. This example is similar to the previous one in that the distribution of $O_3$ is completely ambiguous -- its mean is halfway between the means of $O_1$ and $O_2$. The initial labelling of $O_3$ is equally distributed between labels A and B. Unfortunately, the adjacent objects happen to be labelled in the same manner. Therefore, not only is there no cluster to represent $O_3$, but worse, there is no spatial separation that might otherwise isolate the labels associated with $O_3$ from those same labels in the adjacent regions with which it is globally confused.

The relaxation processes behave as described in the previous example, except that there is a greater clean-up effect in $O_3$. The large surrounding regions provide additional support for their respective labels; whereas in the previous example, $O_4$ was surrounded by a "neutral" label type.

Recovery of $O_3$ as a single region presents a very difficult problem. First, the major regions shown in Figure V.3f would have to be re-histogrammed with the hope that they would reveal significant bimodality. The bimodality, if detectable, would indicate the presence of a new cluster type, say $C_C$, corresponding to $O_3$. The regions with bimodal distributions would then be split, and by recursively applying the whole algorithms to each piece, all of the newly formed region



(a) Image with 3 distinct objects

(b) Histogram reveals 2 clusters

(c) Schematic histogram

(d) Initial labelling

(e) Plurality

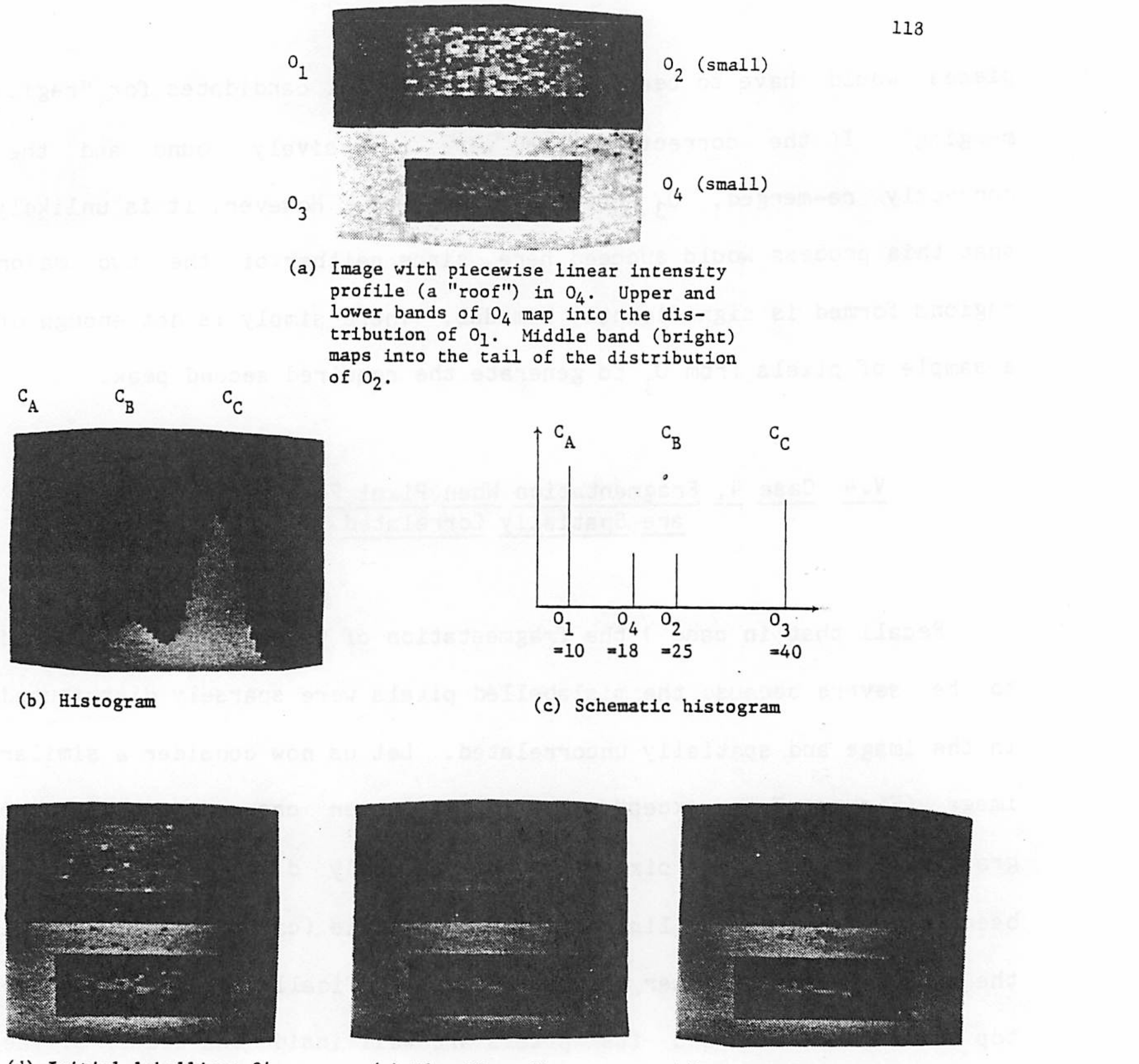
(f) Probabilistic relaxation

<u>Figure V.3</u> <u>Case 3, Fragmentation and Overmerging</u>: $O_3$ is spatially adjacent to two objects whose distributions contains that of $O_3$. Thus, $O_3$ is both overmerged and fragmented with little possibility of recovery.

pieces would have to been checked as potential candidates for "region merging". If the correct pieces were recursively found and then correctly re-merged, $O_3$ could be recovered. However, it is unlikely that this process would succeed here, since neither of the two major regions formed is significantly bimodal. There simply is not enough of a sample of pixels from $O_3$ to generate the required second peak.

## V.4 Case 4, Fragmentation When Pixel Feature Values are Spatially Correlated

Recall that in case 1 the fragmentation of $O_4$ was not considered to be severe because the mislabelled pixels were sparsely distributed in the image and spatially uncorrelated. Let us now consider a similar image (Figure V.4) except that $O_4$ has been changed so that the gray-level value of its pixels are not randomly distributed. $O_4$ has been given a piecewise linear intensity profile (called a "roof") with the center having brighter values. More specifically, the mean of the top band is 11.5 and its pixels are well inside the center of the distribution of $O_1$ (with mean 10). The gradient has been constructed so that the mean of each row is slightly higher than the mean of the previous row. This is continued until the center row of $O_4$ ($\mu = 18$) is reached, at which point the row means are gradually decreased until, in the bottom row, the mean is again 11.5. Significantly, the mean of the center band is 18 which is just inside the tail of the distribution of $O_2$ ($C_B$). Keep in mind however that $O_4$ is perceivable as a single object and should be ideally segmented as such. While it could be



(a) Image with piecewise linear intensity profile (a "roof") in $O_4$. Upper and lower bands of $O_4$ map into the distribution of $O_1$. Middle band (bright) maps into the tail of the distribution of $O_2$.

(b) Histogram

(c) Schematic histogram

(d) Initial labelling: 94 pixels in $O_4$ are labelled as B and the remaining 418 are labelled as A.

(e) Plurality: 51 pixels in $O_4$ are labelled as B.

(f) Probabilistic relaxation: 30 pixels in $O_4$ are labelled as B.

Figure V.4 Case 4, Fragmentation of a Spatially Correlated Region. The impact of fragmentation depends upon both the spatial organization of the pixels involved and their location in feature space. The pixels in the center band of the gradient in $O_4$ map into $C_B$. They form viable regions that the update schemes cannot totally suppress.

argued that there is a central light band that is extractable, we do not believe that there is any clear boundary along which the partitioning would be justifiable.

The initial labelling in the regions representing $O_1$, $O_2$, and $O_3$ (i.e., the output of the minimum distance classifier) shows a small percentage of mislabelled pixels that are randomly distributed across the regions. $O_4$, however, shows a set of mislabelled pixels that are spatially correlated about the center band. Note that at this stage, $O_4$ contains 418 pixels labelled A and 94 pixels labelled B. If we call B the erroneous (or non-dominant) label, then the segmentation of $O_4$ has an initial error rate of approximately 18%.

Now consider the behavior of the plurality relaxation process. The plurality scheme starts with the initial labelling and simply fills in "holes" until stable structures are reached. Since the mislabelled pixels in $O_4$ (labelled B) are more or less contiguous, they are sufficiently cohesive to maintain their own label identity and suppress any pixels labelled A that are within the center band. Of course, the pixels labelled A are competing for dominance at the center band from above and below and are clearly the dominant force across the region. Accordingly, the error rate is reduced from 18% to 10% in 8 iterations.

Next, consider the probabilistic relaxation update. This is the first example in which the probabilities of label types can be shown to be useful. Although the minimum distance classifier generally labels the pixels in the center band of $O_4$ as B, the actual probabilities of these pixels are very close to .5 for both labels A and B. This area is, therefore, highly ambiguous, although slightly biased towards label B. Since the pixels there are ambiguously labelled, it will take many iterations for them to converge to a more consistent label. Indeed, the power of probabilistic relaxation lies with the ability to defer labelling until more contextual information propagates inward from greater distances. The prediction of deferred labelling is borne out by two observations: (1) it requires many iterations (i.e., 15) for the pixels in the center band to reach a high probability in some label, and (2) the error rate ultimately is reduced from 18% to less than 6%, which is a significant improvement over the plurality scheme.

Before leaving this example, let us carefully examine the compatibility coefficients to gain an understanding of how they represent the information in the image. Table V.1 shows the compatibility coefficients for the initial probability labelling of the image. Four tables are presented, each corresponding to a different neighbor relation, namely "above," "to the right," "below," and "to the left" as measured from the central pixel in a square 3x3 window. Each table has 3 rows and columns which correspond to the three labels $C_A$, $C_B$, and $C_C$.

Let us briefly discuss some of the important on-diagonal label relations (other relations where $\alpha \neq \beta$ will be discussed in later cases). First, in all orientations, labels A given A and C given C are highly compatible, while B given B is less compatible. These relations can be explained by noting that the size of a cluster directly affects the image-wide prior probability of that cluster label, and that the

## TABLE V.1

Compatibility coefficients for case 4. Coefficients are a function of the conditional probability, P(α|β) and are specified between all pairs of pixels $A_i$ and $A_j$, where $A_j$ ε Neighborhood of $A_i$.

Compats for Neighbor Relation: "Above" (North)

| I\J | A | B | C |
|---|---|---|---|
| A | 0.4266 | -0.1587 | -0.6132 |
| B | -0.1387 | 0.1836 | -0.2367 |
| C | -0.5463 | -0.2152 | 0.4243 |

Compats for Neighbor Relation: "To the Right" (East)

| I\J | A | B | C |
|---|---|---|---|
| A | 0.4494 | -0.1855 | -0.6330 |
| B | -0.1812 | 0.1940 | -0.2089 |
| C | -0.6341 | -0.2117 | 0.4788 |

Compats for Neighbor Relation: "Below" (South)

| I\J | A | B | C |
|---|---|---|---|
| A | 0.3843 | -0.1516 | -0.5462 |
| B | -0.1716 | 0.1845 | -0.1988 |
| C | -0.6132 | -0.2203 | 0.4687 |

Compats for Neighbor Relation: "To the Left" (West)

| I\J | A | B | C |
|---|---|---|---|
| A | 0.4467 | -0.1790 | -0.6330 |
| B | -0.1833 | 0.1953 | -0.2089 |
| C | -0.6319 | -0.2061 | 0.4752 |

Compats for Neighbor Relation: "Center"

| I\J | A | B | C |
|---|---|---|---|
| A | 1.0000 | -1.0000 | -1.0000 |
| B | -1.0000 | 1.0000 | -1.0000 |
| C | -1.0000 | -1.0000 | 1.0000 |

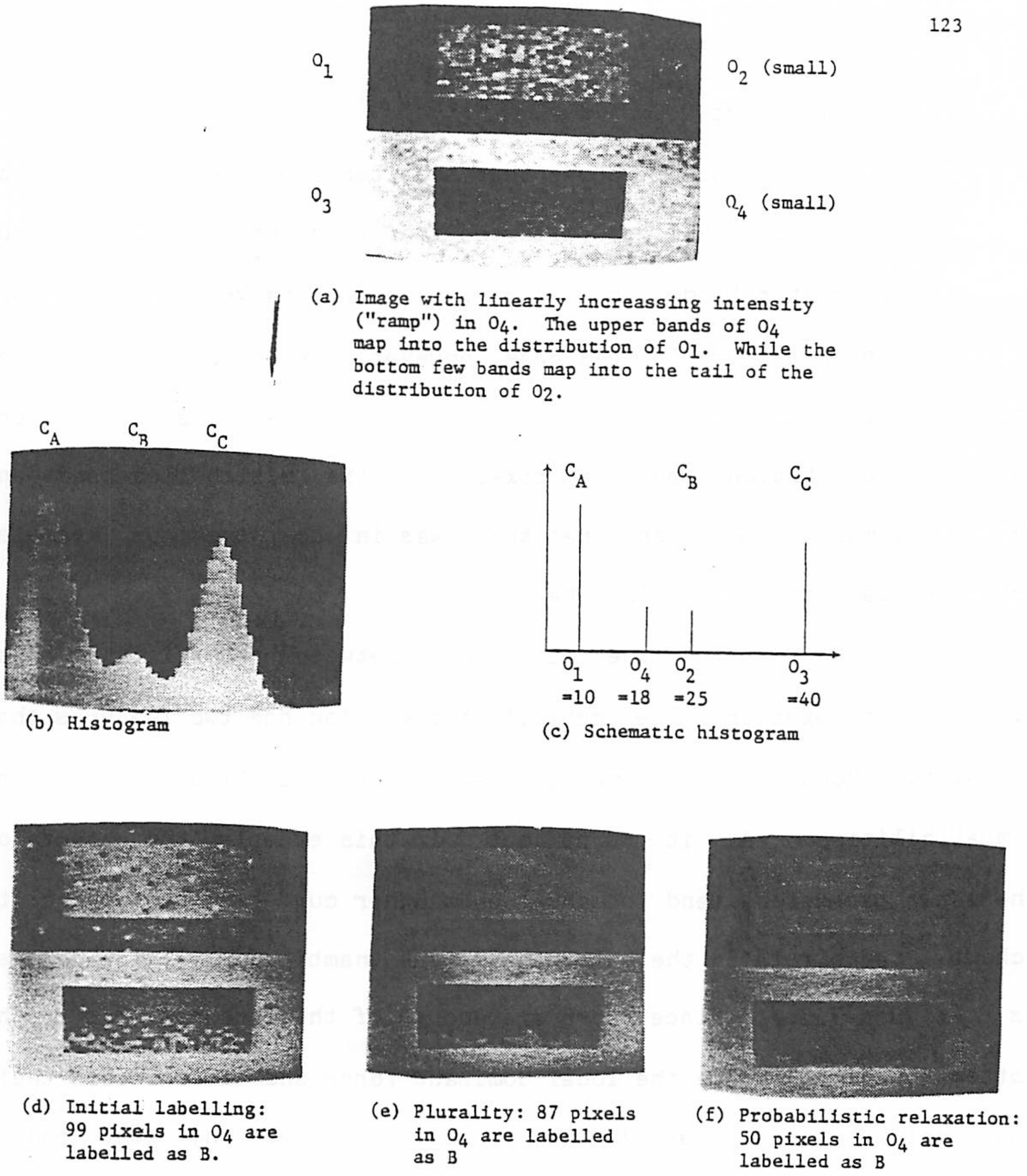
relative sizes of the clusters are size($C_A$) > size($C_C$) > size($C_B$). A small cluster implies that there will be few pixels in the image that have a high probability label for that cluster.

The compatibility coefficients, therefore, indicate that the RLP is biased toward promoting the probability of label A (or C) over label B. This partially explains why the plurality scheme has a higher error rate. All labels are equally likely in the plurality scheme, whereas the inclusion of conditional probabilities biases some labels over others. In this case, the biased coefficients help destroy the B-band in the center of $O_4$.

It is important to realize, however, that the bias is not necessarily desirable. In fact it is simply fortuitous in this case: if cluster A were smaller, then the error rate in $O_4$ would be larger. In general one should ask why a non-local effect -- such as the size of a distant object -- should have any impact on the local decision as to what label should be promoted over another. This question will arise again in later cases.

## V.5 Case 5, A Second Example of Spatially Correlated Intensities; Iterative Update is Not as Effective

This case (Figure V.5) is similar to the previous one except that the intensity across object 4 is linearly increasing from mean 11 to mean 18 (top to bottom). Thus, the initial classification (Figure V.5d) reveals a band at the bottom of the region corresponding to that portion of the distribution of object 4 that is just inside the



(a) Image with linearly increasing intensity ("ramp") in $O_4$. The upper bands of $O_4$ map into the distribution of $O_1$. While the bottom few bands map into the tail of the distribution of $O_2$.

(b) Histogram

(c) Schematic histogram

(d) Initial labelling: 99 pixels in $O_4$ are labelled as B.

(e) Plurality: 87 pixels in $O_4$ are labelled as B

(f) Probabilistic relaxation: 50 pixels in $O_4$ are labelled as B

Figure V.5 Case 5, Fragmentation of a Spatially Correlated Region - Second Example: Here, the relaxation processes yield a higher error rate than in the previous example, because the mislabelled "B" band in $O_4$ is attacked by the "A" label only from above.

distribution of object 2.

This example, however, shows that the iterative schemes did not recover nearly as well as with the roof gradient. In the previous example, the center band (labelled B) was being attacked from above and below by the A label. In this case, however, the bottom band is being attacked only from above. While it is not obvious at this point, there is less competition for the pixels in the mislabelled band--and relatively more cooperation--than there was in the previous example. Let us explain this point in detail.

First, we examine the difference between probabilistic and plurality relaxation. The probabilistic version has two features that plurality does not, namely, label probabilities and label compatibilities. As it turns out, in this example, the effects of these two parameters tend to cancel each other out. In the plurality scheme, the pixels in the bottom bands are unambiguously (mis)labelled as B (Figure V.5e). Since there are enough of the erroneous B's in the bottom bands, they are the local dominant force and can maintain their own label with this updating rule, as well as consume any <u>contained</u> A labels.

In the probabilistic relaxation scheme (Figure V.5f), however, the pixels in the bottom band are highly ambiguous, although they are slightly biased toward label B. One might assume, therefore, that these pixels will simply go with the flow, which in $O_4$ means label A. However, the effect of the compatibility coefficients must also be considered (Table V.2). The compatibility between labels A and C is

TABLE V.2

Compatibility coefficients for Case 5. The coefficients are a function of the conditional probability, $P(\alpha|\beta)$ and are specified between all pairs of pixels $A_i$ and $A_j$, where $A_j \varepsilon$ Neighborhood of $A_i$.

Coefficients for Neighbor Relation: "Above" (North)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.4324 | -0.1729 | -0.6143 |
| B | -0.1322 | 0.1819 | -0.2404 |
| C | -0.5615 | -0.1944 | 0.4253 |

Coefficients for Neighbor Relation: "To the Right" (East)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.4471 | -0.1834 | -0.6316 |
| B | -0.1786 | 0.1940 | -0.2119 |
| C | -0.6330 | -0.2140 | 0.4779 |

Coefficients for Neighbor Relation: "Below" (South)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.3901 | -0.1451 | -0.5615 |
| B | -0.1858 | 0.1827 | -0.1781 |
| C | -0.6143 | -0.2241 | 0.4697 |

Coefficients for Neighbor Relation: "To the Left" (West)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.4443 | -0.1764 | -0.6320 |
| B | -0.1812 | 0.1953 | -0.2113 |
| C | -0.6306 | -0.2091 | 0.4743 |

Coefficients for Neighbor Relation: "Center"

| I \ J | A | B | C |
|---|---|---|---|
| A | 1.0000 | -1.0000 | -1.0000 |
| B | -1.0000 | 1.0000 | -1.0000 |
| C | -1.0000 | -1.0000 | 1.0000 |

much more negative than the compatibility between B and C. This is because of the global relationship betwen A, B, and C, i.e. the mean of $C_C$ is closer to the mean of $C_B$ than it is to the mean of $C_A$.

Now, a pixel inside object 3 is going to have (on the average) a low probability of being label A, a higher probability of being label B and a much higher probability of being label C. Since object 3 is rather large, its pixels will make a rather large contribution to the global compatibility coefficients, and the latter will be strongly influenced by these relationships.

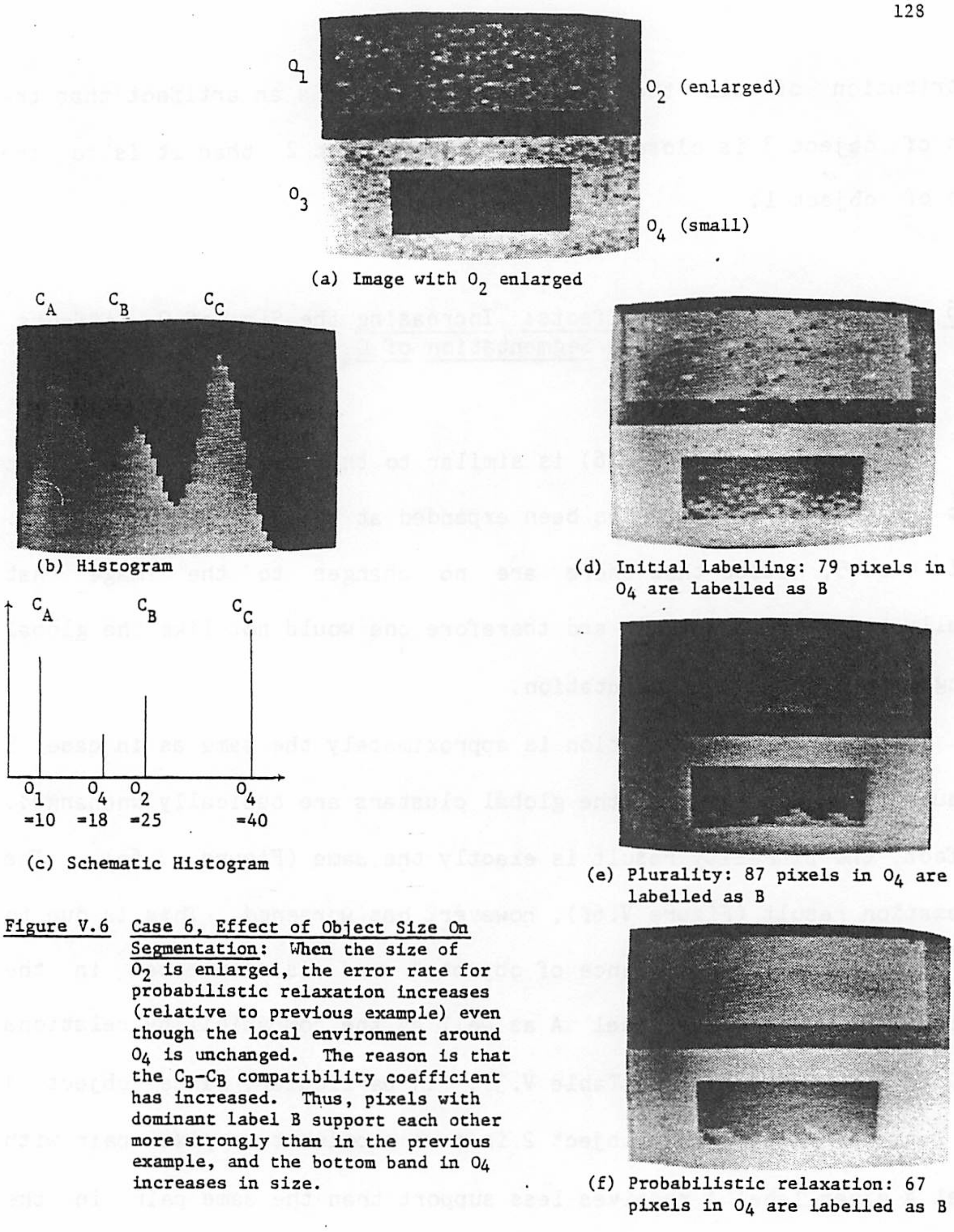
Now consider a pixel located in the bottom band of object 4 with ambiguity tetween labels A and B. The coefficients will favor label B over label A because label B has a less negative compatibility with label C in the orientation "below" (-.1781) than label A has with label C (-.5615). Moreover, since the pixels in object 3 are very strongly biased toward label C, the support that is given to pixels in object 4 is not only less incompatible with label B, but is also highly probable. To summarize, labels B and C cooperate to preserve label B more strongly than labels A and B compete to destroy label B.

One might argue that the relaxation labelling process is behaving in a desirable manner, i.e. that it is preserving a thin structure (the "B-band"). However, this is not the case. First, it should be remembered that the updating process mostly destroyed the thin band (with the same characteristics) associated with the roof gradient. Second, the action of the compatibility coefficients to boost label B at the bottom of object 4 is purely an artifact of the global distribution of all the regions, namely, it is an artifact that the mean of object 3 is closer to the mean of object 2 than it is to the mean of object 1.

## V.6 Case 6, Global Side-Effects: Increasing the Size of $O_2$ Affects the Segmentation of $O_4$

This example (Figure V.6) is similar to the previous one except that the area of object 2 has been expanded at the expense of object 1. As in case 5, notice that there are no changes to the image that locally effect object 4, and therefore one would not like the global changes to effect its segmentation.

The initial classification is approximately the same as in case 5 because the positions of the global clusters are basically unchanged. In fact, the plurality result is exactly the same (Figure V.6e). The relaxation result (Figure V.6f), however, has worsened. This is due to the decreased global influence of object 1 and is reflected in the prior probability of label A as well as the compatibility relations between A-A, B-B, and A-B (Table V.3). In particular, since object 1 is less prominent and object 2 is more prominent, a pixel pair with label A given label A receives less support than the same pair in the image in case 5. The pair B-B receives more support in case 6 for the same reason. Thus in comparing on-diagonal relationships between the two cases, we conclude that B's have gained in self-support. Although we will not explore them here, the reader can check that the overall effects of the off-diagonal relations (e.g., $B\text{-}C_{below}$, $C\text{-}B_{above}$,



(a) Image with $O_2$ enlarged

(b) Histogram

(c) Schematic Histogram

(d) Initial labelling: 79 pixels in $O_4$ are labelled as B

(e) Plurality: 87 pixels in $O_4$ are labelled as B

(f) Probabilistic relaxation: 67 pixels in $O_4$ are labelled as B

Figure V.6 Case 6, Effect of Object Size On Segmentation: When the size of $O_2$ is enlarged, the error rate for probabilistic relaxation increases (relative to previous example) even though the local environment around $O_4$ is unchanged. The reason is that the $C_B$-$C_B$ compatibility coefficient has increased. Thus, pixels with dominant label B support each other more strongly than in the previous example, and the bottom band in $O_4$ increases in size.

## TABLE V.3

Compatibility coefficients for case 6. The coefficients are a function of the conditional probability, P($\alpha|\beta$) and are specified between all pairs of pixels $A_i$ and $A_j$, where $A_j$ ε Neighborhood of $A_i$.

Coefficients for Neighbor Relation: "Above" (North)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.4066 | -0.2176 | -0.5734 |
| B | -0.1780 | 0.2527 | -0.3217 |
| C | -0.5089 | -0.2875 | 0.4014 |

Coefficients for Neighbor Relation: "To the Right" (East)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.4235 | -0.2305 | -0.5930 |
| B | -0.2287 | 0.2669 | -0.3021 |
| C | -0.5934 | -0.3054 | 0.4540 |

Coefficients for Neighbor Relation: "Below" (South)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.3593 | -0.1897 | -0.5086 |
| B | -0.2293 | 0.2536 | -0.2751 |
| C | -0.5730 | -0.3093 | 0.4453 |

Coefficients for Neighbor Relation: "To the Left" (West)

| I \ J | A | B | C |
|---|---|---|---|
| A | 0.4214 | -0.2272 | -0.5923 |
| B | -0.2291 | 0.2677 | -0.3034 |
| C | -0.5919 | -0.3001 | 0.4505 |

Coefficients for Neighbor Relation: "Center"

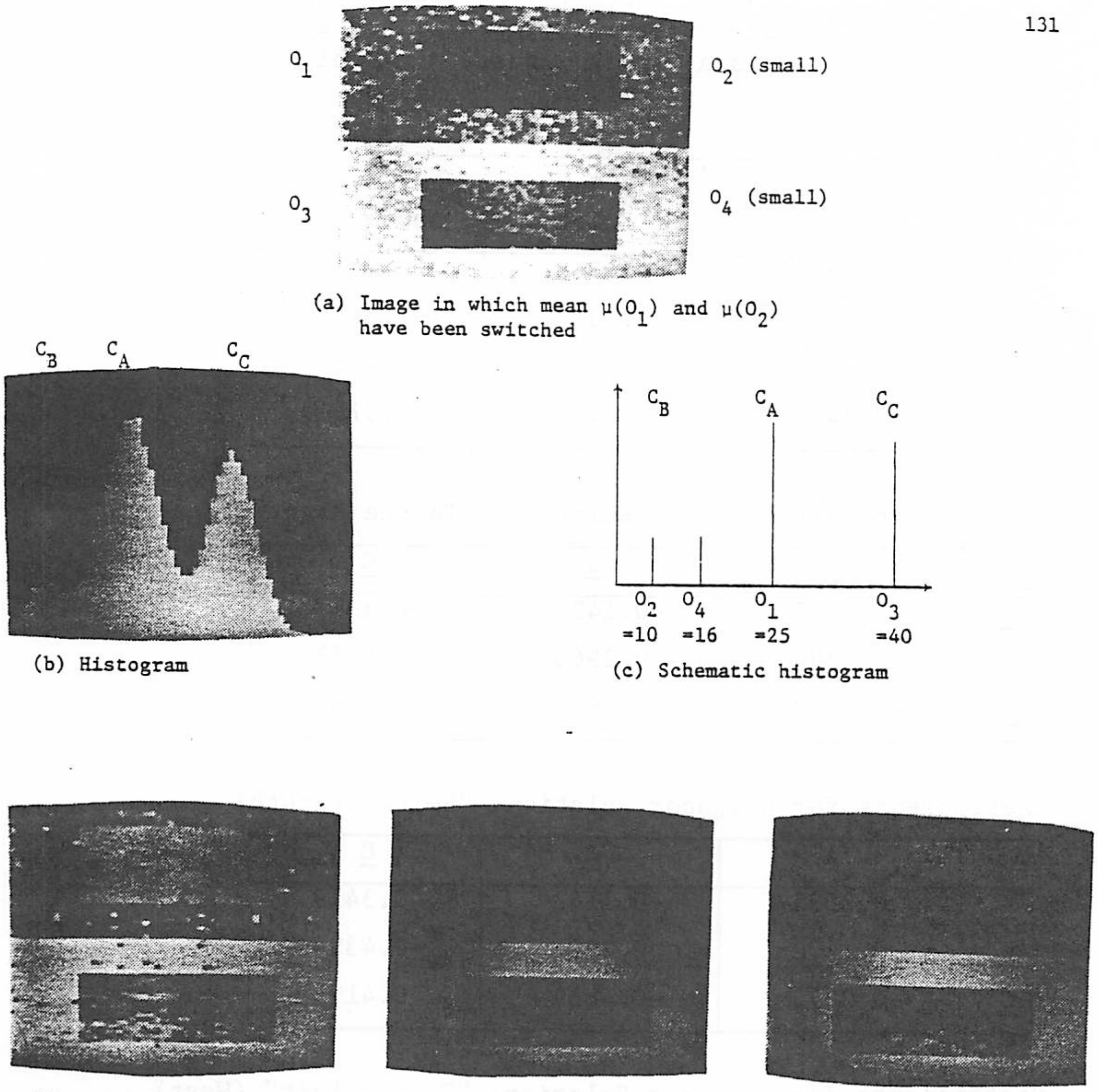
| I \ J | A | B | C |
|---|---|---|---|
| A | 1.0000 | -1.0000 | -1.0000 |
| B | -1.0000 | 1.0000 | -1.0000 |
| C | -1.0000 | -1.0000 | 1.0000 |

$A\text{-}B_{below}$, $B\text{-}A_{above}$) are more or less neutral between the two cases.

## V.7 Case 7, More Global Side-Effects: Swapping the Means of Objects 1 and 2

The lower half of the image in case 7 (Figure V.7a) is the same as in case 5; however, the intensity values of the objects in the upper half have been reversed. Here, $\mu_1$ and $\mu_2$ have been swapped, so that now $\mu_1$ is between $\mu_2$ and $\mu_3$ ($\mu_1 = 25$, $\mu_2 = 10$, $\mu_3 = 40$, $\mu_4 = 16$). This example was constructed to show another global side-effect on the compatibility coefficients and, therefore, on the local performance of the relaxation process.

The initial classification (Figure V.7d) is approximately the same as in case 5 and therefore the plurality result (Figure V.7e) is approximately the same also. However, the result of probabilistic relaxation (Figure V.7f) is somewhat worse due to the weakened compability of label A with itself and the relatively strengthened compatibility of B with itself (compare the on-diagonal relations in Tables V.2 and V.4). This can be understood by considering the new location of cluster A in feature space. Since it now lies between two clusters, any deviation from its mean value not only lowers the probability of A, but at the same time strengthens the probability of B or C. As in the previous cases, a globally distant change alters the performance of the RLP in a local area.



(a) Image in which mean $\mu(O_1)$ and $\mu(O_2)$ have been switched

(b) Histogram

(c) Schematic histogram

(d) Initial labelling: 99 errors

(e) Plurality: 78 errors

(f) Probabilistic relaxation: 75 errors

Figure V.7 Case 7, Global Side-Effects of Switching Intensity Values: The error rate for probabilistic relaxation increases even though the local environment around $O_4$ is unchanged. The reason is that the $C_A$-$C_A$ compatibility has been decreased since $C_A$ lies between $C_B$ and $C_C$, and therefore, the initial probabilities for label A vs label A are reduced relative to those in case 5.

## TABLE V.4

Compatibility coefficients for case 7. The coefficients are a function of the conditional probability, $P(\alpha|\beta)$ and are specified between all pairs of pixels $A_i$ and $A_j$, where $A_j$ $\varepsilon$ Neighborhood of $A_i$.

Coefficients for Neighbor Relation: "Above" (North)

| J<br>I | A | B | C |
|---|---|---|---|
| A | 0.2964 | -0.1353 | -0.3886 |
| B | -0.1029 | 0.2835 | -0.4923 |
| C | -0.3443 | -0.4458 | 0.3771 |

Coefficients for Neighbor Relation: "To the Right" (East)

| J<br>I | A | B | C |
|---|---|---|---|
| A | 0.3047 | -0.1433 | -0.3977 |
| B | -0.1396 | 0.2966 | -0.4783 |
| C | -0.3980 | -0.4807 | 0.4250 |

Coefficients for Neighbor Relation: "Below" (South)

| J<br>I | A | B | C |
|---|---|---|---|
| A | 0.2595 | -0.1147 | -0.3429 |
| B | -0.1472 | 0.2924 | -0.4394 |
| C | -0.3872 | -0.4859 | 0.4195 |

Coefficients for Neighbor Relation: "To the Left" (West)

| J<br>I | A | B | C |
|---|---|---|---|
| A | 0.3015 | -0.1371 | -0.3971 |
| B | -0.1408 | 0.2975 | -0.4789 |
| C | -0.3957 | -0.4765 | 0.4217 |

Coefficients for Neighbor Relation: "Center"

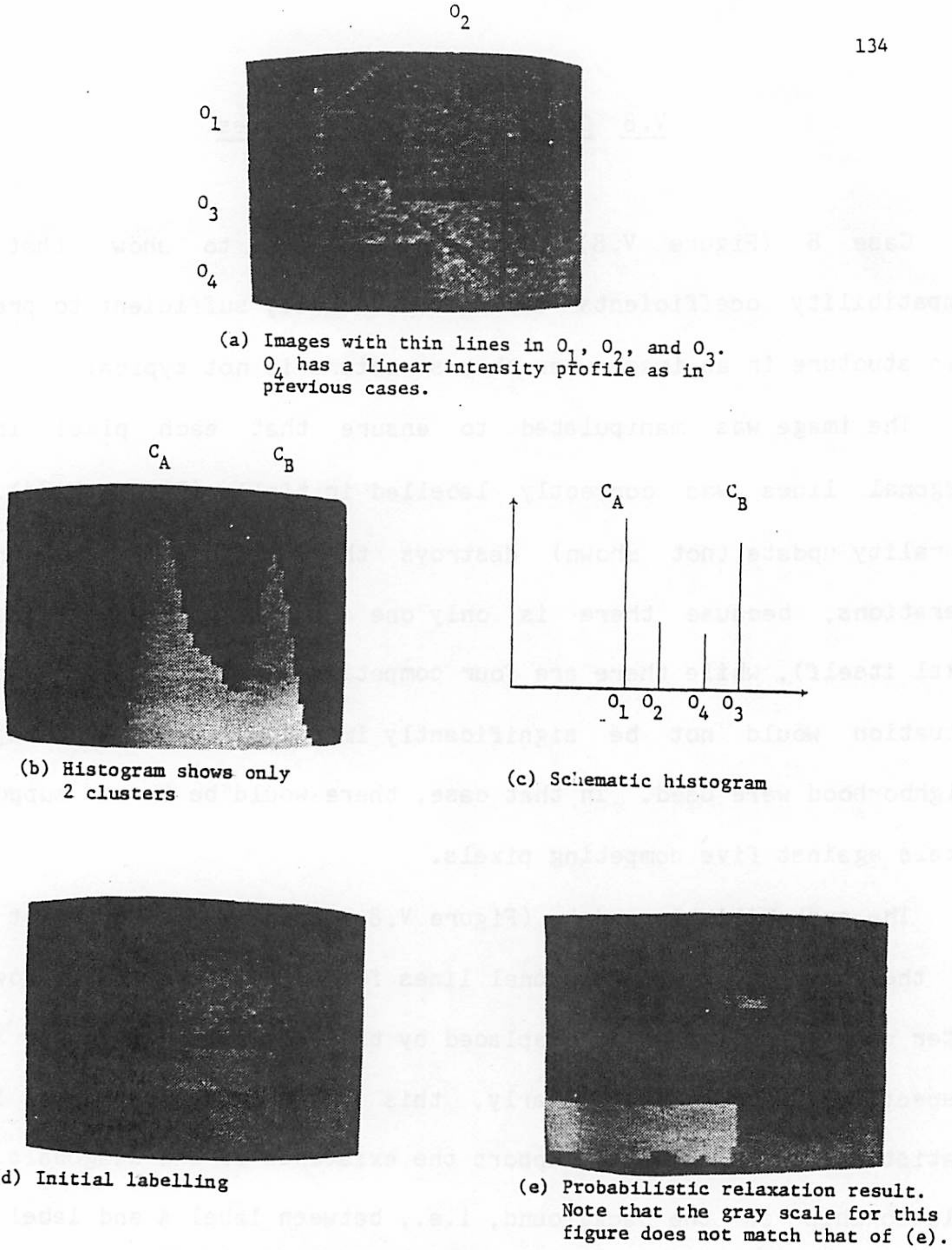
| J<br>I | A | B | C |
|---|---|---|---|
| A | 1.0000 | -1.0000 | -1.0000 |
| B | -1.0000 | 1.0000 | -1.0000 |
| C | -1.0000 | -1.0000 | 1.0000 |

## V.8 Case 8, Adding Thin Lines

Case 8 (Figure V.8) has been included to show that the compatibility coefficients are not necessarily sufficient to preserve thin stucture in an image when that structure is not typical.

The image was manipulated to ensure that each pixel in the diagonal lines was correctly labelled initially (Figure V.8d). The plurality update (not shown) destroys these lines in one or two iterations, because there is only one supporting pixel (the central pixel itself), while there are four competing pixels. Notice that this situation would not be significantly improved even if an 8-adjacent neighborhood were used. In that case, there would be three supporting pixels against five competing pixels.

The probabilistic update (Figure V.8e) maintains the correct label of the pixels in the diagonal lines for a few iterations. However, after convergence, they are replaced by the dominant label in their respective surrounds. Clearly, this image contains very little statistical information to support the existence of the diagonals. The relationship in the background, i.e., between label A and label A (or label B and label B), is much stronger than the relationship across the diagonals, i.e., between label A and label B. Thus the pixels along the diagonal lines get modest positive support from all neighbors while the pixels in the backgrounds receive very strong positive support from their neighbors.



(a) Images with thin lines in $O_1$, $O_2$, and $O_3$. $O_4$ has a linear intensity profile as in previous cases.

(b) Histogram shows only 2 clusters

(c) Schematic histogram

(d) Initial labelling

(e) Probabilistic relaxation result. Note that the gray scale for this figure does not match that of (e).

Figure V.8 Case 8, Thin-Line Fragmentation: All thin lines are destroyed by relaxation because their frequency across the image is too low to be significant in the compatibility coefficients. The plurality result is not shown here.

## V.9 Conclusion

The examples in this chapter were constructed to show the powerful impact of the global image characteristics upon the local iterative update processes. In each case, an image was depicted in which all regions locally were quite discriminable, yet the globally-based segmentation algorithm was unable to yield satisfactory results. Global influences such as partially and totally obscured peaks, region sizes, non-zero gradients across regions, and thin structure frequency were all shown to affect the performance of the segmentation algorithm. In the next chapter we will show that localization of the algorithm to subimages can alleviate many of the problems explored in this chapter.

# CHAPTER VI

## LOCALIZED SEGMENTATION VIA PARTITIONING AND MERGING

The previous chapter explored some obvious pitfalls of the global segmentation algorithm. In each of the cases depicted the data was locally discriminable, yet some information was globally obscured. The results showed that regions could be broken or torn into fragments that might not be readily reassembled. The existence of these cases has motivated us to reformulate the segmentation algorithm as outlined in Figure VI.1. The basic idea is to focus on local areas of the image that are small enough to reveal local clusters and local activity yet large enough to be statistically meaningful.

In the new formulation the image is initially partitioned into regularly spaced, square sub-images, called sectors. The segmentation algorithm is then applied to each local domain just as before. Thus, each sector receives the full focus of the cluster detection and iterative labelling process, thereby relieving the problems of non-local compatibility coefficients and (somewhat) cluster overlap. Now, many of the regions that are obtained by this process are arbitrarily split along the boundaries of adjacent sectors. Therefore, after the set of sectors has been segmented, a post-processing stage is applied to merge selected regions that were artificially split along sector boundaries. The merging process is based on the ability to decide statistically whether the union of two adjacent regions produces unimodal or bimodal distributions of feature values. This chapter will

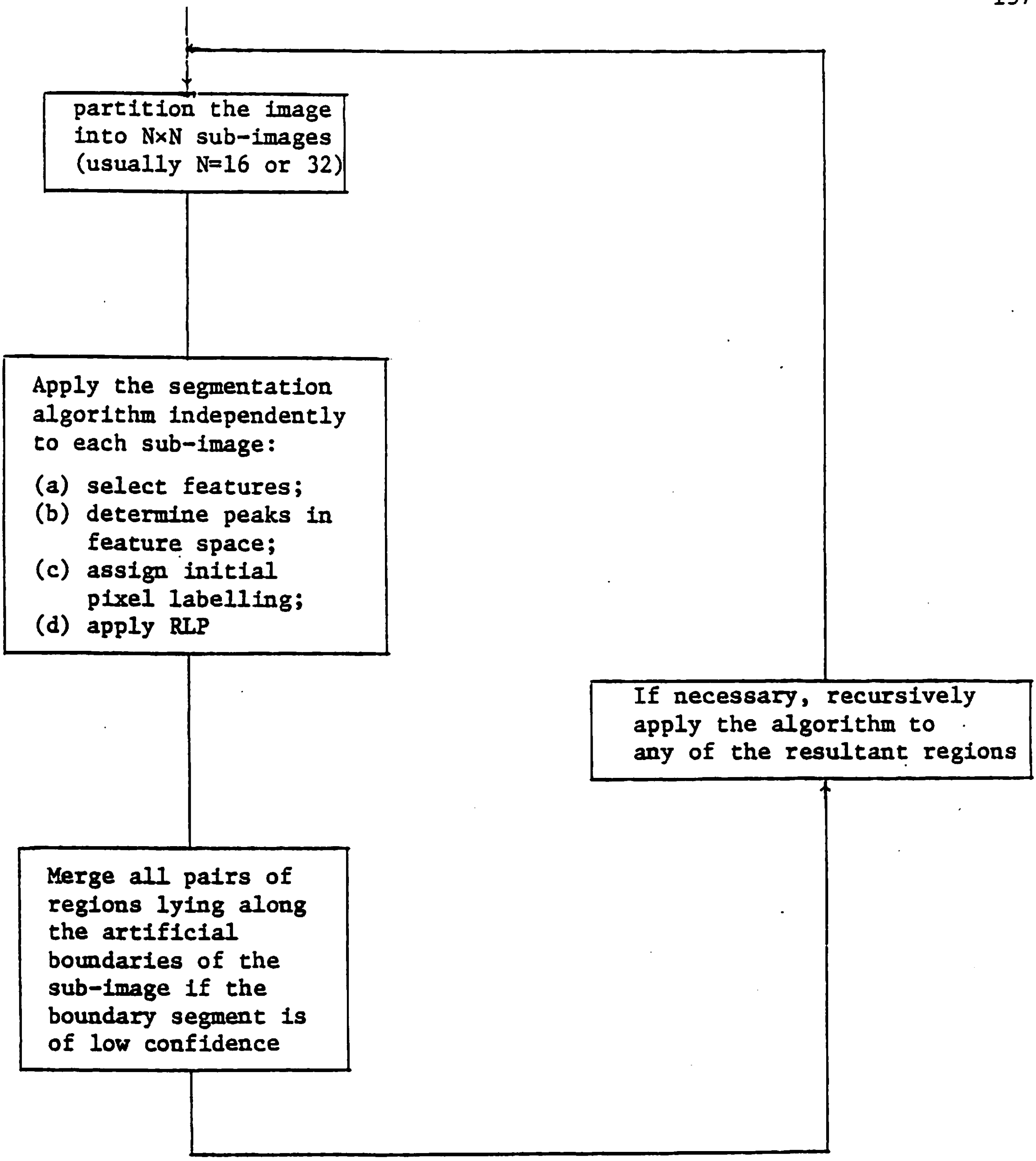

partition the image into N×N sub-images (usually N=16 or 32)
Apply the segmentation algorithm independently to each sub-image:
(a) select features;
(b) determine peaks in feature space;
(c) assign initial pixel labelling;
(d) apply RLP
Merge all pairs of regions lying along the artificial boundaries of the sub-image if the boundary segment is of low confidence
If necessary, recursively apply the algorithm to any of the resultant regions

Figure VI.1 Localized Segmentation Algorithm.

focus on the design issues of the localized segmentation algorithm, and will show some results based on artificial and natural scenes.

## VI.1 Design and Implementation Issues

### VI.1.1 Size of Sectors

The first step of the new localized algorithm requires partitioning the image into a set of smaller sectors. There are two basic concerns in selecting a sector size. The first is the elimination of the hidden cluster problem, so that all locally prominent regions can be associated with a unique cluster. Second, it is desirable for the sectors to be small enough so that any image structures that are present are "prominent" within the sector. This will allow the compatibility statistics to properly represent local activity. In the limit this could require sectors consisting of a very small number of pixels, but, of course, the sector size can not be reduced to an arbitrarily small size, since this would lose the ability to estimate the feature distribution by means of a histogram. In such a case peak selection and feature evaluation could be meaningless.

Our choice of partition size will be restricted to powers of two. Although this is not a rigid requirement, it facilitates the implementation of the algorithm in a parallel fashion in our processing cone [HAN76], where each sector is accessed and processes simultaneously. Sectors of size 16x16 and 32x32 were chosen because they were often sufficient to yield smooth histograms of reasonable appearance in both artificial and natural images.

The partitioning issue has an obvious weakness. Consider a sector with a visually distinct region that is easily detectable. If this region happens to extend slightly into one of the adjacent sectors, it is quite possible that those associated pixels will not generate a detectable peak in the histogram of the adjacent sector. This would mean that a portion of a clear region would be lost in the local segmentation due only to the artificial placement of sector boundaries.

To remedy this situation, each sector will be expanded so that it overlaps with each neighboring sector by 25% on each side. In the case of a 16x16 "inner" sector, it will be expanded to a 24x24 "outer" sector so that it overlaps each adjacent sector by 4 rows or columns. The assumption here is that any small protrusion into the inner sector will be sufficiently represented in the outer sector to be globally detectable in the outer sector histogram.

### VI.1.2 Segmentation

The segmentation of the partitioned image proceeds by independently applying the global algorithm to each sector. It was found that even with sectors as large as 32x32, the feature histograms sometimes were very jagged and the peak/cluster detection was somewhat difficult to analyze subjectively. Therefore, it was decided that the automatic peak selection criteria, when applied to such a small number of points, should be modified to allow more clusters. The justification for this decision, which potentially leads to region fragmentation, depends upon the effectiveness of the merging step to recover from a peak detection error. Hopefully, if an additional cluster label leads to the fragmentation of a region, then the local statistics of the region fragments should be very similar, and they will thus be reunited. Conversely, if the region fragments are significantly different, then they will remain separate as one would expect. The merging process thus provides the means of recovering from certain types of errors in the peak detection process, allowing the use of a less conservative cluster detection mechanism.

Notice that the system now has the ability to employ the feature histogram that is most appropriate for the sector under scrutiny. This will allow much finer discrimination of objects than the global approach permits. However, this flexibility makes the merging process slightly more cumbersome as will be shown in the next section.

### VI.1.3 Merging - Sewing Regions at the Seams

The final stage of the local segmentation algorithm requires the reuniting of the independently segmented sectors to form a continuous segmentation with boundaries only where they are actually indicated by the data. Prior to merging, the image consists of a set of uniquely labelled regions, some of which have been artifically broken into pieces by the partitioning. Thus, there will be vertical and horizontal lines which we will call "seams" that cut across certain regions. The seams are, of course, the artificial sector boundaries. The obvious approach to merging these regions is to base the process on exactly the information that would have produced single or multiple regions in the segmentation process -- the modality of the distribution across the regions under consideration. Consequently, the merging process requires the ability to detect whether a pair of adjacent regions form a unimodal or bimodal distribution.

One method for carrying out the merging process is to examine the histogram formed by the union of the data in the two distributions, using a slight variation of the peak selection algorithm. The goal here is to determine the presence (rather than the location) of either one, or more than one, clusters. If only one peak is detected, the distribution will be assumed to be unimodal and the boundary between the regions will be eliminated; otherwise, it will be left intact. This technique, although providing a lot of information, has the obvious drawback of requiring large amounts of storage for histograms -- one per region. Worse, due to the artifact of partitioning, the number of regions to be histogrammed is much greater than the number of regions one expects to find in the final segmentation.

A simpler technique, although possibly less reliable, uses a statistical measure of the two distributions in question. In a paper on bottom-up region analysis, Yakimovsky [YAK76] suggested using the following criterion for merging atomic regions:

$$C_{12} = V_0/V_1 * V_2$$

where

$V_0$ = the standard deviation across both regions 1 and 2

$V_1$ = the standard deviation measured across region 1

$V_2$ = the standard deviation measured across region 2.

$C_{12}$ can be interpreted as the confidence that regions $R_1$ and $R_2$ are separate regions. For our purposes, when $C_{12} < \theta$ we will consider that $R_1$ and $R_2$ can be merged.

Notice that if region 1 and region 2 have the same distribution ($\mu_1 = \mu_2 = \mu_0$ and $\sigma_1 = \sigma_2 = \sigma_0$), the output of this measure is $1/V_0$, which is the minimum value that this function can take on. As the means of the regions become further apart, $C_{12}$ can get arbitrarily large since the standard deviation of the joint distribution will be larger than that of the individual distributions.

There are two problems with using this function. The first is that $C_{12}$ depends on $V_0$, which means there is not a unique baseline for comparison. To remedy this, we have squared the numerator yielding:

$$C_{12} = V_0^2/V_1 * V_2$$

This effectively normalizes the function, so that $C_{12}$ is 1 when $V_0 = V_1 = V_2$.

The second problem with the original measure arises whenever there is a large difference in the size of the two regions. In this event, $C_{12}$ will be approximately equal to the standard deviation of the larger region, say region 1. Then

$$C_{12} = V_1/V_1 * V_2 = 1/V_2$$

which is generally much less than 1. Thus, a size difference biases the function toward merging. To remedy this, the computation of $V_0$ can be changed as follows:

$$V_0 = (\Sigma(x_i - \mu_1)^2/m + \Sigma(x_i - \mu_2)^2/n)$$

where m and n are the sample sizes of the two regions. This treats the two distributions as if they were of equal size. The final measure is therefore:

$$C_{12} = V_0{}^2/V_1 * V_2$$

where $V_0$ is computed as above.

## VI.2 Examples of Local Segmentation

We will now demonstrate the effectiveness of the local partitioning algorithm over the global algorithm. First, let us examine some of the examples in the previous chapters and then apply the algorithm to new, more difficult cases. In each case, probabilitic relaxation with conditional probabilities as compatibility coefficients and a 5-adjacency neighborhood are employed.

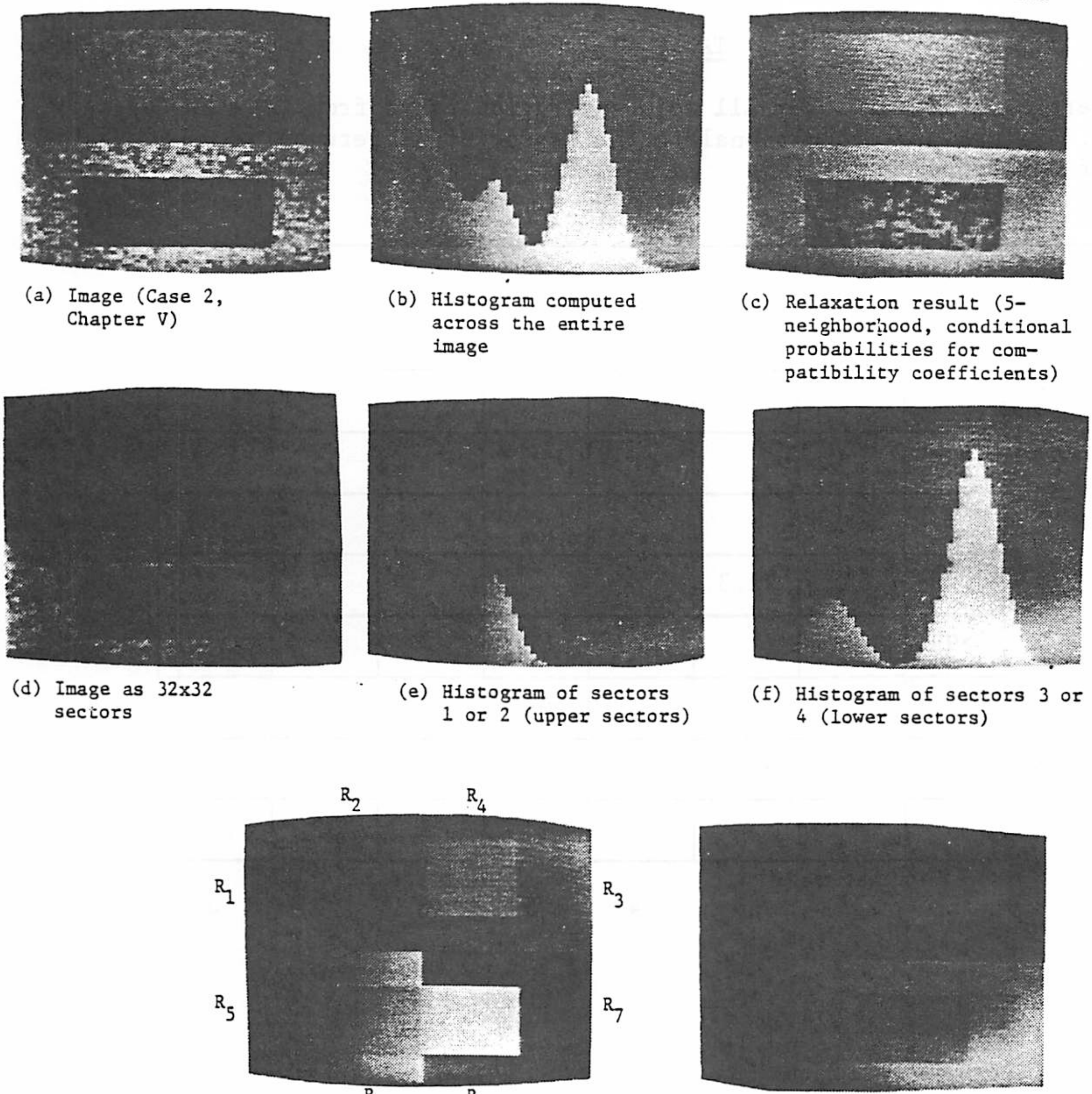
### VI.2.1 Case 2, Chapter V: Recovery from Fragmentation

This example is the same as Case 2 from Chapter V; Figure

VI.2a,b, and c recapitulate the global segmentation. Recall that the distribution of object 4 was hidden by those of objects 1 and 2. The global segmentation fragmented object 4 into many small pieces. Let us now look at the performance of the local segmentation. When the image is split into 32x32 sectors (Figure VI.2d), the local histograms clearly reveal all the relevant peaks. The histograms of the top 2 sectors are similar in appearance since the noise statistics are basically the same in each sector; this is also true for the bottom 2 sectors. The pixels in each sub-image are initially classified, and relaxation labelling yields the result shown in Figure VI.2g. Finally, Figure VI.2h shows the result of applying the merging process across all of the adjacent regions. Table VI.1 shows the merging statistics for all pairs of adjacent regions that touch the artificial sector boundaries. A zero entry indicates non-adjacency. Notice that although there is a wide range in the merging statistics, the values tend to cluster around values less than 2 and greater than 20. The threshold for merging is set to 2 and applied across the image. This last step yields the result shown in Figure VI.2h. The error rate in $O_4$ has been reduced from 30% in the global analysis to 0%. Notice that the merging threshold would have to be increased tenfold before any adverse remergng would take place.

### VI.2.2 Case 5, Chapter V

This example is the same as Case 5 from Chapter V. Figure VI.3 demonstrates again, that with the disappearance of the mislabelled



(a) Image (Case 2, Chapter V)

(b) Histogram computed across the entire image

(c) Relaxation result (5-neighborhood, conditional probabilities for compatibility coefficients)

(d) Image as 32x32 sectors

(e) Histogram of sectors 1 or 2 (upper sectors)

(f) Histogram of sectors 3 or 4 (lower sectors)

(g) Composite showing the 4 sectors segmented independently, each gray level in a sector represents a unique label.

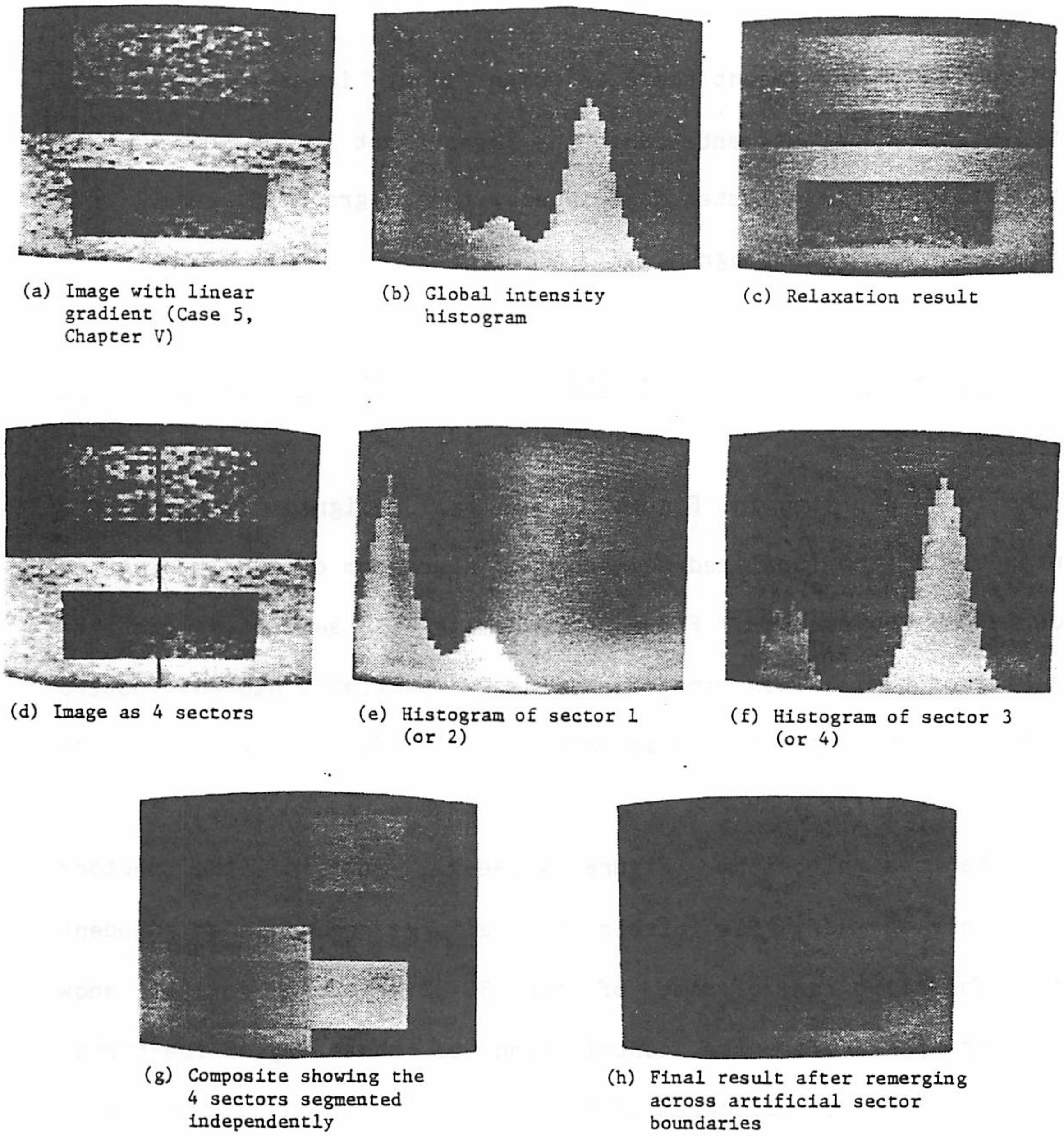
(h) Final result after merging across artificial sector boundaries

Figure VI.2 Localized Segmentation of Case 2, Chapter V: An image is first broken into 32x32 sectors. Each sector is then independently segmented. Finally, regions that were artificially broken at the sector boundaries are merged if their distributions are similar.

## TABLE VI.1

Merging statistics for all adjacent region pairs from Case 2, Chapter V. Values are proportional to the degree of difference between a pair of regions.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | 19.9 | | | | | | | |
| 3 | 1 | | | | | | | |
| 4 | | 1 | 22.3 | | | | | |
| 5 | >99 | | | | | | | |
| 6 | | | | | >99 | | | |
| 7 | | | >99 | | 1 | | | |
| 8 | | | | | | 1 | >99 | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |



(a) Image with linear gradient (Case 5, Chapter V)

(b) Global intensity histogram

(c) Relaxation result

(d) Image as 4 sectors

(e) Histogram of sector 1 (or 2)

(f) Histogram of sector 3 (or 4)

(g) Composite showing the 4 sectors segmented independently

(h) Final result after remerging across artificial sector boundaries

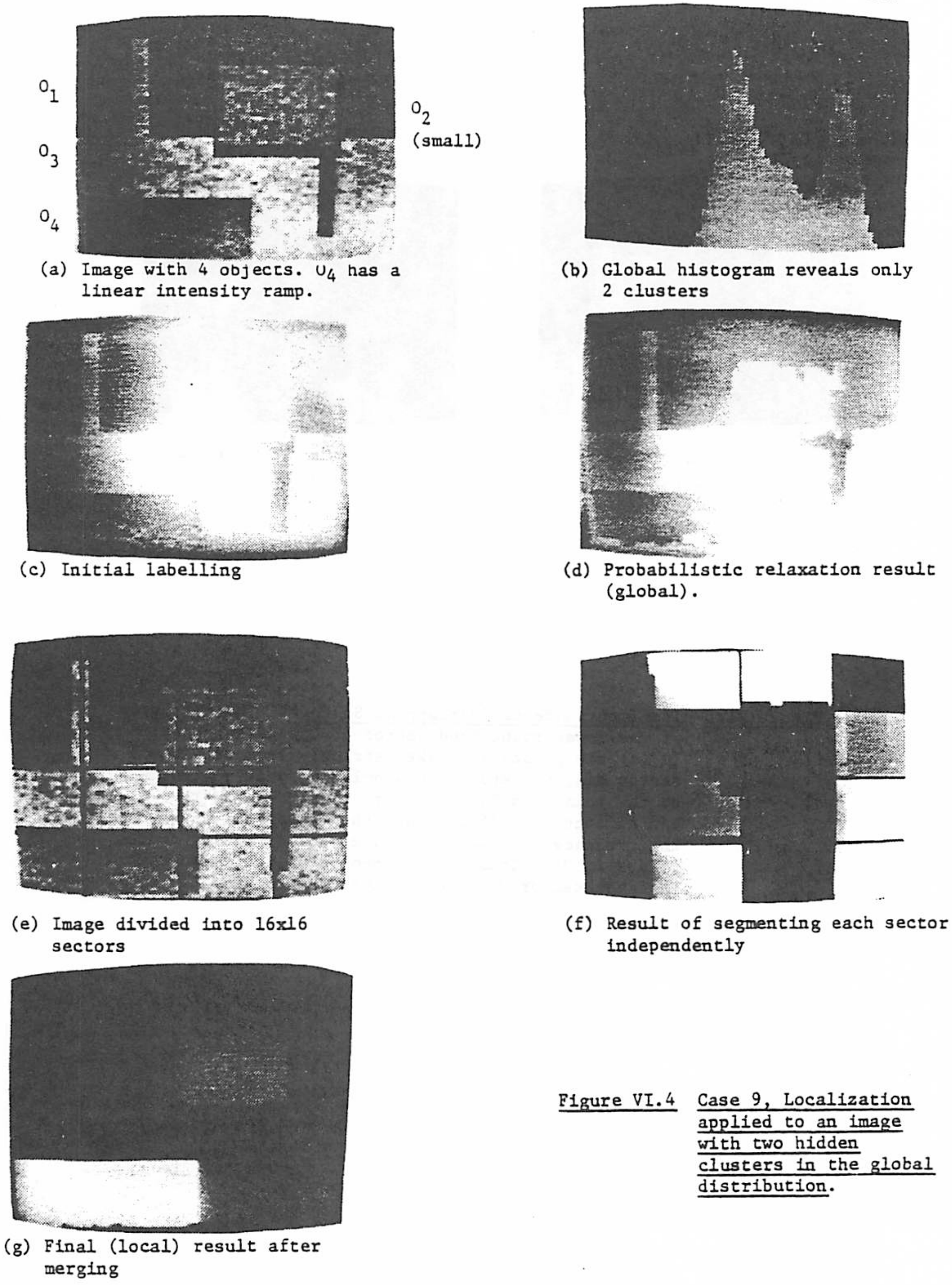
Figure VI.3 Localized Segmentation Applied to the Image in Case 5, Chapter V.

bottom band of the gradient there is significant improvement in the output of the local segmentation. The improvement is of course due to the visibility of the cluster associated with the gradient, which was hidden in the global histogram.

### VI.2.3 Case 9, Demonstrate the Effectiveness of Overlapped Sector Boundaries

The image depicted in Figure VI.4a was designed to show the importance in the localized segmentation algorithm of overlapping the boundaries of the sectors. First, the image is segmented via the global algorithm. This image contains two partially hidden clusters and therefore the segmentation is particularly bad, as shown in Figure VI.4d.

In the local algorithm (Figure VI.4e-g), many of the sectors contain only a very few pixels from a large region in an adjacent sector. The histograms of some of the 32x32 sectors do not show significant peaks for the contribution of those region fragments. Figure VI.5 shows the histogram of the lower left-hand sector, with and without the "extra" points. The peak in Figure VI.5a corresponds to pixels from $O_3$, while the additional peak in Figure VI.5b corresponds to the band of pixels from $O_1$. To remedy this, all sectors are extended by 25% in each direction, yielding 48x48 domains. In general, the augmented histograms reveal the presence of the distributions of the poorly represented regions. Without these augmented histograms, some of the sectors would be incorrectly



(a) Image with 4 objects. $O_4$ has a linear intensity ramp.

(b) Global histogram reveals only 2 clusters

(c) Initial labelling

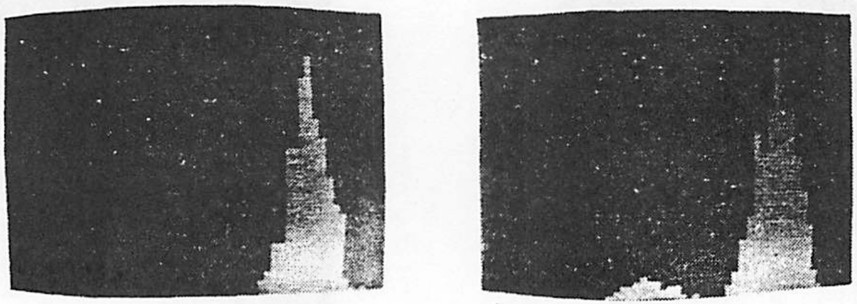
(d) Probabilistic relaxation result (global).

(e) Image divided into 16x16 sectors

(f) Result of segmenting each sector independently

(g) Final (local) result after merging

Figure VI.4 Case 9, Localization applied to an image with two hidden clusters in the global distribution.



(a) Histogram of lower right sector

(b) Histogram of expanded sector

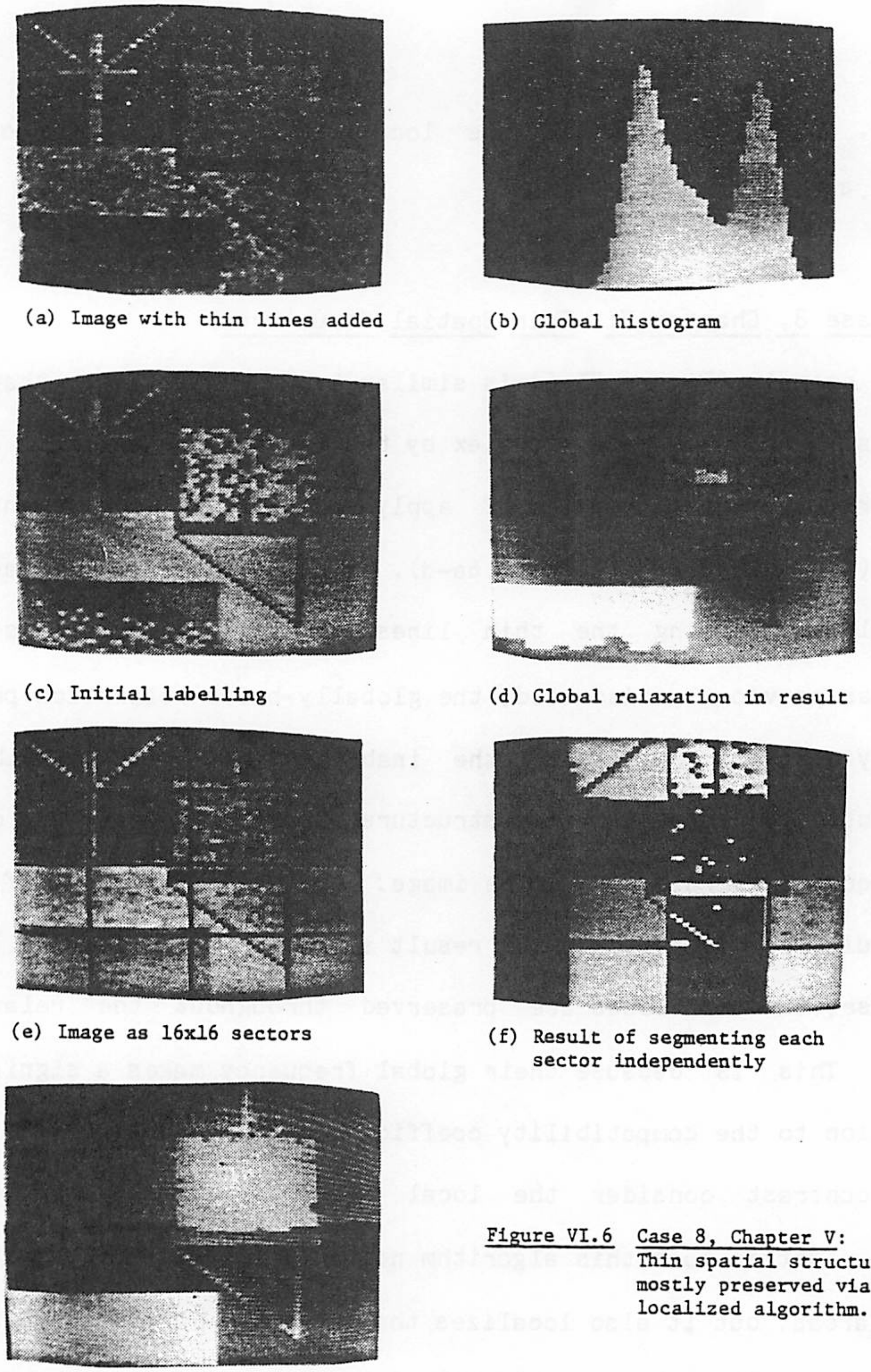
Figure VI.5 The Importance of Overlapping Sectors: The histogram of the lower right hand sector of Case 9 is shown. In (a) the points are taken strictly from within the sector and, therefore, show only one peak corresponding to the points in $O_3$. However, by increasing the size of the sector by 25 percent, the histogram (b) reveals the presence of a second cluster corresponding to the points in $O_1$. Thus, the augmented histogram ensures that the sector will be segmented correctly.

segmented. The final result of the local segmentation is shown in Figure VI.4g.

### VI.2.4 Case 8, Chapter V: Thin Spatial Structures

This example (Figure VI.6) is similar to Case 9 except that the image has been made more complex by the introduction of thin lines. Let us first review the result of applying the global segmentation process (Figure V.8 or Figure VI.6a-d). Notice that initially most of the pixels comprising the thin lines are correctly classified. However, as previously discussed, the globally-based relaxation process ultimately destroys them due to the inability of the compatibility coefficients to preserve thin structures whose feature values do not occur frequently across the entire image. This segmentation of this case should be compared to the result shown in Figure IV.8b. In the latter case, the thin lines are preserved throughout the relaxation process. This is because their global frequency makes a significant contribution to the compatibility coefficients.

By contrast consider the local segmentation results (Figure VI.6e-g). Notice that this algorithm not only localizes the histogram to small areas, but it also localizes the range of the compatibility coefficients. Thus the new algorithm is capable of focusing on local, orientation-dependent co-occurences of label pairs as well as seeing local peaks. The final results are again a tremendous improvement--in both of these respects--over the global segmentation.



(a) Image with thin lines added

(b) Global histogram

(c) Initial labelling

(d) Global relaxation in result

(e) Image as 16x16 sectors

(f) Result of segmenting each sector independently

(g) Final merged result.

Figure VI.6 Case 8, Chapter V: Thin spatial structures are mostly preserved via the localized algorithm.

### VI.2.5 Localized Segmentation Applied to Our Example Outdoor Scene

Finally, in Figure VI.7, we return to the natural, outdoor image that was presented at the end of Chapter IV. The global segmentation (Figure VI.7c) yielded poor results in the following areas:

(1) the roof and right-hand tree were inseparable;

(2) the left-hand tree was severely fragmented;

(3) the house roof and garage roof were partially merged.

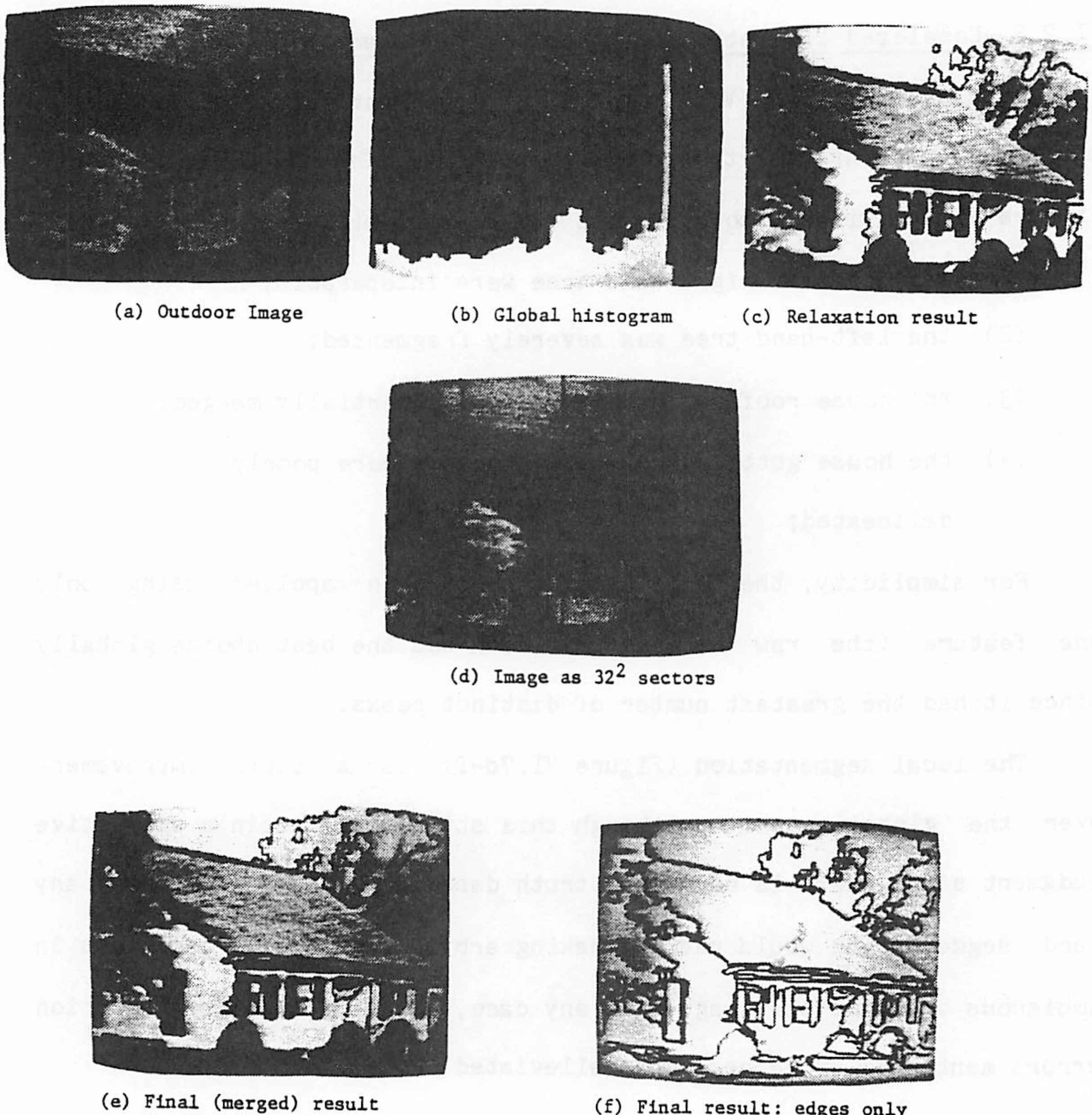
(4) the house gutter and window shutters were poorly delineated;

For simplicity, the local algorithm has been applied using only one feature (the raw blue data) which was the best choice globally since it had the greatest number of distinct peaks.

The local segmentation (Figure VI.7d-f) is a clear improvement over the global result although this still must remain a subjective judgment since there is no ground truth data for this image and any hand segmentation would require making arbitrary boundary decisions in ambiguous areas of the image. In any case, most of the segmentation errors mentioned above have been alleviated.

## VI.3 Conclusion

This chapter has shown that a dramatic improvement in the quality of a region analyser can be obtained by localizing the focus of the system. The new paradigm consists of artifically partitioning the image, segmenting each partition, and finally, merging regions that



(a) Outdoor Image

(b) Global histogram

(c) Relaxation result

(d) Image as $32^2$ sectors

(e) Final (merged) result displayed as edges over the date

(f) Final result: edges only

Figure VI.7: Localized segmentation applied to our example natural outdoor scene.

were artificially broken at the sub-image boundaries. A simple, apparently robust, merging statistic was developed for detecting whether two regions are unimodal or bimodal. Results for both artificial and natural scenes showed dramatic improvements.

It should be emphasized that the merging process involves a threshold operation that may not always produce results that are globally desirable. There is less risk of making an error in merging when the merging operation is restricted to regions that are broken at known sector boundaries. In those cases, one may assume that the target region <u>must</u> have been broken somewhere along the sector boundary (since they are arbitrary) and it might be sufficient to simply find the region on one side that is most like it on the other. However, one may want to apply the merge test in general, to all pairs of adjacent regions, as a post-processing check for fragmentation. In this case, the risk of merging regions that are better left alone increases.

It is interesting to note that the local algorithm, although requiring apparently much more overhead than the global algorithm, does not actually take much longer to compute . The reason is that each local segmentation step is shorter not only because there are fewer pixels, but also because fewer iterations are required to reach convergence. In the global algorithm, the label probabilities of all pixels are updated until the last pixel converges. In the local algorithm, relatively unambiguous sub-images can converge at a rate independent of other more ambiguous sub-images.

# CHAPTER VII

# CONCLUSIONS

This thesis has evaluated the results of various segmentation algorithms applied to both a natural scene and computer-generated test images. It appears that carefully constructed test images provide more insight into the capabilities and limitations of these algorithms than the natural scene. The structure of the information in the test images was chosen to be particularly difficult for these algorithms in an effort to demonstrate both their capabilities and limitations. Such results, coupled with natural scene segmentations, allow insights that otherwise would not have been available. Let us now review some of the major findings of this research.

## VII.1 Histograms

It was shown that segmentation by histogram clustering/pixel labelling is a technique that is quite prone to error. Distributions of objects in an image tend to overlap by varying degrees, with the result that some pixels cannot be accurately classified. A set of test images was examined that showed how certain arbitrary image properties can greatly affect the quality of the segmentation. These properties include the spatial arrangement of objects, the spatial distribution of pixel values in an object, and the shape of an object.

## VII.2 Relaxation and Feature Space

Next, a more complex segmentation algorithm that greatly improved the region analysis was presented. Instead of simply assigning a discrete label to each pixel, a probabilistic labelling scheme was introduced in which the label of a pixel is encoded by its relative location in feature space. Then, a probabilistic relaxation labelling process was applied to attain locally consistent labellings. Again, the use of test images proved fruitful to explore parameters such as the choice of neighborhood configuration, probabilistic relaxation vs. plurality relaxation, and the computation of the compatibility coefficients.

In areas of an image that lack fragile spatial structures, it was shown that any of the relaxation techniques improved the pixel classifications. However, widely varying results were found in areas that display fine structures. In the latter case, all of the techniques were shown to destroy fine structure when the center pixel was excluded from its own neighborhood. Including the center pixel as its own neighbor has the effect of adding "self-belief" to the RLP by increasing support from like labels in the neighborhood.

The definition of the compatibility coefficients also had a pronounced effect on the classification error rates. Three variant formulations were explored:

1. no compatibilities, as in the plurality relaxation scheme;
2. "simple" compatibilities $(r(i,\alpha,j,\beta) = 1$, $r(i,\alpha,\beta,j) = -1$;

and

3. compatibilities as conditional probabilities of label pairs at particular orientations.

The first scheme does not use any information about the particular image being explored and, consequently, does a poor job of preserving fine details. The second scheme does not use image-specific information, but it does use meta-information about images; namely that one may expect to find -- and should promote the likelihood of -- adjacent labels of the <u>same</u> type. This assumption is valid in coarsely structured objects, but it is not adequate in finely structured objects. In the latter, there is a large percentage of boundary area and thus <u>dissimilar</u> label adjacencies are expected. Accordingly, the second scheme does not yield very good results in areas of fine detail. Both schemes 1 and 2 quickly reach a minimum error rate, only to diverge drastically at later iterations.

The third scheme is the most complex and the only one that uses image-specific information. Here, we are attempting to capture label dependencies using the framework of conditional probabilities. Upon careful examination, we were able to show how specific structures in the image were translated into strong and weak compatibility coefficients in the compatibility tables. This scheme yielded the best results when applied to the test image. Moreover, it displayed the least divergent behavior, reaching a minimum error rate in a few iterations and staying there over time.

In addition to exploring relaxation schemes, some effort was put into exploring clustering techniques in one- and two-dimensional color feature spaces. We found that opponent colors tended to heighten color differences, yielding improved cluster detection. Multi-dimensional spaces were found to yield more clusters than one-dimensional spaces, and thus give better sensitivity to image characteristics without requiring costly recursive steps in the segmentation process.

## VII.3 Problems with Global Segmentation

Another set of test images was explored which showed that recovery from classification errors via relaxation is not always successful. Errors persisted when the initial classification of pixels in a region contained: (1) a dense population of errors (cases 2 and 3), or (2) errors that were spatially correlated (cases 4-7). In these cases, the RLP tended to maintain the errors since there was significant local support for them.

In addition, it was shown that the compatibility coefficients, because of their global nature, often biased the RLP in an undesirable manner. Thus, for instance, changing the size (case 6) or the mean (case 7) of certain objects affected the segmentation of other objects that were spatially distant.

Finally, it was shown that thin spatial structures (case 8) could be suppressed during relaxation even if they were initially segmented correctly. Again, this was due to the lack of sufficient global information to support the existence of these structures. Therefore, in lieu of sufficient compatibility information, the geometry of the neighborhood configuration dictated their segmentation.

## VII.4 Partitioning Prior to Segmentation

Analysis of segmentation results on test images led to a new formulation of the segmentation algorithm based on the use of sub-images (sectors). The idea here is to partition the image into sectors that are small enough to reveal local clusters and local structures yet large enough to be statistically meaningful. By artificially breaking the image into small units, the problems of cluster overlap and insensitive compatibility coefficients were overcome to a great extent. The new paradigm thus minimizes non-local side effects. After segmentation of each of the sectors, a simple merging technique is applied so that regions that were artificially broken at the sector boundaries can be remerged to form whole regions. This technique was shown to be robust and did not leave any obvious region fragments.

## VII.5 Future Work

Let us consider a few areas that should be further explored. First, the technique used to initially label pixels could be easily improved. The use of the Euclidean distance of a point to a cluster center is inadequate since it does not take into account the shape of the clusters. It is very likely that there are points in a histogram that are close to one cluster yet which lie in the distribution of another. In such cases, the initial labelling of a pixel will be in error. To remedy this, one could assume that regions have normal distributions and then estimate the mean and variance of each cluster. Then, the likelihood of any point belonging to any of the clusters could be computed statistically.

Second, it seems clear that more work needs to be done on the formulation of the relaxation process. Peleg [PEL79] and Zucker [ZUC79] have made some headway into characterizing RLPs and are supplying non-heuristic methods for their derivation. The tendency in the current research is toward the use of hierarchical relaxation schemes and those that use compatibility coefficients which are better approximations to the spatial dependencies that appear in the image [RIS78]. Here, joint conditional probabilities of the set of labels in a neighborhood can provide more effective updating criteria suggested by a Bayesian probability framework. The limitation of this approach is that m labels of n neighborhood pixels requires estimation and application of nxm compatibility coefficients.

Finally, we feel that a larger set of test images should be developed. In particular, effects such as blurring (mixed pixels) and complex texturing should be incorporated into the images. Moreover, it would appear that these kinds of images should be constructed as a collaborative effort of the image understanding community and made available to those involved in applying their techniques to natural scenes. We may then further understand the areas of difficulty for current algorithms and to address these problems in a structured way.

# BIBLIOGRAPHY

[BAR78] Barrow, H.G. and Tenenbaum, J.M., "Recovering Intrinsic Scene Characteristics from Images," Technical Note 157, SRI International, April, 1978.

[BIN57] Bingley, F.S., Color Vision and Colorimetry, Television Engineering Handbook, G.D. Fink, Ed., McGraw-Hill, New York, 1957.

[BRI70] Brice, C.R. and Fennema, C.L., "Scene Analysis Using Regions," Artificial Intelligence, 1, 205-226, 1970.

[CHE78] Chen, P.C. and Pavlidis, T., "Segmentation by Texture Using A Co-occurrence Matrix and a Split-end-Merge Algorithm," Fourth IJCPR, Kyoto, Japan, November, 1978.

[CHO71] Chow, C.K. and Kaneko, T., "Boundary Detection of Radiographic Images by a Threshold Method," Proc. IFIP 71, Ta-7, 130, 1971.

[COL77] Coleman, G.B., "Image Segmentation by Clustering," USCIPI Report 750, Image Processing Institute, University of Southern California, July, 1977.

[COR70] Cornsweet, T.N., Visual Perception, Academic Press, New York, 1970.

[EHR76] Ehrich, R.W. and Foith, J.P., "Representation of Random Waveforms by Relative Trees," IEEE Trans. Computers, Vol. C-25, 725-736, 1976.

[EKL78] Eklundh, J.O., Yamamoto, H., and Rosenfeld, A., "Relaxation Methods in Multispectral Pixel Classification," Technical Report-662, Computer Science Center, University of Maryland, July, 1978.

[FEL74] Feldman, J.A. and Yakimovsky, Y., "Decision Theory and Artificial Intelligence 1: A Semantics-based Region Analyzer," Artificial Intelligence, 5, 349-371, 1974.

[FRE76] Freuder, E.C., "Affinity: A Relative Approach to Region Finding," Computer Graphics and Image Processing, 5, 254-264, 1976.

[HAN74] Hanson, A. and Riseman, E., "Preprocessing Cones: A Computational Structure for Scene Analysis," COINS Technical Report 74C-1, University of Massachusetts, September, 1974.

[HAN75] Hanson, A.R. and Riseman, E.M., "The Design of a Semantically Directed Vision Processor," Technical Report 75C-1, Computer and Information Science Department, University of Massachusetts, Amherst, February, 1975.

[HAN75] Hanson, A.R., Riseman, E.M., and Nagin, P., "Region Growing in Textured Outdoor Scenes," Proc. of 3rd Milwaukee Symposium on Automated Computation and Control, 407-417, 1975.

[HAN76] Hanson, A.R. and Riseman, E.M., "A Progress Report on VISIONS: Representation and Control in the Construction of Visual Models," COINS Technical Report 76-9, University of Massachusetts, Amherst, 1976.

[HAN78] Hanson, A.R. and Riseman, E.M., "VISIONS: A Computer System for Interpreting Scenes," in Computer Vision Systems, (Hanson and Riseman, Eds.), Academic Press, New York, 1978.

[HAR78] Haralick, R.M., "Statistical and Structural Aproaches to Texture," Fourth IJCPR, Kyoto, Japan, November, 1978.

[HOR78] Horowitz, S.L. and Pavlidis, T., "A Graph-Theoretic Approach to Picture Processing," Computer Graphics and Image Processing, 7, 282-291, 1978.

[KEL71] Kelly, M.D., "Edge Detection in Pictures by Computer Using Planning," Machine Intelligence, 6, 379-409, 1971.

[KEN76] Kender, J.R., "Saturation, Hue and Normalized Color: Calculation, Digitization Effects, and Use," Technical Report, Department of Computer Science, Carnegie-Mellon University, November, 1976.

[NAG77] Nagin, P.A., Hanson, A.R. and Riseman, E.M., "Region Extraction and Description Through Planning," COINS Technical Report 77-8, University of Massachusetts, Amherst, May, 1977.

[NAG78] Nagin, P.A., "Segmentation Using Spatial Context and Feature Space Cluster Labels," IJCPR, Chicago, Illinois, May, 1978.

[OHL75] Ohlander, R., "Analysis of Natural Scenes," Ph.D. Thesis Carnegie-Mellon University, April, 1975.

[PRA78] Pratt, W.K., Digital Image Processing, John Wiley and Sons, New York, 1978.

[PEL78] Peleg, S., and Rosenfeld, A., "Determining Compatibility Coefficients for Curve Enhancement Relaxation Processes," IEEE Transactions on Systems, Man, and Cybernetics, SMC-8, 7, July, 1978.

[PRE72] Preparata, F.P. and Ray, S.R., "An Approach to Artificial Non Symbolic Cognition," Information Science, 4, 65-66, 1972.

[PRE66] Prewitt, J.S.M. and Mendelsohn, M.L., "The Analysis of Cell Images," Annual New York Acad. Sci., 128, 1035-1053, 1966.

[PRI77] Price, K. and Reddy, R., "Change Detection and Analysis in Multispectral Images," Proc. of 5th International Joint Conference on Artificial Intelligence, Cambridge, 619-625, 1977.

[RIS74] Riseman, E.M. and Hanson, A.R., "The Design of a Semantically Directed Vision Processor," COINS Technical Report 74C-1, University of Massachusetts, January, 1974.

[RIS77] Riseman, E.M. and Arbib, M.A., "Computational Techniques in the Visual Segmentation of Static Scenes," Computer Graphics and Image Processing, 6, 221-276, 1977.

[RIS78] Riseman, E.M., "Mechanisms for Parallel Processing of Local Contexts with Uncertainty," Informal Working Paper, SRI, December, 1978.

[ROS76] Rosenfeld, R.A., Hummel, R.A., and Zucker, S.W., "Scene Labelling by Relaxation Operations," IEEE Transactions on Systems, Man, and Cybernetics, 6, 420-433, 1976.

[ROS77] Rosenfeld, A. and Davis, L.S., "Iterative Histogram Modification," TR-519, Computer Science Center, University of Maryland, April, 1977.

[SLO75] Sloan, K.R. and Bajcsy, R., "A Computational Structure for Color Perception," Proceedings of ACM75, Minneapolis, Minnesota, 1975.

[TAN75] Tanimoto, S.L. and Pavlidis, T., "A Hierarchical Data Structure for Picture Processing," Computer Graphics and Image Processing, 4, 104-119, 1975.

[TEN74] Tenenbaum, J.M., Garvey, T.D., Weyl, S., and Wolf, H.D., "An Interactive Facility for Scene Analysis Research," SRI Technical Note 87, Artificial Intelligence Center, Stanford Research Institute, 1974.

[TEN76] Tenenbaum, J.M. and Barrow, H.G., "Experiments in Interpretation Guided Segmentation," Artificial Intelligence, 8, 241-274, 1976.

[TSU73] Tsuji, S. and Tomita, F., "A Structural Analyzer for a Class of Textures," Computer Graphics and Image Processing, 2, 216-231, 1973.

[UHR72] Uhr, L., "Layered `Recognition Cone' Networks that Preprocess, Classify, and Describe," IEEE Transactions on Computers, C-21, 758-768, 1972.

[WAL75] Waltz, D.L., "Understanding Line Drawings of Scenes with Shadows," in The Psychology of Computer Vision, (P.H. Winston, Ed), McGraw-Hill, New York, 1975.

[WAT74] Watanabe, S., "An Automated Apparatus for Cancer Prescreening: CYBEST," Computer Graphics and Image Processing, 3, 350-358, 1974.

[WES74] Weszka, J.S., Nagel, R.N., and Rosenfeld, A., "A Threshold Selection Technique," IEEE Transactions on Computers, C-23, 1322-1326, 1974.

[WES78] Weszka, J.S., "A Survey of Threshold Selection Techniques," Computer Graphics and Image Processing, 7, 259-265, 1978.

[ZUC75] Zucker, S.W., Rosenfeld, A., and Davis, L.S., "Picture Segmentation by Texture Discrimination," IEEE Transactions on Computers, C-24, 1228-1232, 1975.

[ZUC76] Zucker, S.W., "Relaxation Labelling and the Reduction of Local Ambiguities," Proc. of 3rd International Joint Conference on Pattern Recognition, San Diego, 1976.

[ZUC76] Zucker, S.W., "Region Growing: Childhood and Adolescence," Computer Graphics and Image Processing, 5, 382-399, 1976.

[ZUC78] Zucker, S.W. and Mohammed, J.L., "Analysis of Probabilistic Relaxation Labelling Processes," IJCPR, Chicago, Illinois, May, 1978.

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
COINS TR 79-15

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
STUDIES IN IMAGE SEGMENTATION ALGORITHMS BASED ON HISTOGRAM CLUSTERING AND RELAXATION

5. TYPE OF REPORT & PERIOD COVERED
INTERIM

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Paul A. Nagin

8. CONTRACT OR GRANT NUMBER(s)
ONR N00014-75-C-0459

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Computer and Information Science Department
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
9/79

13. NUMBER OF PAGES
180

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

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
image processing
segmentation evaluation
histogram clustering
relaxation labelling processes
global/local analyses

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
The research in this thesis has focussed upon the algorithms and structures that are sufficient to generate an accurate description of the information contained in a relatively complex class of digitized images. This aspect of machine vision is often referred to as "low-level" vision or segmentation, and usually includes those processes which function close to the sensory data. The bulk of this thesis devotes itself to the exploration of

some of the problems typically encountered in segmentation. In addition, a new and robust algorithm is presented that avoids most of these problems.

The analysis is carried out through the use of a series of computer-generated test images with known characteristics. Segmentation algorithms of varying degrees of complexity are applied to each image and their performance is carefully evaluated. It will be shown that even the most sophisticated algorithms that are currently in use often perform poorly when confronted with certain apparently simple images. In particular, it is shown that techniques which rely on histogram clustering often generate gross segmentation errors due to overlap in the distributions of the individual objects in a scene. Moreover, the relaxation processes used to correct these errors are themselves prone to errors, but of a different kind. Here, we show that the globally computed compatibility functions are inadequate to preserve image structure, even in some surprisingly simple images.

Both techniques, clustering and relaxation, fail because they are based on information which is too global to be effective in complex scenes. Clustering fails because most algorithms do not take into account the spatial feature information contained in the image. Relaxation-type algorithms take the spatial content into account by utilizing global information applied to local neighborhoods. However, global compatibility functions very often fail to resolve local image structure. This implies that improvements in performance might be obtained by localizing the algorithm to sub-images of the original image. In fact, a dramatic improvement in performance is obtained when this is done. Each sub-image is defined to be small enough so that the distributions of distinct visual elements are revealed as distinct histogram clusters. Moreover, the compatibility coefficients are measured over a sufficiently small area so that their characterization of the local image structure is not diluted by global effects. After segmenting each sub-image, a merging algorithm is applied so that regions that have been artificially split at sub-image boundaries can be sewn together to form the final segmentation.