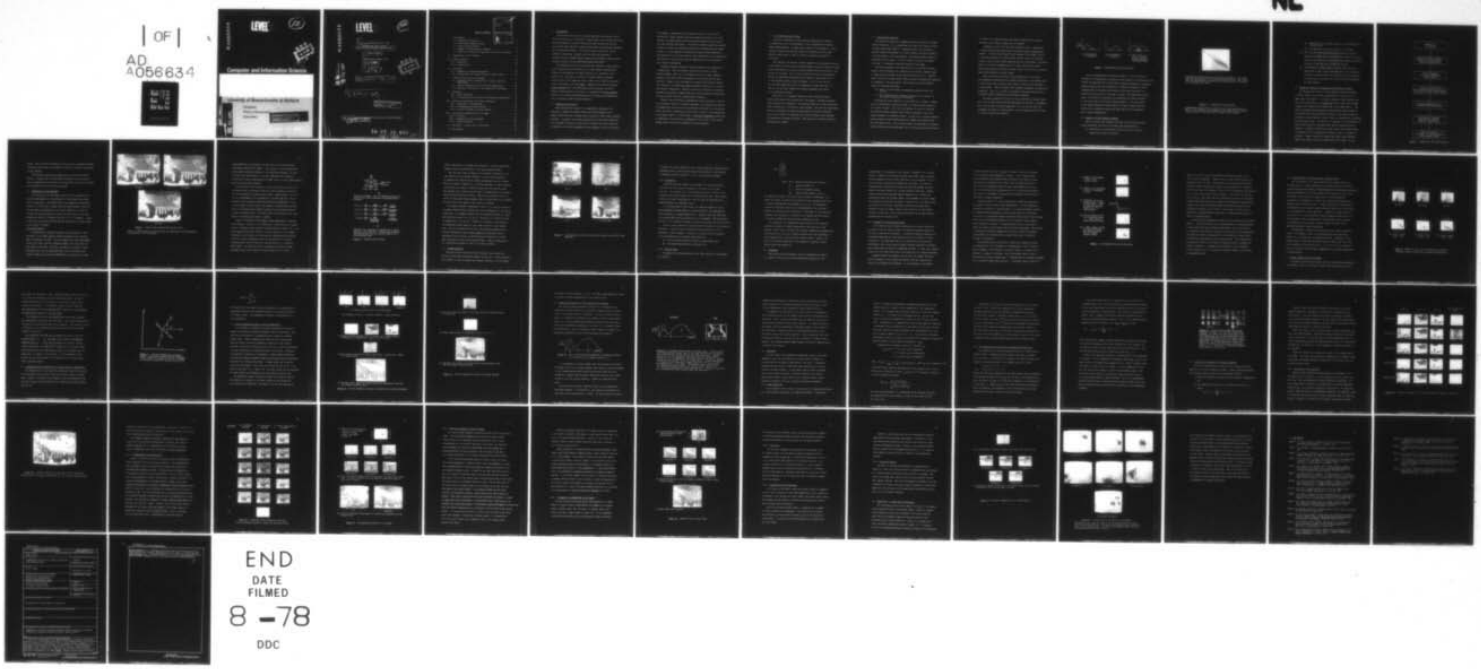
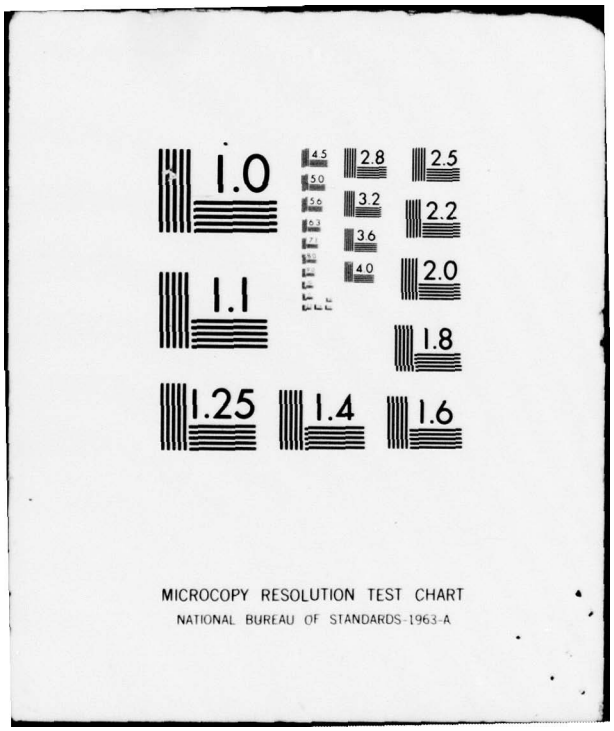
AD-A056 634 MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/G 20/6
SEGMENTATION USING SPATIAL CONTEXT AND FEATURE SPACE CLUSTER LA--ETC(U)
MAY 78 P A NAGIN
UNCLASSIFIED COINS-TR-78-8 N00014-75-C-0459
NL



END
DATE FILMED
8 -78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

12

Computer and Information Science

University of Massachusetts at Amherst

Computers

Theory of Computation

Cybernetics

LEVEL II

12

SEGMENTATION USING SPATIAL CONTEXT AND FEATURE SPACE CLUSTER LABELS.

Paul A. Nagin

COINS Technical Report 78-8

May 78

56p.

COINS-TR-78-8

Interim rept.

This document has been approved for public release and sale; its distribution is unlimited.

[1]This research has been supported by the Office of Naval Research under Grant N00014-75-C-0459.

TABLE OF CONTENTS

I. INTRODUCTION

The focus of this paper is on image segmentation processes, collectively referred to as a "low-level" vision system. The low-level processes have been applied to various unconstrained image domains and function to partition large amounts of sensory visual data into organized components with associated attributes. This output forms the basis for further (semantic) processing. Together, the low-level and high-level processes form the VISIONS (Visual Integration by Semantic Interpretation Of Natural Scenes) system [HAN74,HAN76,HAN78,RIS74,RIS77].

The programs which will be discussed here transform a large spatial array of pixels (picture elements) into a more compact representation through the exploitation of visual features, e.g., intensity, color, texture, etc. The goal is to detect a relative feature invariance across an area of the image and then to label all the pixels in any such area as belonging to the same region. Regions can be detected through global analyses which find interesting areas by ignoring the local textural configurations of the data, in conjunction with local analyses which act as a fine-tuning mechanism both to resolve global ambiguities and to accurately delimit region boundaries.

I.1 Segmentation Evaluation

When evaluating the results of a segmentation algorithm, it is important to ignore the semantic biases that enable humans to see complex visual entities as single objects instead of seeing their component elements. In general, the regions generated by the low-level system will not correspond to objects or even parts of objects, but rather they will correspond to relatively homogeneous visual surfaces or parts of surfaces.

For example, a shadow which lies beneath the body of a car will be detected as a region separate from the pavement upon which it is cast. Further, the shadow region could conceivably merge with black tires or any other adjacent dark object. The overall VISIONS system will hopefully be able to recover the underlying object parts through predictive model fitting. In this example, the high-level processes might hypothesize the presence of a car on the basis of other information in the segmentation. The location of the tires may then be deducable even in the absence of clear sensory information.

Boundary placement presents a further problem in the evaluation of a segmentation. When two adjacent surfaces do not have a clearly defined boundary, that is, if for example there is a slowly changing gradient across them, then the placement of the boundary may be quite arbitrary. Again, it may be possible to predict and accurately delimit the location of the underlying object boundaries, based on, e.g., shape and linearity assumptions. However, it may not be possible to precisely determine--by machine or human--where the surface boundaries belong since surfaces can have arbitrary (unpredictable) features.

Finally, since the level of segmentation detail necessary to satisfy an interpretive system is in general ambiguous, the regions which comprise the segmentation may have to be processed and structured in a hierarchical graph, each layer representing a finer level of detail in the segmentation of the parent region. In this manner, a recursive segmentation (which will be addressed in Section VIII) is analogous to the process of describing complex objects in terms of components and sub-components.

I.2 Local Spatial Region Growing

Initially researchers in scene analysis approached the problem of segmentation with the development of local (neighborhood-oriented) image transformations. A region grower [BRI70] is an example of a local operator through which adjacent pixels are associated with the same symbolic region label if they are within a predefined threshold of each other.

This operator will extract surfaces optimally if the minimum difference in the feature value across any surface boundary is greater than the maximum feature difference for a pair of adjacent pixels which are internal to any surface in the image. Even if such a feature were found, the problem of setting the difference-threshold (theta) remains: theta should be set to that maximum internal feature difference. If it is set too low, the resulting segmentation will be fragmented in undesirable ways. If theta is set too high, regions will appear overmerged with respect to the underlying surfaces.

Given that it is extremely unlikely to find a single threshold which works correctly on all portions of the image, a variable-threshold region grower can be developed. This operator might use locally measurable pixel variation as a criterion for pixel merging. But even this improvement will not necessarily facilitate discrimination of internal variations (e.g., due to texture or lighting) from those variations which represent the boundaries of adjacent surfaces. This results in arbitrary splitting and merging of regions.

I.3 Global Feature Analysis

An alternative approach to segmentation relies primarily on global feature statistics, i.e., computations that ignore the spatial location of pixels [NAG77,OHL75,PRI77]. Prominent peaks in the probability density function (histogram) of a feature indicate the most frequently occurring values in the feature-image. The global analysis makes the assumption that the peaks -- and the clusters of points that extend from them -- correspond to distinct surfaces in the image.

The basic paradigm of this approach is to (1) identify the major peaks in the distribution of a feature and (2) assign a symbolic label to image pixels according to the cluster that they fall in. Adjacent pixels that bear the same cluster label can then be grouped and relabelled as belonging to the same region. This approach will work optimally under the following conditions:

(1) There is a one-to-many correspondence between cluster and surface.

(2) The distributions of individual surfaces do not overlap in the overall histogram of the feature.

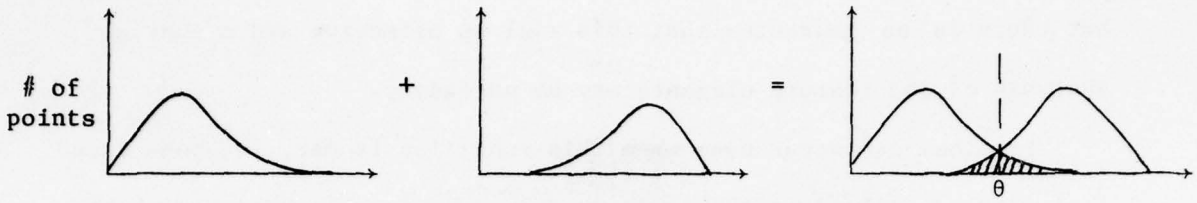
Violating the first condition is very difficult to remedy. Suppose that the distribution of feature values of a single surface generates two or more clusters. This can occur whenever a surface is textured with distinct atomic elements (micro-texture elements) so that each element belongs to a different cluster. In this case, a region labelling process based on cluster affiliation will fragment the single surface into many small pieces. This situation might be preventable by judicious feature selection and preprocessing, such as smoothing textural variation,

but there is no guarantee that this will be effective and a spatial analysis of the texture elements may be necessary.

Problems can occur even when this condition is met. Suppose that a cluster actually represents values from two visually distinguishable surfaces. If these two surfaces are not adjacent in the image, that is if they are separated by a surface whose feature values lie in another cluster, then the final region labelling will be, fortuitously, successful. However, if the two surfaces touch or if they touch another surface whose feature values lie within the same cluster, they will be incorrectly merged and labelled as the same region.

There is a partial solution to this problem, but it is costly. The segmentation algorithm can be applied recursively to each region found in the previous step for which there remain clusters in the histogram of some feature. When all regions have unimodal histograms, the algorithm terminates. This is, of course, an iterative, non-parallel process and recursion should be minimized for real-time processing.

The second condition is rarely, if ever, satisfied in natural scenes and it is the reason why clustering algorithms are difficult to implement. But even the best clustering algorithm, i.e., the one that best discriminates the peaks in a distribution, can only minimize errors if there is cluster overlap (see Figure 1).



(a) Distribution of surface 1

(b) Distribution of surface 2

(c) Distribution of 1 and 2. θ is the optimal threshold for mislabelling.

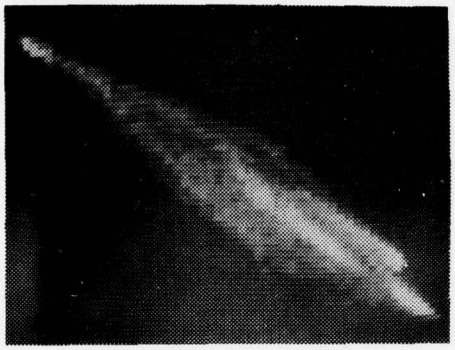
Figure 1. Histogram Overlap Causes Errors.

In the one-dimensional case shown in Figure 1 it can be seen that although it is perhaps easy to isolate the two clusters of the combined distribution, the algorithm will clearly induce a small percentage of erroneously labelled image pixels; the information is simply not available in this representation to determine which points in the shaded area will correspond to which surface in the image. Further, in n-dimensional feature space, the problem of identifying and delimiting the clusters produced by real data poses a non-trivial problem (Figure 2). One would like to use higher dimensional feature spaces to locate clusters which are hidden in the individual features and which may be more recognizable due to feature dependencies.

I.4 Summary of Global Analysis Problems

There are three major problems associated with the global feature clustering method, the first two of which have been alluded to:

(1) Clustering is a non-trivial process which often involves making two difficult decisions:



2-dimensional feature space for the image in Figure 4. The y-axis shows the distribution of red-filtered intensity values. The x-axis shows the distribution of black-white intensity values. The origin is in the upper-left corner.

Figure 2. n-Dimensional Feature Space.

n-dimensional feature space is difficult to cluster even when n is small. It is difficult to determine (1) what constitutes a peak and (2) what points in the "gray areas" belong to what clusters.

(a) identification of the major peaks of the distribution of a feature, and

(b) determination of the full extent of the clusters.

(2) Even the best clustering algorithm will lead to erroneous labelling of pixels since the formation of clusters in feature space does not take into consideration the spatial distribution of pixels in the image which formed the clusters.

(3) The mapping of a single symbolic cluster label back to an image pixel is only a gross representation of the information available in feature space; this disregards the relationship of each pixel to the cluster as a whole, and its relationship to other clusters.

I.5 Relaxation Labelling in Image Space Using Feature Clusters

Our approach will take into account both the global information in feature space and the spatial organization of this data in the image space (see Figure 3). Instead of mapping a single cluster label back to each image point, the probability that an image point belongs to each of the clusters can be mapped back to the image [SCH77]. This will be accomplished by extracting a representative center point for each cluster and using the relative distance of the feature values of the pixel to these clusters in feature space to determine the probability for each cluster label. The effect is to map most of the information in feature space back into the image where spatial information can be utilized. A relaxation labelling process is now rather natural since the probability that an image point belongs to each of N clusters is available. Similar labels will support each other, while different labels will compete over local neighborhoods in the image. In this

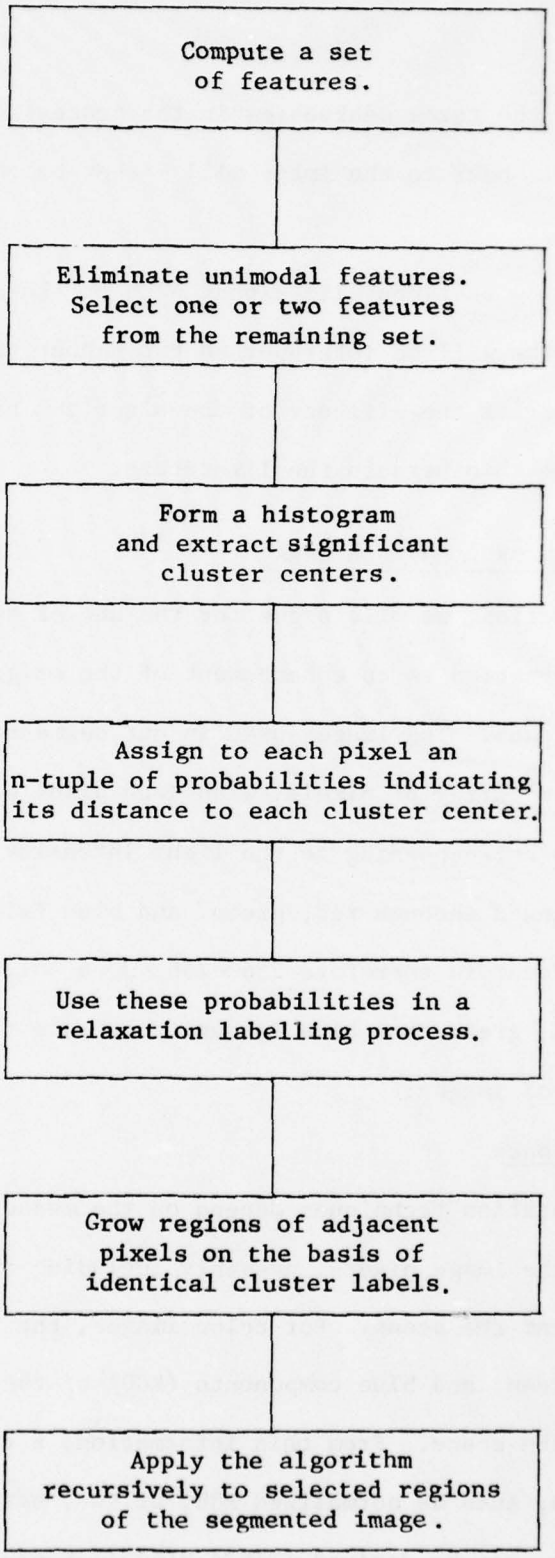


Figure 3. Segmentation Algorithm Overview.

manner, each of the three weaknesses in the process of mapping histogram clustering labels back to the image will either be entirely circumvented or else reduced.

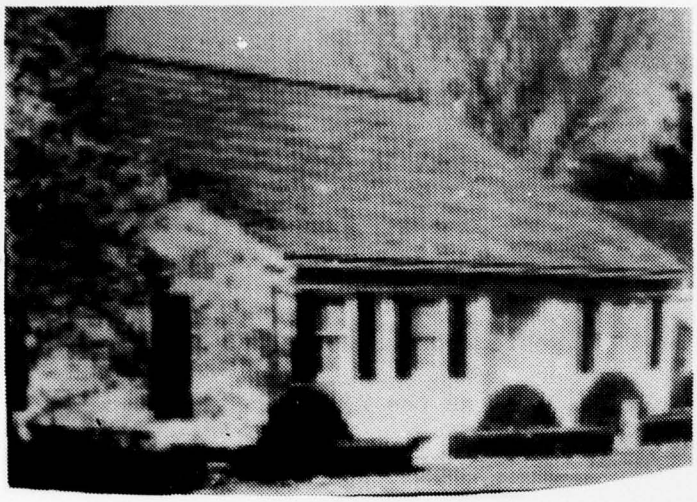
The following sections will expand each of the steps listed in Figure 3. Results will be interspersed throughout the text and will serve to demonstrate the efficacy of the algorithm as well as to contrast our techniques with others in the literature.

II. COMPUTATION OF COLOR FEATURES

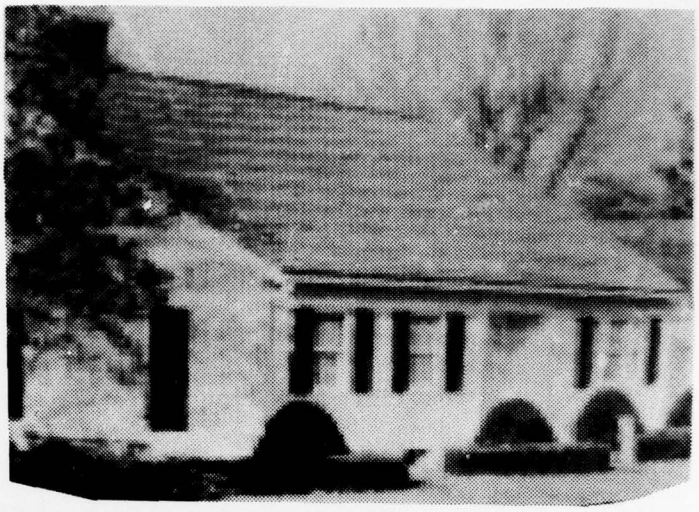
In this section, we will argue for the use of an opponent-color feature transformation as an enhancement of the original red, green, and blue image data. The images used in our segmentation experiments consist of a 256^2 array of pixels, with each pixel having a triple of six-bit numbers corresponding to the light intensity at a point in the grid as scanned through red, green, and blue filters. The total information content is therefore $256 \times 256 \times 3 \times 6 \simeq 1.2$ megabits. Figure 4 shows the red, green, and blue intensity outputs for a typical image in our library of images.

Color Feature Space

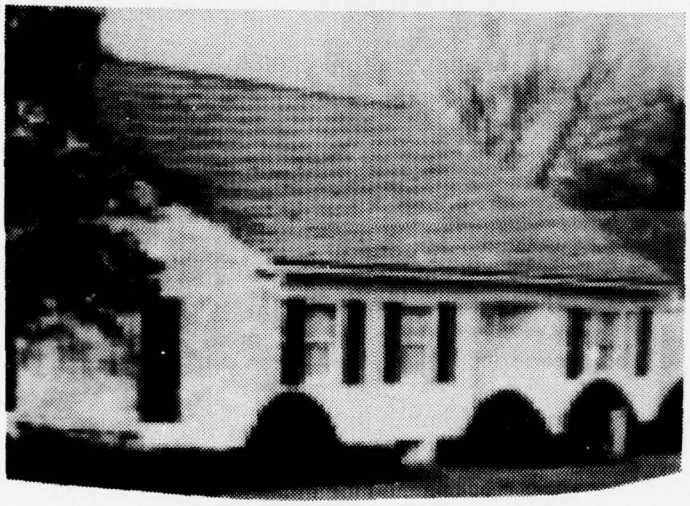
The segmentation techniques depend on the measurement of some feature(s) of the image pixels, possibly including those originally used to represent the scene. For color images, the usual measurements are the red, green, and blue components (RGB) of the light level at each pixel in the scene. From this information, a variety of other representations, such as normalized RGB, or hue, saturation, and intensity (HSI), may be derived [TEN74,RIS77]; because many of these



(a) Red



(b) Green



(c) Blue

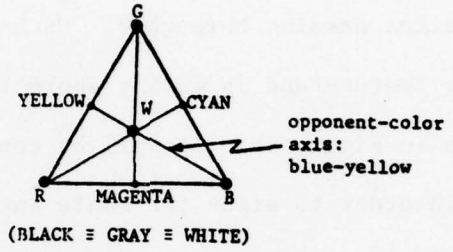
Figure 4. Typical Scene Showing RGB Intensity Data.

This is a 128^2 portion of a picture which was digitized to 512^2 resolution and quantized to six bits/pixel.

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 representation used in the television industry.

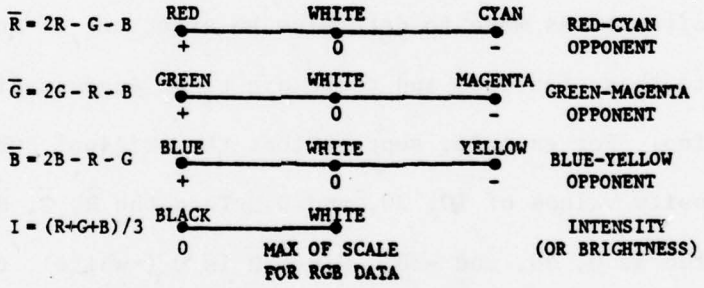
More recently, Sloan and Bajcsy [SLO75] have argued for the use of an opponent-color representation which has been proposed as underlying the color mechanisms in human vision [COR70]. Simply stated, the effect of this transformation is to parameterize the RGB color data into an equivalent set of features which have particular 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.

Figure 5 illustrates a very simple linear computation of opponent color features. Figure 5a is a standard way of depicting color information on a triangle, where the most saturated possible values of R, G, and B are associated with the vertices. A point interior to the triangle represents a color which can be obtained by combining specific amounts of the R, G, and B primaries; points on the perimeter are totally saturated while interior points are less saturated (i.e., diluted by white light). The interior point W, equidistant from the vertices, represents white light composed of equal amounts of R, G, and B. It



(a)

The color triangle. The axes passing through the neutral gray point represent the opponent-color features.



(b)

Opponent color features are computed as a linear function of the RGB data. They provide a way for assessing actual color as a scalar feature in a more meaningful way.

Figure 5. Opponent-Color Features

forms a neutral gray, including black and white. An axis representing intensity would be perpendicular to the page passing through W.

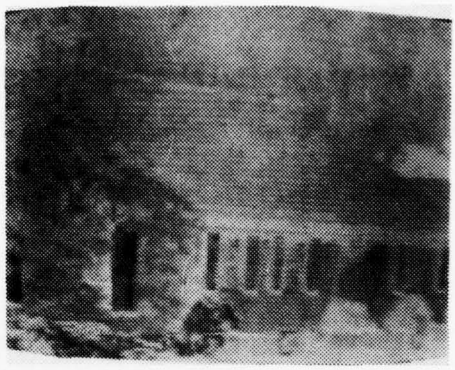
The three axes shown in Figure 5a are uniquely determined by the line from each vertex passing through W. Each of these represents an opponent-color feature and is easily approximated from the original RGB data as shown in Figure 5b. It will be convenient to add a constant to each feature in order to slide the scale into the positive range. Each opponent color feature has the effect of heightening color contrast between particular types of colors. They will be referred to by $\overline{R}$, $\overline{G}$, and $\overline{B}$, but the reader should remember that the letter used only represents one end to which the opponent color feature is anchored.

No attempt was made to determine an exact set of analytic equations to compute these features and there are clear inadequacies in the current formulation. For example, suppose that the original RGB data at a pixel had intensity values of 10, 20, and 0. Then the $\overline{R}$, $\overline{G}$, $\overline{B}$ features would be computed as 0, 30, and -30. Since $\overline{R}$ is 0 (=white), there should be correspondingly equal amounts of red, green and blue in the original data. This is of course not the case. We conclude by adding that in informal experiments using a simple segmentation algorithm and one-dimensional feature spaces, the $\overline{R}$ $\overline{G}$ $\overline{B}$ features consistently provided more discrimination than the original RGB data. Figure 6 demonstrates the transformation applied to the R, G, B data in Figure 4.

III. FEATURE SELECTION

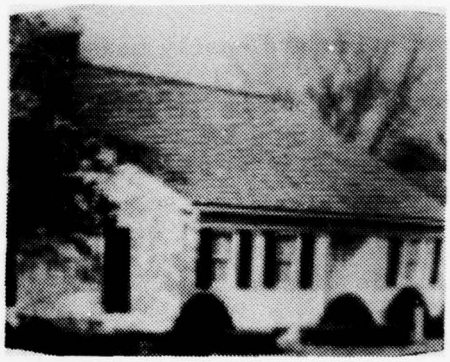
Once the various features have been computed, there arises the problem of selecting an appropriate subset to work with. In the discussion that follows, it will be assumed that semantic guidance is not available.

(a) $\overline{R}$



(b) $\overline{G}$

(c) $\overline{B}$



(d) Intensity

Figure 6. Transformation of the R, G, B Data from Figure 4 into $\overline{R}$, $\overline{G}$, $\overline{B}$, and Intensity.

Of course, in a full visual processing system, high-level, domain-specific knowledge could be used to guide segmentation routines in the selection of features previously found to be useful for particular problems.

III.1 Evaluation

Work has been done toward the development of a feature selection rule which can rank feature histograms on the basis of their peak structure. We postulate that a good histogram consists of many clearly separated equal height peaks with low minima between them. These characteristics are easily measured and a function has been developed that ranks histograms in a desirable manner.

It is interesting to note though, that what may appear to be the most promising histogram does not necessarily lead to the segmentation that is closest to a manual segmentation. Clearly, there are properties of a feature which are not represented in its histogram. This observation has led us to a different strategy for feature selection (see section VIII.1).

For the current discussion, feature selection reduces to an elimination mechanism: a feature will be eliminated from the working set if its histogram is nearly unimodal. Conversely, a feature will be acceptable if its histogram has the following characteristics:

(1) the two largest maxima are nearly the same height; and

(2) the minima between them is relatively low.

III.2 Selection Rule

We propose the following measure of the "peak quality" of a histogram of a feature:

$$M(f) = \frac{\left(\frac{P_2}{M_1}\right)}{\left(\frac{P_1}{P_2}\right)}$$

where: $M(f)$: measure associated with feature f

P_1 : absolute maxima of f

P_2 : next most prominent maxima

M_1 : lowest minima between P_1 and P_2

The numerator accounts for how well the two peaks are separated. The higher its value, the greater the probability that histogram points can be clearly distinguished as belonging to P_1 or P_2. The denominator accounts for how many pixels can be distinguished by the two peaks. Clearly, its optimal value is therefore 1. When M(f) is less than some threshold, the histogram is considered to be unimodal and the feature is therefore eliminated from the current segmentation step.

For simplicity, the segmentation results given in the following sections are based on a single decomposition step using one or two features. Since this is not the full recursive segmentation (demonstrated in Section VIII.3), the set of regions obtained will vary depending on on the sensitivities of the particular feature chosen. However, the power of the overall algorithm is not degraded by alternate feature choices within the working set.

IV. CLUSTERING

The previous section proposed a rule for evaluating the utility of a feature on the basis of certain measurable properties of the

distribution in a histogram of that feature. Although it is a trivial matter to identify local maxima and minima and equally easy to measure certain parameters of the overall structure of the histogram, it is not at all trivial to decide which maxima are true peaks. The decision mechanism which identifies true peaks and discards subpeaks and noise peaks is the major problem for our peak selection algorithm. However, standard clustering algorithms address themselves to a further and perhaps more difficult problem; namely, the assignment of cluster labels to the points lying beyond the peaks or cluster cores. The next section will review a few heuristic approaches that have been taken and is included to give the reader a sense of the difficulty of the problem. Section IV.2 will give a more detailed discussion of the peak selection algorithm that we currently use.

IV.1 Examples of Clustering Algorithms

Let us briefly review several alternatives for cluster extraction. We wish to point out that a variety of clustering algorithms appear in the pattern recognition literature. In pattern recognition applications, clustering algorithms are often applied only once to produce a characterization of the underlying data; in the application discussed here, clustering is one of many steps in region formation and must be repeated many times during the course of segmenting an image. In this case, computational cost is an important factor in the selection of a clustering method.

Ohlander [OHL75] has defined a set of rules for cluster detection based on analysis of local peaks and valleys, and their relative distances in one-dimensional histograms. In two dimensions, the problem

is more difficult because it appears to involve a search in two-space for the highest valley between two clusters. If the minimum value on each possible path between clusters represents the degree to which that path is considered to be a valley, then the limiting valley is that path which maximizes across all paths the minimum value on the path. This implies that an examination of all connected paths between the clusters is necessary--a computationally expensive process which is even worse in higher dimensions.

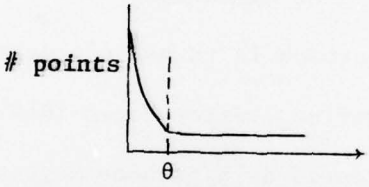
Another approach is to use a "conservative clustering algorithm" in an attempt to define cluster cores [HAN75,NAG77]. The two-dimensional histogram is treated as a pseudo-image; it is two-dimensionally averaged by reducing spatial resolution, and then weak values are thresholded. The effect is to spatially collapse relatively high values of the histogram which are in close spatial proximity into a connected cluster region, while deleting the valleys. A region growing process is then used to label the cluster cores in this reduced resolution histogram. This process is reasonably effective, although the criteria by which the threshold is determined as a function of the reduced values must be carefully studied for reliability.

One mechanism that we have used to compute the threshold involves an examination of a 1-D histogram of the magnitudes of the 2-D feature space histogram. This technique, as shown in Figure 7, reduces the 2-D problem to a simpler 1-D problem. The new histogram tends to have a characteristic inverse sigmoid shape -- assuming that the original histogram has a relatively normal peak structure. A threshold placed at the left

(a) Compute 2-D histogram.
x-axis is red.
y-axis is blue.

(b) Compute a 1-D histogram of the 2-D histogram values.

(c) Threshold (b) at its maximum slope. Points to the right of θ may be peaks or lie near peaks in the 2-D distribution.



(d) Turn on points in the 2-D distribution which > θ. Label these as unique cluster centers.

(e) A region grower can now be used to associate the remaining points with the nearest cluster center.

Figure 7. 2-D Histogram Clustering Algorithm.

edge of the tail of the 1-D distribution effectively devides the 2-D space into two components. Points in the 2-D space which lie below the threshold ("type 1") have a high frequency of occurrence and therefore will most likely correspond to valleys in the distribution. Similarly, points lying above the threshold ("type 2") are low-frequency and therefore they are most likely peaks (cluster centers). Our concern is with the type 2 points. Once these have been identified across the 2-D histogram, a region labelling process can be applied to distinguish isolated groups: adjacent type 2 points should be merged (0-difference merge) and then relabelled as unique cluster centers. These points can serve as the "seeds" for a minimum-distance classifier (or region grower) which will label the remaining points.

An iterative peak enhancement process has been described by Rosenfeld [ROS77]. On every iteration, each histogram bucket is compared pairwise to each bucket over some predefined neighborhood. The central bucket is increased or decreased as a function of the values in the neighborhood; the amount is directly proportional to the difference in bucket values and inversely proportional to their distance apart. This algorithm can be applied in parallel to all buckets, causing clusters to dynamically organize themselves. It appears quite appealing in that thresholds are not necessary, but it is sensitive (hopefully weakly) to the choice of neighborhood size.

IV.2 Identification of Cluster Centers in Feature Space

For our purposes, the cluster identification problem reduces to the selection of a single representative prototype point for the center of each cluster in the histogram. The algorithm need not be sensitive to relatively small misplacements of the center unless cluster centers are very close. The limits of each cluster no longer have to be determined--the probability of belonging to a cluster will automatically decrease with distance from the cluster center. A simple algorithm for extracting the representative center points involves pruning a sorted list of the maxima of the distribution of some feature(s). A maxima will remain active on the list if it is a preset distance from any maxima already on the list. In this manner, only the largest most isolated peaks will be selected to represent the feature space. Figure 8 shows the result of the algorithm applied to various histograms.

The algorithm is clearly crude and could easily generate arbitrary errors. For instance, areas of the histogram that lack significant maxima (i.e., plateaus) may be completely missed. Conversely, "noise maxima" might be identified which will lead to clusters that may or may not be meaningful (see Section VII.3). As it turns out, our experiments lead us to believe that the overall segmentation is relatively unaffected by these problems.

V. LINKING FEATURE SPACE TO THE IMAGE

The global analysis phase involves using the histogram representation to determine a small set of feature values around which the rest of the

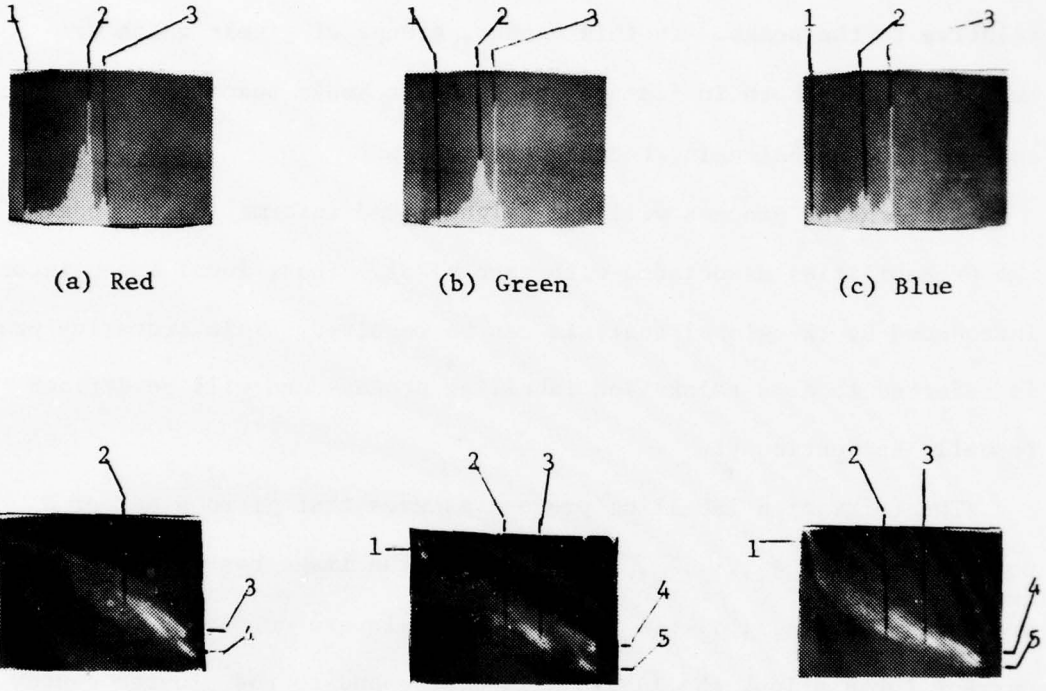


Figure 8. Examples of One- and Two-Dimensional Histograms. Indicated cluster centers were determined automatically.

data tends to concentrate. After identifying these peaks, the next step is to link this information with the individual pixels. 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.

The linking process will use neighborhood information to update the probabilities associated with each pixel. Thus, local inconsistencies introduced by the global analysis can be resolved. This iterative process is referred to as a relaxation labelling process and will be defined formally in Section VI.

The relaxation labelling process assumes that given a set of N possible labels, $\lambda_1,\ldots,\lambda_N$, each point in the image has associated probabilities $p(\lambda_1),\ldots,p(\lambda_N)$ that the labels are correct. In the current formulation, the labels will correspond to the cluster center representatives and the probabilities reflect the confidence that the image point is a member of that cluster. In the remainder of this section, we will discuss some of the properties of the initial probability labelling scheme.

V.1 Assigning Initial Probabilities of Cluster Labels to Image Pixels

The probabilities of the labels for some pixel should be a function of the distance of its position X from each cluster center in feature space. Figure 9 illustrates the situation in the two-dimensional case. Our choice among several possibilities for computing the initial probabilities is:

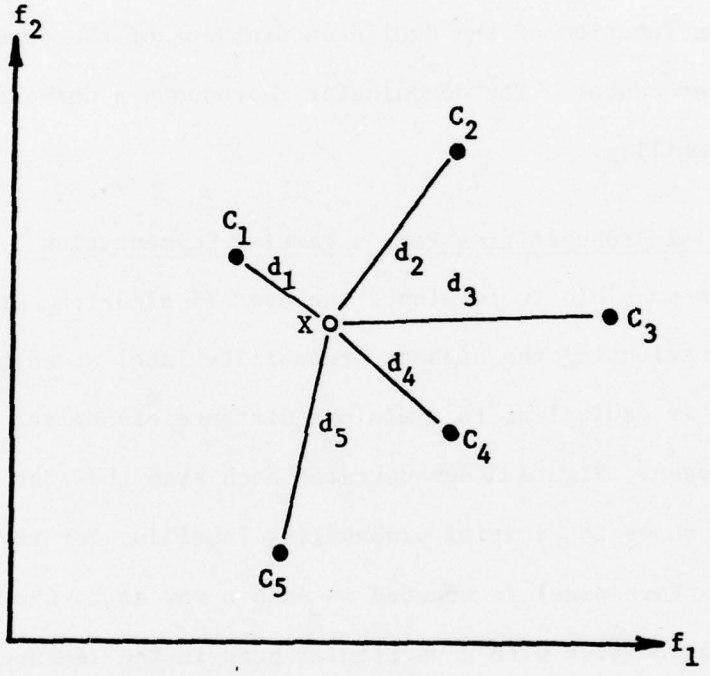


Figure 9. Initial Probabilities of Cluster Labels. The initial probability that histogram point X belongs to cluster C_i will be a function of distance d_i relative to the set of d_j, $j \neq i$.

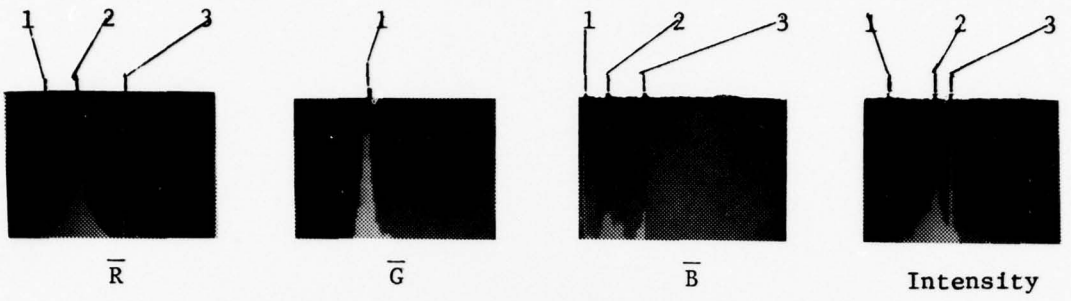
$$p_X(\lambda_i) = \frac{1/d_i}{\sum_{i=1}^{N} 1/d_i} .$$

This choice has the property that the probability is a monotonically decreasing function of the Euclidean distance of the point X from the i^{th} cluster center. The denominator represents a normalization to a true probability.

V.2 Initial Probabilities Form a Partial Segmentation

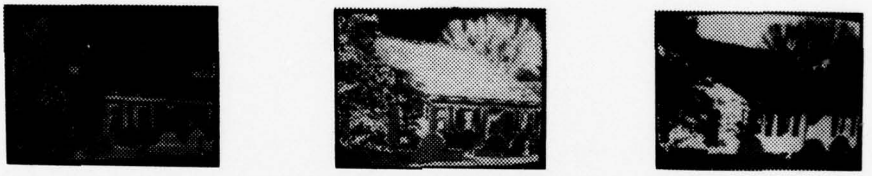
It is possible to terminate the overall algorithm at this point simply by selecting the highest probability label at each pixel. Note that this is equivalent to a minimum distance classifier in global feature space. Figure 10 demonstrates each step thus far discussed. Figure 10c shows the initial probability labelling for the feature selected. Each pixel is encoded in such a way as to show the probability of being associated with a particular peak in the feature. Brightness is displayed in proportion to probability. Figure 10d was obtained by selecting the highest probability label at each pixel and then displaying each label as a distinct gray label. A region grower can be applied across the label-image in such a way as to merge adjacent pixels if they bear the same label (i.e., theta = 0). The final result (10e) is displayed as region boundaries superimposed over the original intensity data.

By comparison, Figure 11 shows the result of slicing the feature distribution into buckets at the minima around the same peaks that were identified in Figure 10a. The pixels in 11 have been labelled as



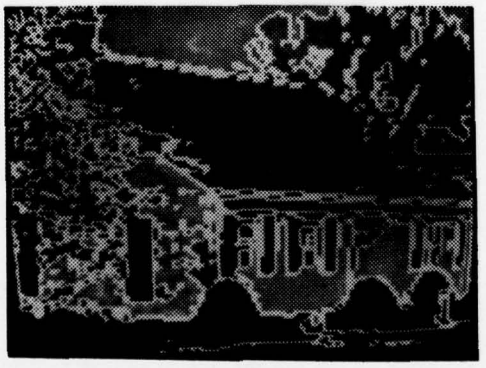
(a) Identify peaks in the feature-histograms.

(b) Eliminate $\overline{G}$ since it is nearly unimodal. Select intensity.



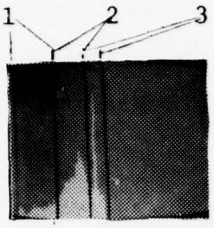
(c) Compute initial probabilities for each of the 3 peaks selected. Probability is displyaed as a gray level (0 = black).

(d) Pick highest probability label at each pixel. Display the 3 labels as 3 gray levels.



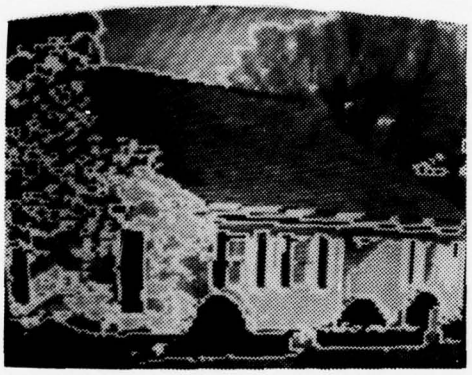
(e) Determine where edges lie between labels and superimpose these over the original intensity data.

Figure 10. Partial Segmentation Based on Probabilistic Labelling Technique.



(a) Slice intensity histogram at the minima around the peaks selected in Figure 10a.

(b) Encode image pixels as belonging to cluster 1, 2, or 3.



(c) Determine where edges lie between labels and superimpose these over the original intensity data.

Figure 11. Partial Segmentation Based on Histogram Slicing.

belonging to either cluster 1, 2, or 3. The final transformation in Figure 11c shows the region boundaries over the intensity data.

V.3 Comparative Evaluation of Two Segmentation Techniques

How can one compare Figure 10e to Figure 11c? Although the two outputs are similar, Figure 11c is actually a slightly better representation of the global feature information. In the distribution shown in Figure 12, consider the labelling in the image of a pixel whose feature value is X. By the probabilistic-labelling technique, any such pixel will be best labelled as belonging to P_1, since it is "globally" closer to P_1 than to P_2. This is unfortunate since X appears to belong to P_2 and would be labelled as such by the histogram-slicing method.

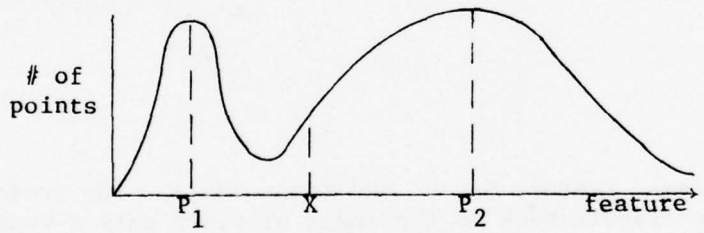


Figure 12. Error Associated with Probabilistic Labelling Technique. X is closer to P_1 but seems to belong to P_2.

It is important to realize though, that the decision as to whether X belongs to P_1 or P_2 is tenuous anyway, since there is no way of knowing in this representation whether pixels with feature value X will be contained in regions that are mostly composed of pixels in the cluster around P_1 or in the cluster around P_2. Figure 13 illustrates this point.

We now arrive at the most important aspect of the probabilistic labelling technique: it brings back to the image the relationship of each pixel to the distribution as a whole. The next step of the overall

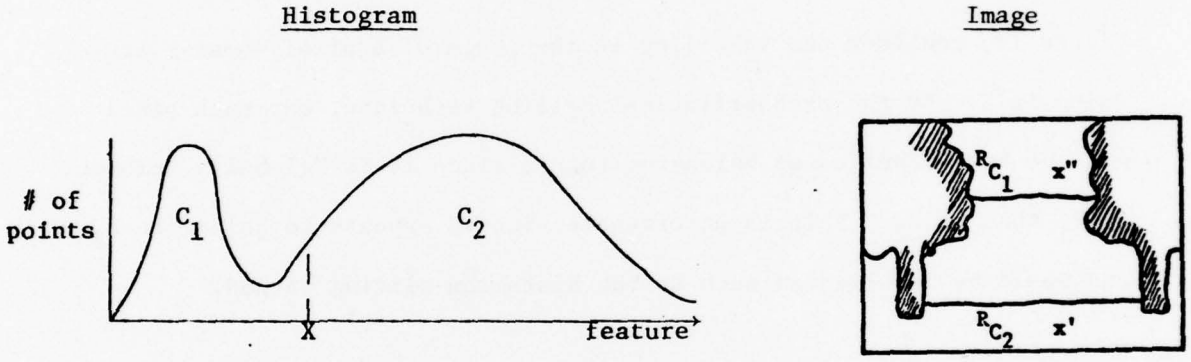


Figure 13. Histograms, Feature Space, and Image Space. The projection of histogram cluster labels back to the image provides only a weak mapping of information between feature space and image space. Consider some point x in the histogram of feature f, where its affiliation to cluster C_1 or C_2 is ambiguous. Now assume that R_{C_1} and R_{C_2} are regions produced by the clusters C_1 and C_2, respectively. A pixel contributing to histogram point x may have an image location x' or x", or in fact lie anywhere else in the image. The problem is more complex since this uncertainty exists even if x is in the cluster core of C_1 or C_2. Decisions regarding the region association of x should be a function of the information in both feature space and image space.

segmentation algorithm will resolve the errors introduced by the ambiguities inherent in the global representation and will produce a more accurate segmentation than presented in either Figure 10e or 11c.

To summarize, we note that in its crudest form, the algorithm we have presented so far is only slightly worse than the histogram slicing algorithm. This is because the former does not take into account the actual shape and extent of the feature clusters. It should be pointed out though, that the probabilistic-labelling technique relies solely on peak selection, a technique which is computationally faster, easier, and better defined that cluster analysis. This is especially true with n-dimensional feature spaces and in fact, as shown in the data given in Figure 8, this simpler approach makes analysis of pairs of features quite reasonable and effective.

VI. RELAXATION

The next stage of the segmentation algorithm resolves pixel-label ambiguities that were introduced by the global feature analysis by concentrating on the spatial organization of the data. A relaxation labelling process is used to defer the final labelling until a local concensus has been reached: a pixel whose feature value is globally close to one cluster yet which is spatially adjacent to a group of pixels whose feature values tend to lie in another cluster, can be labelled according to the local contextual information available.

VI.1 Formal Definition

Let us provide a brief review of the key ideas of relaxation processes; for a more complete discussion, see [ROS76,ZUC76,RIS77]. The general

idea is to compute some probability updating contribution Δ for the central pixel as a function of the probability of the neighboring pixels. It is assumed that each pixel location has a set of N possible labels $\{\lambda_1,\ldots,\lambda_n\}$ which can be associated with it. We will use $P_i(\lambda_k)$ to denote the probability of label k at the i^{th} pixel location, LOC_i. Furthermore, it is assumed that there is some means for computing a reasonable initial probability for each label at each pixel location. Then each label at each LOC_j contained in the neighborhood N_i of LOC_i will be used to update $P_i(\lambda_k)$, k = 1,...,n. $P_i(\lambda_k)$ will be increased (decreased) by label λ_m at LOC_j if the labels are compatible (incompatible) where the effect of this change is weighted by $P_j(\lambda_m)$.

Compatibility is defined in terms of a function r_{ij}:

$$r_{ij}(\lambda_k,\lambda_m) > 0 \quad \text{if } \lambda_k \text{ and } \lambda_m \text{ are compatible,}$$
$$< 0 \quad \text{if } \lambda_k \text{ and } \lambda_m \text{ are incompatible,}$$
$$= 0 \quad \text{if } \lambda_k \text{ and } \lambda_m \text{ are independent.}$$

Then, $\Delta P_i(\lambda_k) = \sum_{j\in N_i} d_{ij} \sum_{m=1}^{n} r_{ij}(\lambda_k,\lambda_m)P_j(\lambda_m)$, where d_{ij} is a weighting of the influence of LOC_j upon LOC_i and keeps ΔP_i in the interval from -1 to +1. Denoting the probability of label λ_k after the t^{th} iteration as $p_i^{(t)}(\lambda_k)$, it will be updated as follows:

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

Note that the denominator is a normalizing factor computed across the new probabilities of the n labels, so that the new values for p_i^{t+1} will sum to one.

In practice it is useful to keep the probabilities of all labels non-zero. Once a label has probability zero it will remain there during relaxation because the updating of probabilities involves a multiplicative function. Therefore, points with $d_i = 0$ (i.e., which are zero distance from the i^{th} cluster center) are treated as a special case; for these points, the i^{th} label will have probability approaching one while other labels are assigned small (but non-zero) values so that they sum to one; thus, all labels will have non-zero probabilities. This will allow the probabilities of other labels to grow if the context so demands, even for image points associated with a cluster center.

VI.2 The Compatibility Coefficients and Updating Probabilities

The compatibility coefficient between each pair of labels defines whether labels of neighboring pixels support each other or compete with each other. The coefficient is positive for identical labels and negative for differing labels. The simplest choice is to have

$$r_{ij}(\lambda,\lambda') = 1 \quad \text{if} \quad \lambda = \lambda'$$
$$r_{ij}(\lambda,\lambda') = -1 \quad \text{if} \quad \lambda \neq \lambda' \tag{1}$$

Notice that the linear summation across labels implies that the updating contribution from pixel j to $\Delta p_i(\lambda_K)$ will be zero if the probability of λ_K at location j is equal to .5 and will be negative if the probability is less than .5. However, even if all labels have total contributions which are negative, the probability of that label whose Δp_i is least negative will increase, relative to the other labels.

This simple specification of compatibility coefficients works reasonably well, but it can be improved by introducing relative weights on the coefficients which reflect the confidence that the two clusters really are distinct in feature space. This effect is incorporated for labels λ and λ' simply by scaling its negative contribution by the ratio of the distance between clusters λ and λ' to the maximum distance between any pair of clusters. Let $d_{MAX} = \underset{\lambda,\lambda'}{MAX}[d_{\lambda\lambda'}]$; then

$$(2) \qquad r_{ij}(\lambda,\lambda') = -\frac{d_{\lambda\lambda'}}{d_{MAX}} \quad \text{for} \quad \lambda \neq \lambda'.$$

This slows down the changes in label probabilities induced by the relaxation process in ambiguous cases where clusters are close together, and is relatively faster in clear cases where clusters are far apart. Note that the most distant pair of clusters will have an $r_{ij} = -1$.

There is one additional problem in the definition of the neighborhood of a region. If an 8-neighborhood is employed, right angle corners often cannot survive as probabilities converge to one. Figure 14 shows a pixel with label λ_2 at the corner of a region. In its 8-neighborhood, there are only three similar labels of λ_2 and five dissimilar labels. This causes the central pixel at the corner to change affiliation from λ_2 to λ_1 which then produces a stable situation. Use of a 4-neighborhood removes this difficulty, and any particular diagonal element still will have an influence upon the central pixel indirectly via two intermediate neighbors.

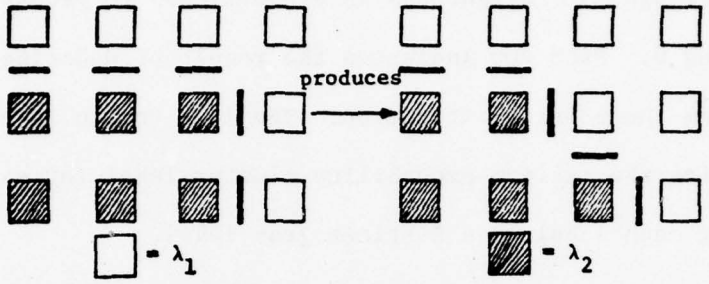


Figure 14. An 8-adjacency Neighborhood Causes Problems at Corners. As the label probabilities converge, the label λ_2 of the corner pixel will have competition from the high probability labels λ_1 at five neighboring pixels, and support from high probability labels λ_2 at only three neighboring pixels. As the neighborhood of labels converge, the corner pixel will switch affiliation from λ_2 to λ_1. The use of a 4-adjacency neighborhood removes this difficulty.

VII. RESULTS AND VARIATIONS OF THE BASIC ALGORITHM

We will now demonstrate the effects of the relaxation operator applied to the initial probability images. Unless otherwise specified, the following conditions apply to the relaxation operator:

(1) pixels are updated by looking at their 4-adjacent neighborhoods and

(2) the compatibility function is as defined in Section VI.2, namely:

$$r_{ij}(\lambda,\lambda') = -\frac{d_{\lambda\lambda'}}{d_{MAX}} \quad \text{for} \quad \lambda \neq \lambda'.$$

Figure 15 shows the results obtained from relaxing on the data shown in Figure 10c. Brightness is proportional to probability with black being 0. Each row indicates the result of updating the probabilities from the previous iteration. The last column is an image formed by selecting the maximum probability cluster label for each pixel and then displaying each label as a distinct gray level.

Notice that there is not very much change of label affiliation after about the tenth iteration. This effect has been observed in every image that we have tested. In the interest of saving computational time, we usually arbitrarily terminate the updating process after a few iterations. Although this is clearly not a true convergence, it seems sufficient for our purposes.

Figure 15b enables one to judge the accuracy of the segmentation at this stage. This figure was obtained by superimposing the boundary image formed from the maximum probability image after 25 iterations, over the original intensity data.

VII.1 Variations of the Algorithm

Figure 16 shows relaxation results applied to a sub-image of the intensity feature of Figure 6. Only the maximum probability label-images are shown. Each row represents a further iteration as indicated. The first column shows the result of the updating rule using 4-neighborhood adjacency and r_{ij} as defined in equation 2, Section VI.2. By comparison, column 2 shows the effect of 8-neighborhood updating. As predicted, corners are eventually destroyed, giving a notched appearance after 25 iterations. Finally, in column 3, the effect of the simpler compatibility function

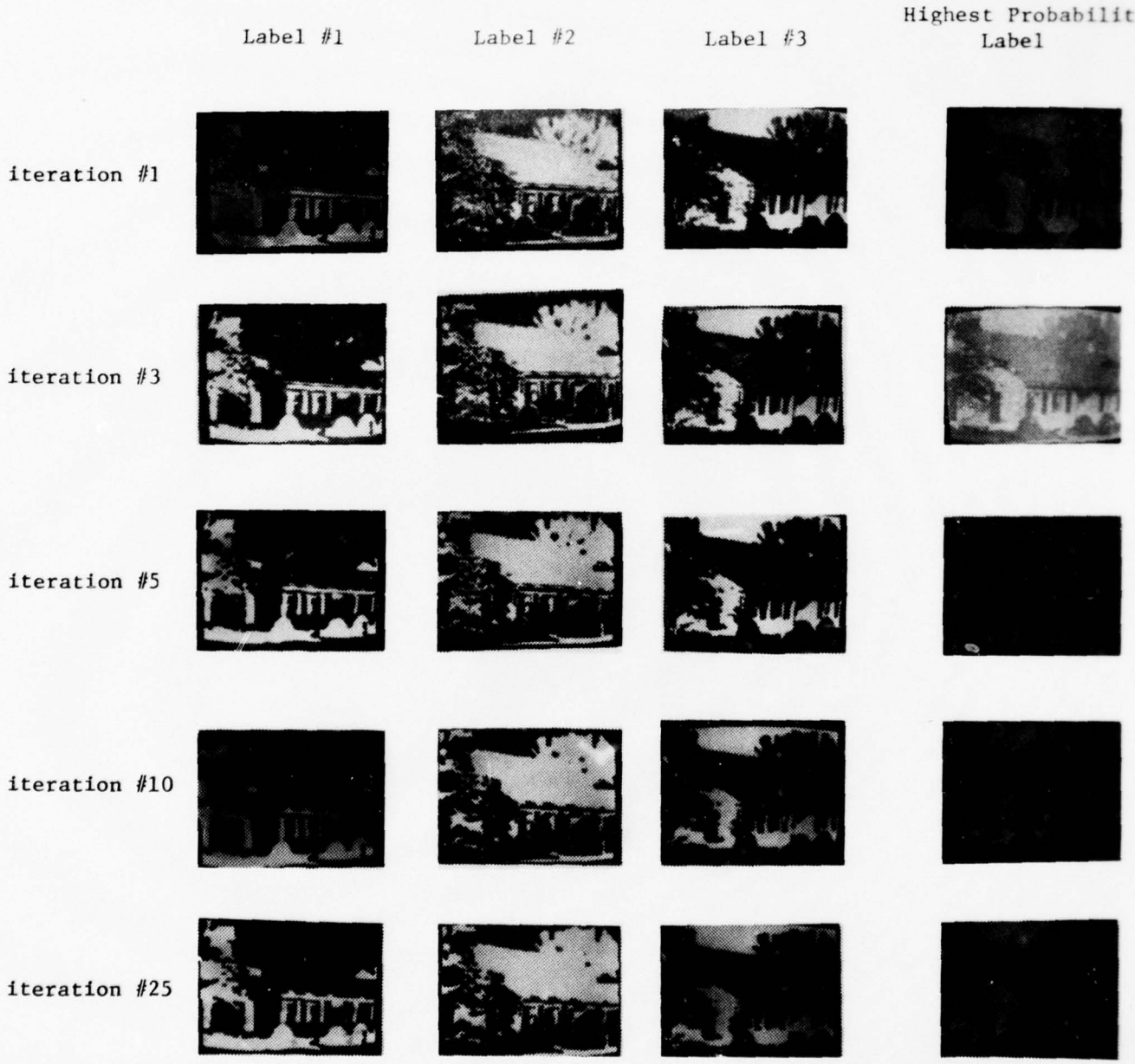


Figure 15a. Relaxation Applied to the Initial Probabilities Shown in Figure 10c.

Figure 15b. Highest Probability Label-Image After 25 Iterations. Display shows label-edges superimposed over the original intensity data.

(equation 1, Section VI.2) is demonstrated. The effect of this is to speed convergence--although the final result is very similar to the final results with the more sensitive function.

The relaxation labelling process is apparently very robust and insensitive to the variations shown above. However, since the 8-neighbor updating and "-1 compatibility function" have, in theory, inherent weaknesses, they will not be used in further experiments.

VII.2 Two-Dimensional Feature Analysis

Figure 17 shows the steps of the algorithm using two-dimensional feature analysis; in this case, $\overline{R}$ and intensity. The two-dimensional space provides a view of the data that is lacking in either of the one-dimensional features selected. Clusters which are hidden in 1-D are sometimes revealed in this representation. This of course leads to finer discrimination of regions. Notice that the final 2-D result (17c) provides discrimination of the bushes from the shrubs, but it also fragments the bushes into two pieces (highlighted crown and shadowed base). There is no easy way to decide whether the fragmentation that occurs is desirable or not. The joint distribution of the spectral characteristics of the pair of features indicates that the bush is composed of two visual features. Yet the grosser analysis of a single feature might segment the bush into one (type of) region. However, this is fortuitous, and the highlights of the crowns which are visible in this image could be a semantically important entity in another image. Thus, we conclude that this algorithm really is operating in a desirable fashion.

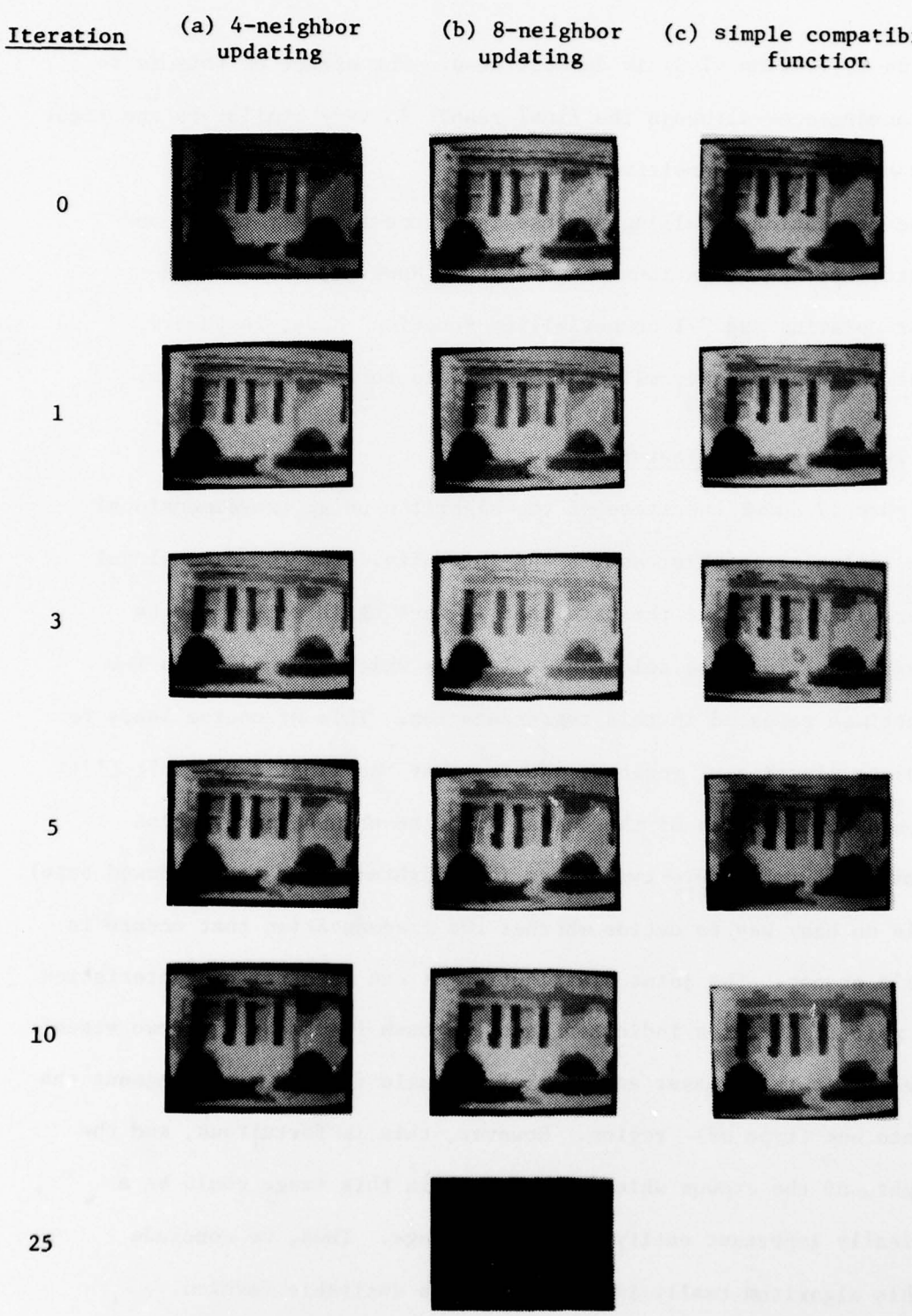


Figure 16. Variations of the Relaxation Algorithm.
Only the maximum probability labels are shown (see text).

(a) Form the 2-D histogram and isolate the six cluster centers as shown.
x-axis = intensity
y-axis = $\bar{R}$

iteration 0 iteration 1 iteration 3

iteration 5 iteration 10 iteration 25

(b) Form the initial probabilities for each label and then apply relaxation. For each iteration, only the highest probability label at each pixel is shown.

$\bar{R}$ intensity

(c) After 25 iterations label-edges are located and displayed over the original data.

Figure 17. Two-Dimensional Analysis of the Image.

VII.3 Varying the Number of Cluster Centers

The cluster center selection algorithm may make errors in detecting peaks. Several potential problems must therefore be kept in mind. Since the goal is to eventually obtain regions which are in close correspondence to the surfaces appearing in the scene, the algorithm should minimize the chances of arbitrarily splitting regions due to misidentification of cluster centers. This can occur in two ways: if a cluster is missed or if a cluster is mistaken to be two clusters.

If the clustering algorithm misses an obvious cluster in feature space (and consequently no label for this cluster is defined), the image points comprising this cluster will gravitate towards the clusters which are nearest in feature space. If there is only one, then the net effect will be to absorb the missing cluster into one which has been labelled. This type of error is not serious since recursive application of the region formation process will probably recover it later. On the other hand, if the cluster which is missed happens to lie between two or more clusters, then some of the feature points of the missing cluster may lie closer to one of the identified clusters, while others may be near a different cluster. This can be a more difficult error from which to recover. A region in image space which corresponds to the missing cluster could be split and absorbed into other nearby regions, if these nearby regions also happen to be associated with the clusters competing for the affiliation of the points in the missing cluster. It is much more difficult to recover from this kind of splitting since local evidence of similarity no longer exists -- the characteristics of the split region can be swamped by each of the regions which absorbed the pieces.

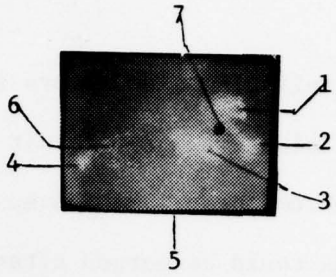
Arbitrary splitting also occurs if a single cluster is identified as two distinct clusters. The result is split regions which also could lead to the problems described above. However, in some cases two adjacent regions could be merged afterwards based upon similarity of their region characteristics.

Figure 18 shows the relaxation results (maximum probability label-images only) after purposely adding an extra cluster label as indicated in the 2-D histogram. Notice that the extra label has been added at a particularly dramatic location, namely at the centroid of the three largest and most closely spaced clusters. As expected, there is a degradation in the quality of the segmentation. A few of the regions seem to be arbitrarily fragmented and arbitrarily merged to adjacent regions. It is curious though, that the effect of altering the number of global reference points is not as drastic as might be expected and the correct regions can possibly be recovered (see Section VIII.2). The explanation for this is not fully understood but must be a function of the high degree of spatial organization inherent in the data.

VIII. HIERARCHICAL DECOMPOSITION OF THE IMAGE

We have pointed out that any given scene depicted in an image admits to various levels of description; for example as an outdoor scene; or house, trees, sky, and grass; or windows, doors, roof, leaves, blue sky, clouds, blades of grass, etc. We are examining a multi-level description of the scene based on region properties.

(a) 2-D histogram of $\overline{R}$*Intensity showing location of the extra cluster.



iteration 0 iteration 1 iteration 3

iteration 5 iteration 10 iteration 25

(b) Relaxation results. Only the highest probability label at each pixel is shown.

(c) Label edges over Intensity.

Figure 18. Adding an Extra Cluster Label.

The results of this analysis could be stored hierarchically [FRE76], in which relationships between ancestors and descendants represent descriptive properties of the structure of the visual elements.

VIII.1 Recursion

Our recursive segmentation algorithm will be designed so as to enable recovery from two kinds of region mislabelling errors: fragmentation and overmerging. We define the segmentation at step n to be a PLAN [KEL71,PRI77,NAG77] to be refined at step n+1. The proposed structure of the PLAN will consist of a set of parallel segmentations, each one reflecting the application of a different feature transformation. Therefore, there will be no need for feature selection/ranking; it will be sufficient only to eliminate unimodal feature-histograms.

VIII.2 Fragmentation and Overmerging

It is expected that small regions are likely victims of fragmentation and thus the regions in each PLAN segmentation can be readied for refinement by first merging all small regions into nearby large regions. The merging criteria should be a function of nearness both in space and in average feature difference.

Each of the surviving PLAN regions -- hopefully few in number -- can then be checked for overmerging. The criteria here is simply the detection of a multi-modal histogram in any of the features of a PLAN region. The process of PLAN and REFINE will be repeated for each such region.

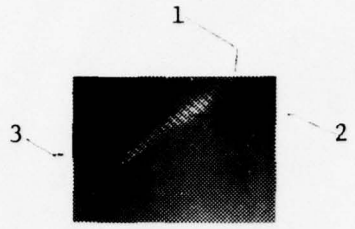
Hopefully, this process will not be overwhelming in terms of computation time and storage requirements. Fortunately, it has been observed that regions rarely require more than two or three recursive decomposition steps. Further, in the complete VISIONS system, regions will be decomposed selectively (i.e., in terms of their semantic interest) and with a small set of hypothesized features.

VIII.3 Recursion Results

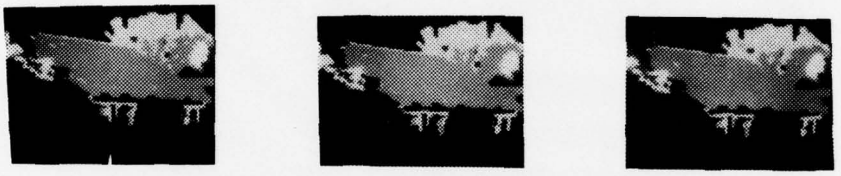
The idea of a recursive decomposition is demonstrated in Figure 19. Here, the roof/tree region is somewhat better partitioned by a recursive pass restricted to that portion of the image. The house roof, garage roof, and the tree with bare branches all have very similar features. When the 2D histogram is confined to only the overmerged roof region, the subtle visual differences in these areas appear as a major cluster with a nearby minor cluster which did not show up in the original histogram.

IX. CONCLUSION: A GLOBAL VIEW OF RELAXATION

The presentations thus far have shown the results of the segmentation algorithm solely in the spatial domain. However, a simple inverse mapping of pixels as a function of the cluster label to which they are ultimately linked can show the effects of the processing from a global perspective. Figure 20 is based on the final two-feature segmentation shown in Figure 17c. It consists of six 2-D histograms obtained by plotting, as separate 2-D histograms,



(a) 2-D histogram of roof/tree region from Figure 15a, iteration 25.



iteration 0 iteration 1 iteration 3

iteration 5 iteration 10

(b) Relaxation results restricted to the single region, only the highest probability label at each pixel is shown.

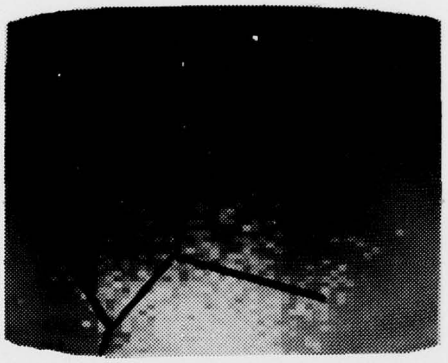
Figure 19. Recursive Segmentation on a Single Region.

(a) Cluster 1

(b) Cluster 2

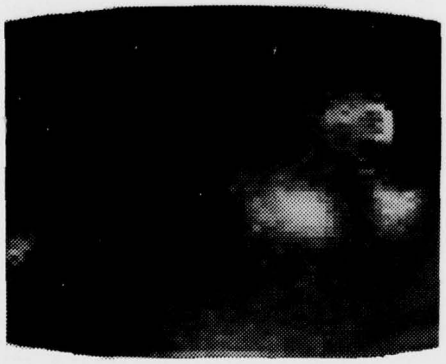
(c) Cluster 3

(d) Cluster 4



(e) Cluster 5

(f) Cluster 6



(g) overall histogram

Figure 20. A Global View of the Effects of Relaxation.

Histograms (a)-(f) show the location of the converged pixels in each cluster after 25 iterations. The overall histogram (x-axis = intensity; y-axis = $\bar{R}$) is shown in (g). The black lines indicate the location of the minimum distance decision surface.

the original $\overline{R}$ and intensity feature values of the converged pixels in each of the six labels. Had this mapping been done at iteration $\emptyset$ (initial probabilities), each histogram would merely contain a unique point corresponding to the frequency of occurrence of pixels whose feature values corresponded to the cluster center location for that label. However, after 25 iterations of relaxation, this mapping reveals the manner in which the original data was distributed around the representative cluster center points. By comparing these a posteriori histograms to the initial overall histogram, the extent of cluster overlap can be appreciated. That this overlap can be successfully detected is an indication of the power and necessity of applying a spatial organizing process beyond the global analysis to disambiguate the global information.

X. BIBLIOGRAPHY

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

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

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

[HAN74] A. Hanson and E. Riseman, "Preprocessing Cones: A Computational Structure for Scene Analysis," COINS Technical Report 74C-7, University of Massachusetts, September, 1974.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER COINS TR 78-8	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* SEGMENTATION USING SPATIAL CONTEXT AND FEATURE SPACE CLUSTER LABELS		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 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 5/78 13. NUMBER OF PAGES 52
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)*

segmentation, low-level visual processing, feature extraction, histogram clustering, relaxation labelling process, region analysis

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

The focus of this paper is on image segmentation processes, collectively referred to as a 'low-level' vision system. The programs which will be discussed here transform a large spatial array of pixels (picture elements) into a more compact representation through the exploitation of visual features, e.g., intensity, color, texture, etc. The goal is to detect a relative feature invariance across an area of the image and then to label all the pixels in any such area as belonging to the same region. Regions can be detected through

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

global analyses (e.g., histogram clustering) which find interesting areas by ignoring the local textural configurations of the data, in conjunction with local analyses (e.g., relaxation) which act as a fine-tuning mechanism both to resolve global ambiguities and to accurately delimit region boundaries.