REGION GROWING IN TEXTURED OUTDOOR SCENES*

By

A. R. Hanson*
E. M. Riseman†
P. Nagin†

COINS Technical Report 75C-3
February 1975

This research was supported by the Office of Naval Research under
Grant ONR N00014-67-A-0230-0007

*School of Language and Communication
Hampshire College
Amherst, MA 01002

†Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01002

*This paper is the Proceedings of the 3rd Milwaukee Symposium on Automatic Computation and Control, pages 407-417.

REGION GROWING IN TEXTURED OUTDOOR SCENES

Allen R. Hanson
School of Language & Communication
Hampshire College
Amherst, Mass. 01002

Edward M. Riseman
Computer and Information Science
University of Massachusetts
Amherst, Mass. 01002

Paul Nagin
Computer and Information Science
University of Massachusetts
Amherst, Mass. 01002

Abstract

Some progress has been made in scene analysis in finding boundaries between textured objects. However, standard line-finding and region growing algorithms have not been able to deal with strong local variations of intensity and color that repeat themselves more globally in images of natural scenes. These problems are significant with respect to both micro-texture, say the leaves of a tree, as well as macro-texture, the light green of leafy branches vs. dark green or black shadows between branches in a tree. How can local variations be globally bound together into a cohesive region for perceptual identification.

The approach that is being examined involves the reduction in size of the image while extracting features such as intensity, color, and color variation in local windows across the image. Two-dimensional histograms are being used to determine clusters of points with similar characteristics. Major clusters are labelled and the corresponding points in the reduced image are also labelled. This can directly detect regions of homogeneous micro-texture. Now an adjacency matrix of N x N clusters will denote the types of image points adjacent to each other. Peaks in the adjacency matrix can determine macro-texture.

All of this information can be used to grow regions in the reduced image and then in the detailed image. Experimental results on color images of natural outdoor scenes will demonstrate our results.

I. INTRODUCTION

One of the primary problems in scene analysis of natural two-dimensional images is the segmentation of the image into meaningful regions. Usually, the goal of such low level processing is to define regions whose identity can be used to form conceptual models of the world being viewed. Current region growing techniques work well in ideal environments, i.e., images in which regions are basically homogeneous with small variations in intensity (little texture). For this reason, images with highly textured regions containing variations in texture coarseness, directionality, and homogeneity are usually not examined. A notable exception to the constrained environment in which region growing takes place is the work of Yakimovsky and Feldman [1]. Extensive semantic direction for merging small regions was successfully applied to outdoor road scenes. However, we believe that the decision-theoretic techniques cannot deal with the wide variety of textures found in most natural scenes. Few researchers have examined the texture of a tree: in summer both the micro-texture of leaves and the macro-texture of shadows between leafy branches; in winter the micro-texture of webbed branches and the macro-texture of blue sky showing through the branches.

Both region growers and line finders are subject to similar difficulties due to the variety of textures which may be present. Texture may involve a simple statistical distribution (in 2 dimensions) of intensity; it might involve a mixture of two or more colors (hues); there might be a structural relationship between the size and adjacency of several micro-textures in forming a macro-texture; in fact, there could be a variation in saturation as exhibited by clouds in a wispy sort of white and blue sky. We are therefore using the term "texture" in a wide sense; it may involve variation in any measurable perceptual attribute. When a local boundary is found is this part of some global boundary that separates distinct regions or is this local texture that is a defining characteristic across this region? The line finder does not know whether to track the local boundary; the region grower does not know whether to jump the boundary. Although algorithms have been suggested, none have proven to be sufficiently reliable.

The problems we wish to adress ourselves to are:

a) what features of texture should be extracted in order to deal with natural scenes.
b) how can local characteristics of texture elements of varying color, size, and shape be globally bound together into a perceptual region. And
c) how can an algorithm deal flexibly with micro- and macro-texture.

Previous research has examined a number of characteristics of texture. Rosenfeld [2, 3] has examined texture in order to find boundaries between regions of distinct texture by looking for variations in either average gray level or edge/unit area (i.e. average amount of variation). This is one of the few attempts to deal with micro- and macro-texture simultaneously. He has also worked on measures of texture orientation and coarseness, but the techniques have not been applied to the full complexity of natural scene analysis.

Bullock [4] has recently carried out an excellent comparison of several edge detection operators. He presents extensive experimental results showing the confusion between detection of "micro-structure edges" of texture elements, and "macro edges" of major surface boundaries. This work seems to have goals similar to those reported here. However, we believe the extraction of multiple features of a region may prove more powerful than the edge characteristics.

Haralick [5] has examined micro-texture using spatial gray level adjacency matrices. Here an adjacency matrix of gray level i vs. gray level j for all gray levels can characterize various types of texture.

Bajcsy [6] has applied Fourier analysis to extract characteristics of texture orientation and texture gradients in natural scenes. However, these descriptors can only be obtained when operating on high resolution data and are primarily concerned with micro-texture. It is our contention that in fact it is often macro-texture that is most useful in perceiving the world. In most cases, it is not the size or shape of an individual blade of grass or leaf that permits us to see a lawn or the crown of a tree, but rather the global interaction of many of these micro-structures. Thus, we seek more flexible measures on coarser data, both for the computational practicality as well as utility. Consequently, we have chosen measures similar to those of Rosenfeld but generalized to color space. We have embedded these in a unique computational structure for adapting to any distinctly perceivable type of macro- and micro-texture.

II. Perceptual Motivation for our Approach

A region that is perceived to be cohesive is bound by some features which are relatively consistent across the region. One of these features might itself be a measure of the amount of local variation. Thus, a smooth blue sky has a smaller value for the attribute "average variation" than a roughly textured region. Let us assume for the moment that we have a number of features which can be applied to a local window. Further, we assume that this window is sufficiently large so that in a region with homogeneous texture (not necessarily color or intensity) adjacent local windows would have approximately the same amounts of variation within them.* Thus, we hypothesize that perception of a region as some cohesive entity requires the repetition of similar micro-texture across many local windows, although these windows need not be contiguous. This is a sort of statistical property requiring a certain frequency of distribution of a large number of these windows with similar properties.

The recognition of a macro-texture requires a distinctly different kind of perspective, one that analyzes certain structural relationships between types of micro-texture;

*Note that variance of these values would become greater as the window is reduced in size.

in particular, we will confine ourselves to the adjacencies between these types although other structural relationships could be examined. There is the trivial case of a large homogeneous region of micro-textured windows (call them type I) adjacent to each other; for example, a large expanse of blue sky has many type I windows of texture which are adjacent to each other.

On the other hand, an area of blue sky highly interspersed with green foliage should have many local windows of two types: smooth blue (type I) and roughly varied green (type II). Since there are many of each, by our first premise both will be noticeable. By our structural mechanism, many of these pairs are noticed to be adjacent and if these pairs occur contiguously across some large region, this will be detected as a macro-texture. We conjecture that at first glance people examining an image do not focus attention upon small textured areas or perceive such a macro-texture unless there are other prominent features of these regions. Thus, if only a small part of a sparse tree with blue and green is in the picture, the macro-texture may not be picked out and immediately recognized.

III. Computational Structure: Preprocessing Cones

This work is part of an attempt to develop a scene analysis system, called VISIONS (Visual Integration by Semantic Interpretation of Natural Scenes), which will build conceptual models of 2D color images of natural outdoor scenes [7-9].

In this paper, we will refer only to the front end of the system, the processing cone. It consists of a parallel computational structure which transforms and reduces large amounts of visual data in a layered fashion as shown in Figure 1. Information flows up, down, and laterally within the cone by defining local parallel functions which are duplicated to operate across the entire array. Parallel line finders, region growers, texture analyses, and color mappings, among others, operate on a 256^2 grid of raw image data and reduce it layer by layer to a 16^2 grid, and in some cases even to the 1 x 1 level, which contains information extracted from the entire scene.

There are three basic forms of processing available in the cone. The first is reduction as shown in Figure 1 which allows a function of a local window to pass a value to the next layer; thus, at each level a cell could have a value stored that is a function of the subcone below it; in this way, spatial relationships of extracted features are preserved. An iteration process allows the function of a local 5 x 5 window to be mapped into the center cell of the window. The last type is called projection and allows information to be passed down to a cell from the unique ancestral cell at each level above it. At any particular time, only a single layer of functions are being computed, but it is being applied uniformly across a layer with each local function applied to its local window. It should also be noted that each cell has a memory of several words so that a number of values can be saved for one or more processes.

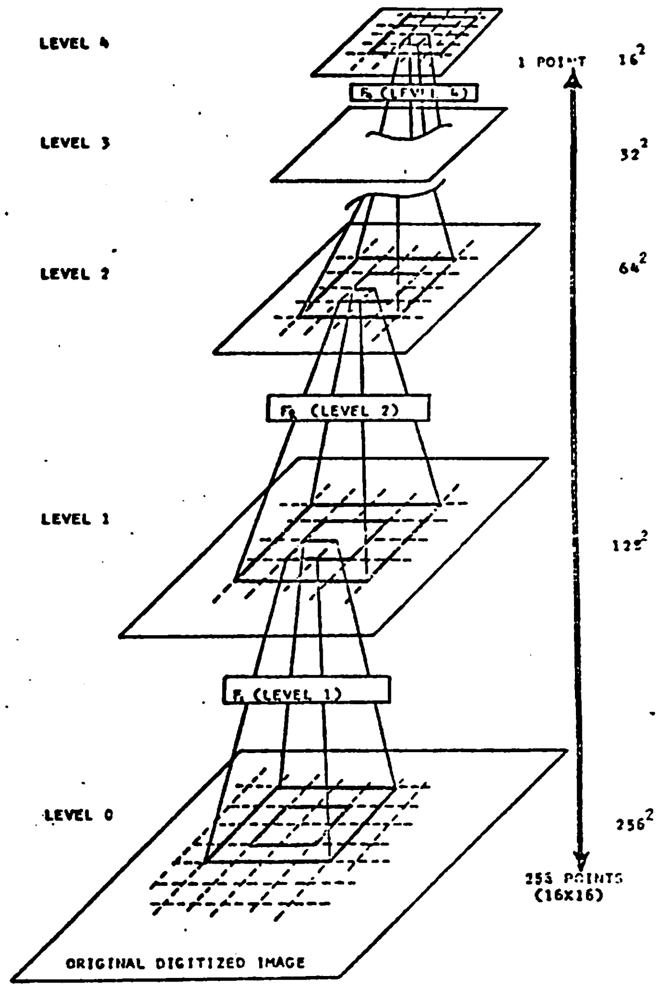
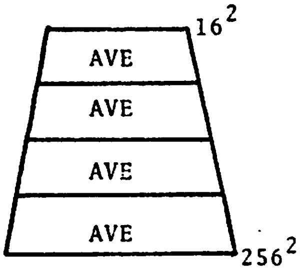


Figure 1. PROCESSING CONE

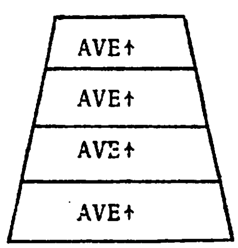
Algorithms for extracting features in the cone consist of a sequence of parallel reductions, iterations, and projections, possibly interleaved with sequential programs which may analyze and change information in a layer. In order to make this process a bit clearer, we will give a few simple examples of features that we will employ. We start with digitized color data: gray level intensities measured through blue, green, and red filters. This raw data can be mapped upward in the following manner.

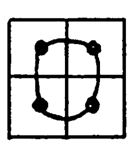
An average image in any color combination can be computed by letting each local function average its 2 x 2 center cells. Average intensity (a measure of the brightness) can be computed by summing the three components and averaging upward Hue and saturation [10] can be extracted by first normalizing each color component; two of these normalized components can then be used as coordinates of a digitized chromaticity diagram to map hue (the kind of color). Of course, this computation can be performed by our iteration type of function at any level, on the normalized raw data at level 0 or on the normalized averaged data at a higher level. The latter computation would extract average hue. A simple computation of saturation is a function of the normalized components



$S = 1 - 3 * \min(b_n, g_n, r_n)$.

All of the above functions measure some type of average value across a local area of the original image (whose size is dependent upon the level). Several measures of variation can also be extracted. The parameter that has been most valuable to us so far is average color variation in 3-space. This is just a differentiation and averaging algorithm. There are many possible differencing mechanisms; for computational simplicity, we choose $|b_1-b_2| + |g_1-g_2| + |r_1-r_2|$. Within the 2 x 2 center of a local window, we choose the max in the horizontal and vertical directions; i.e., the max of the four directions shown. Thus, at level 3 we have computed the average color variation in an 8 x 8 window of the original data. Although we will not discuss them further, variations of intensity, saturation, etc., can be computed in a similar manner.





IV. DETECTION OF MICRO TEXTURE

We have shown how the cone structure allows us to compute values for various parameters (features) which characterize certain aspects of the image. The ultimate goal is to extract from this information the types of textures in various parts (regions) of the image without attempting to quantify them by arbitrary predefined values of texture attributes. These parameters may be used individually to separate textures. For example, a histogram plot of hue from a region may indicate the contribution of the hue attribute to a perceived texture. One might expect, however that single parameters are not sufficient to adequately define the points comprising a textured region. Single paramter separations, such as average intensity across a region, fail when the region is composed of two types of textures, both of which exhibit the same or nearly the same average intensity. Clearly these remarks apply to any arbitrarily chosen attribute. For example, suppose we have a region composed of four texture types; we define a texture type as characterized by distinct values of the parameters chosen to describe the texture. Let us assume we choose average intensity and maximum intensity variation over a window as our parameters. Figure 2 illustrates a histogram derived from an image composed of these four textures.

Each circle in Figure 2 represents a distribution (or cluster) of image points having similar values of the parameters. In this case, removal of either coordinate results in a loss of discrimination; III and IV become one cluster or I and II become a single cluster.

If there are n parameters, one could map each image point into n-space, forming an n dimensional histogram in an attempt to discriminate texture types. We have chosen 2-space (two parameters) because a) the computation described below becomes unwieldy in n-space even for reasonably small values of n; and b) the cone structure is eminently suitable for 2-space manipulations in parallel as will become evident in the following discussion.

As a further simplification, let us assume we have only two texture types making up a region R as shown in Figure 3. If we

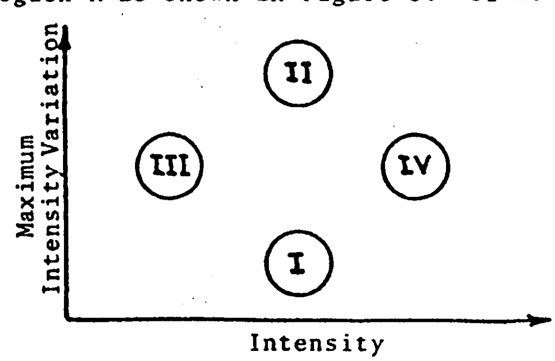


Figure 2. Histogram of Image Points Representing Texture Types

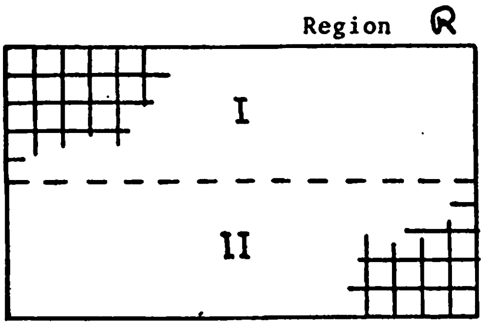


Figure 3. Distribution of two Textures in R

form a histogram of the values of two parameters, average intensity and maximum intensity variation, over each of the small windows shown in the figure, the clusters shown in Figure 4 could be formed. In the ideal case, in which the clusters are well separated, a simple region growing algorithm may be applied to the histogram in order to determine which histogram points comprise each cluster. In effect, we have formed a clustering algorithm in the cone that will produce results similar to stand ard clustering algorithms applied to two dimensional data. The clusters are labelled I and II as shown in Figure 4. Each window in region R may then be labelled type I or type II, since each of these image points map into one cluster or the other.

A running example of this algorithm applied to real data will now be introduced. Figure 5 is a B & W version of a digitized

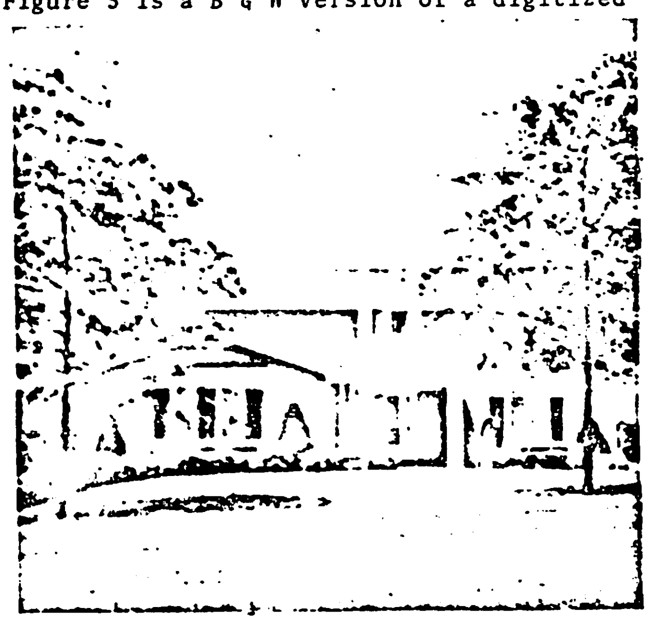


Figure 5. B & W Version of color image used in examples

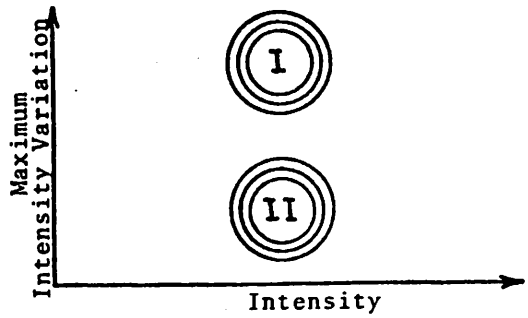


Figure 4. Histogram derived from Figure 3

color image kindly supplied to us by R. Reddy at Carnegie-Mellon. Interesting results have been obtained by applying the algorithm to the following pairs of features:

intensity vs. color variation
hue vs. color variation
hue vs. saturation
and hue vs. intensity.

This example will be restricted to intensity vs. color variation. The average intensity and average color variation from level 3 (32 x 32) in the cone is computed as we described in the previous section. This level of the reduced image is chosen because the local windows are reasonably large (8 x 8) and the 32^2 image still preserves some detail. These results are shown in Figures 6 and 7. The reduced 32 x 32 image is now mapped into a histogram of intensity vs. color variation, part of which is shown in Figure 8. All values greater than 61 are mapped into 61 so that it is a 61 x 61 matrix.

The real data shown in Figure 8 seldom conforms to our ideal examples for the following reasons. Two clusters may be linked by "bridges" of very low values which will cause the region growing algorithm to grow them into one region when in fact at a "gestalt" level, they should be distinct clusters. Furthermore, clusters which are distributed over an area may contain minor breaks which would halt the region grower before completely connecting the points while at a "gestalt" level, the points comprise one cluster. This implies that intermediate processing must be performed on the histogram in order to obtain the clusters.

One characteristic of clusters is that they are composed of a set of points spatially related according to some distance property which provides a gestalt grouping. Inherent in the cone structure is a similar distance property. Points that lie within a certain distance of each other (e.g., no

	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240	248
8	47	47	46	55	55	54	49	47	48	48	48	49	49	48	48	51	54	54	55	52	50	54	54	55	55	55	55	55	54	53	51
16	47	47	48	48	46	54	50	47	47	48	48	49	49	48	48	53	54	54	54	53	50	52	52	54	55	55	55	55	51	41	43
24	43	40	39	52	48	52	50	47	47	47	48	48	48	49	48	49	50	52	52	53	52	52	54	54	54	55	54	54	41	44	39
32	47	47	50	37	45	52	49	47	47	47	47	48	48	48	48	53	54	54	51	54	54	52	54	54	54	49	44	33	35	37	41
40	36	46	30	51	54	53	47	46	47	47	47	48	48	49	48	52	55	55	55	55	50	51	51	46	47	31	35	28	46	44	28
48	40	39	43	47	48	51	47	46	46	47	48	52	49	48	49	53	54	55	55	54	55	54	55	51	41	32	24	35	34	36	22
56	33	41	40	40	42	50	38	46	46	47	52	53	54	54	54	54	54	54	54	53	53	54	54	51	38	37	35	34	32	33	34
64	27	25	25	31	36	41	44	44	52	50	49	53	53	54	54	54	53	53	53	52	52	51	47	42	31	38	31	26	46	38	27
72	31	34	25	26	23	36	35	33	35	54	50	50	53	53	53	53	53	53	53	53	53	54	52	49	46	47	28	23	25	30	29
80	32	40	23	23	25	33	26	39	50	53	49	49	49	50	50	50	51	51	52	52	51	51	45	34	26	44	41	43	50	36	47
88	28	42	28	48	43	37	-33	43	51	50	50	49	49	49	50	50	50	50	51	51	51	42	38	29	29	40	24	27	23	28	32
96	19	34	32	42	51	35	40	28	30	33	44	49	49	49	50	50	50	50	54	54	54	55	42	48	32	36	32	31	31	32	30
104	29	34	20	26	26	27	31	21	27	25	28	38	38	40	40	40	40	40	42	43	42	36	43	31	34	32	28	25	34	35	33
112	31	33	27	37	36	35	26	27	27	28	28	29	31	31	32	32	32	33	33	34	32	33	34	27	32	29	29	24	18	23	34
120	24	35	25	41	51	41	28	27	25	27	28	30	31	32	34	36	37	39	40	41	42	37	26	32	27	38	38	26	26	45	38
128	32	39	29	23	31	24	33	19	19	29	19	19	19	20	21	21	21	36	32	37	27	22	26	25	42	29	26	23	19	30	38
136	30	45	21	36	40	42	34	35	23	25	31	47	27	22	23	23	23	25	30	25	26	28	30	28	27	18	24	23	19	29	28
144	35	34	19	30	33	19	17	19	20	21	20	19	18	31	45	25	24	25	26	26	26	26	26	27	25	23	21	24	21	40	34
152	45	22	22	19	18	17	31	45	43	54	55	53	38	21	21	24	36	35	36	37	37	38	34	50	48	41	44	35	22	37	49
160	30	36	27	19	22	21	33	34	46	27	39	25	54	28	26	20	25	19	11	24	14	44	38	28	35	42	32	56	30	45	53
168	50	50	31	25	16	28	43	43	39	28	27	21	54	17	32	28	26	19	11	27	16	48	30	24	39	43	33	55	22	46	50
176	33	35	25	30	15	27	46	50	52	44	42	34	53	18	24	27	13	14	11	26	16	48	17	14	47	49	39	47	17	43	36
184	30	33	22	27	29	31	29	30	20	27	22	31	24	20	29	28	24	29	26	33	16	48	14	16	24	21	21	31	23	31	32
192	28	26	17	14	15	16	25	28	30	33	36	37	39	39	39	39	39	38	40	39	40	40	40	40	40	42	41	43	27	40	41
200	16	12	21	35	36	34	32	30	27	26	24	26	28	31	31	36	37	38	38	37	39	38	37	37	36	35	33	32	30	32	34
208	29	25	21	22	22	22	25	28	30	33	33	35	36	37	37	36	36	37	35	34	34	34	34	35	34	35	36	38	39	38	39
216	32	31	31	31	30	31	31	33	35	35	37	35	37	38	38	38	36	36	34	34	34	34	34	35	35	35	35	36	36	37	38
224	32	32	31	31	30	31	31	34	34	35	35	36	36	37	34	35	36	37	37	36	38	37	37	40	38	39	38	36	39	39	37
232	33	32	34	34	33	32	31	30	33	32	34	36	36	38	37	37	37	36	37	37	37	38	36	39	40	39	38	38	37	38	39
240	34	34	34	31	31	31	33	33	34	36	37	40	39	38	37	38	38	39	37	37	39	37	37	37	40	40	42	43	42	42	43
248	36	36	37	37	38	36	36	35	36	37	35	35	38	41	40	41	42	39	42	40	40	39	40	40	42	42	43	45	44	45	46
256	36	37	36	38	39	40	39	38	38	39	40	40	41	41	43	42	41	40	42	45	49	47	49	51	52	51	50	51	51	51	51

Figure 6. Average Intensity of Original Image at 32^2 level

	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240	248
8	2	2	2	12	8	11	10	3	3	3	3	2	3	3	2	6	6	2	3	7	8	7	3	1	1	1	1	1	1	2	4
16	2	1	7	41	23	11	9	4	3	3	3	3	3	3	4	6	3	1	2	3	8	6	4	2	1	1	0	0	9	21	25
24	22	28	42	29	43	24	6	4	3	3	3	2	3	4	3	6	7	4	3	3	7	10	4	2	2	1	1	3	30	51	48
32	22	36	65	56	42	13	5	6	2	2	3	3	2	2	3	6	9	4	6	5	3	7	3	3	3	13	31	46	39	68	48
40	47	42	61	58	23	26	8	2	3	2	2	3	7	5	6	8	3	2	4	1	6	4	5	23	35	61	51	44	62	62	54
48	47	63	63	51	28	3	5	2	2	2	3	11	11	1	3	8	4	2	1	1	3	6	5	10	24	47	57	72	71	61	54
56	46	51	57	59	60	32	28	2	4	5	9	6	4	8	11	3	3	3	2	2	3	2	2	9	40	62	52	62	69	58	53
64	61	46	45	52	72	51	36	20	15	10	7	5	4	2	1	1	2	2	3	2	3	17	18	34	47	50	55	51	65	50	56
72	54	69	64	55	59	70	57	60	58	23	9	4	3	2	2	3	3	3	3	2	4	20	45	42	55	67	53	58	50	52	78
80	71	65	64	50	48	65	42	62	35	24	5	3	4	3	6	4	2	3	4	3	4	4	16	33	52	63	67	76	43	64	55
88	84	54	53	52	67	65	66	50	7	5	4	4	2	1	1	2	2	2	2	4	4	18	48	50	50	61	45	66	54	60	69
96	62	70	84	49	35	48	69	57	74	49	32	2	2	2	2	2	3	4	4	3	4	17	37	48	60	57	40	70	74	73	49
104	55	68	61	62	47	56	49	51	46	54	35	18	12	11	9	12	9	8	6	5	14	29	52	47	58	55	63	43	60	51	62
112	70	53	66	47	54	52	49	35	33	36	40	17	10	8	8	9	9	13	9	17	35	27	22	41	43	38	33	39	40	46	51
120	56	46	95	78	22	63	71	47	44	39	41	40	33	34	34	36	36	32	29	37	30	27	40	39	34	43	43	37	44	42	50
128	66	85	124	56	51	77	71	31	23	24	14	7	8	7	7	7	8	45	30	27	33	36	36	33	42	45	49	30	35	50	59
136	62	57	79	72	59	64	67	63	60	62	81	53	30	8	7	7	7	43	30	32	32	26	36	38	37	39	47	32	24	45	42
144	44	70	56	71	81	68	43	19	13	9	7	12	47	59	54	21	6	8	19	8	8	18	27	30	16	23	24	30	36	52	68
152	69	70	54	66	30	20	40	51	80	40	12	17	29	8	23	36	49	51	48	47	49	48	29	34	39	58	60	42	46	60	67
160	52	55	46	40	27	18	47	44	44	51	16	37	42	26	20	15	84	37	23	37	8	19	32	32	66	48	53	25	51	47	9
168	30	9	55	38	41	23	26	29	48	43	35	39	40	24	24	13	98	42	22	67	15	19	48	46	73	40	57	18	51	40	18
176	30	28	41	32	24	23	12	33	35	26	34	57	36	37	23	28	53	10	31	58	5	18	37	28	71	36	60	53	24	71	14
184	22	14	32	32	35	35	43	55	54	62	50	78	52	43	40	37	39	35	39	28	6	20	16	32	58	69	63	48	23	57	26
192	26	24	28	24	36	39	57	36	38	36	24	25	26	23	30	35	41	41	49	24	25	32	28	36	34	52	38	46	29	39	43
200	32	22	22	21	26	21	13	16	16	21	12	18	19	16	19	13	11	9	11	11	9	10	12	22	24	23	25	46	29	35	32
208	30	33	34	29	20	16	18	18	19	18	22	23	14	20	22	23	14	12	13	12	13	20	15	20	17	15	25	20	16	14	23
216	20	70	16	15	16	21	20	17	17	16	15	15	16	15	17	15	17	14	16	15	18	21	18	20	21	20	21	16	17	22	15
224	16	14	12	18	18	13	17	17	17	17	18	16	19	17	16	17	16	18	16	18	15	16	14	16	18	17	16	20	18	17	18
232	15	13	16	13	18	14	23	19	21	19	22	15	17	18	14	16	16	15	12	14	13	16	16	15	16	16	24	16	17	18	21
240	16	16	16	10	19	15	20	17	19	21	22	16	19	17	13	17	17	16	20	21	17	19	19	28	16	26	29	23	24	29	20
248	16	16	13	20	21	24	19	19	15	19	19	24	22	20	22	21	25	26	19	17	22	23	16	19	26	21	18	30	24	25	23
256	19	26	20	24	27	24	24	21	23	23	16	22	22	19	16	18	15	19	26	20	18	23	24	18	22	17	16	13	6	6	12

Figure 7. Average 3-Color Differences of Original Image at 32^2 level.

	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	0	0	0	18	30	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		SKY		0	12	24	12	15	12	6	3	21	27	6	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	18	42	15	9	6	9	27	33	9	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	6	3	9	12	18	9	6	18	3	0
5	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	6	0	9	3	3	0	3	3	3	0
6	0	0	0	0	0	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	3	3	3	9	6	3	9	6	0	0
7	0	0	0	0	0	0	0	0	3	6	6	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	3	3	3	9	0	3	0	0
8	0	0	0	3						0	6	3	0	0	3	6	0	0	0	0	3	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	0	0	6	6	3	3	3	3	0
9	0	0	0	0		HOUSE				0	3	0	0	0	0	0	0	0	0	0	0	6	3	0	0	0	0	3	3	6	0	0	0	0	0	0	0	0	0	9	6	3	3	3	0	0
10	0	0	0	3						0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	3	3	3	3	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	3	0	3	0	0	0	0	0	0	0	0	3	0	0	3	0	6	0	0
12	0	0	0	0	0	0	0	0	3	0	0	0	0	3	0	0	0	0	0	0	3	0	0	3	0	0	9	3	0	3	0	0	0	0	0	3	0	0	0	0	3	0	0	0	6	0
13	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	3	6	3	6	3	3	9	0	0	0	0	0	0	0	0	0	0	0	3	0	3	3	0	0	0	0
14	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	6	3	0	0	12	9	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	3	0	0
15	0	9	0	0	0	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	9	0	3	6	3	12	3	9	3	0	3	0	0	3	0	0	0	0	0	0	0	3	0	0	0	0
16	0	0	0	3	0	0	0	0	0	0	0	3	0	0	3	0	3	0	0	6	9	3	0	18	3	18	15	12	12	15	0	0	3	0	3	0	0	0	0	3	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	3	0	6	12	9	9	6	12	9	3	0	0	0	0	0	0	0	0	0	0	6	0	3	0	3	0
18	0	0	0	0	0	0	0	0	0	0	3	0	0	0	3	6	0	3	0	3	3	0	6	6	3	3	6	12	3	3	0	6	3	0	0	0	3	3	3	3	3	0	0	0	3	0
19	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	0	3	0	6	3	3	0	3	6	9	9	0	3	6			GRASS				0	3	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	3	0	0	0	0	3	0	0	0	3	0	0	0	0	6	3	3	3	9	6	9	3	0	0							0	3	0	0	0	0	0	3	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	3	0	3	6	9	3	3	6	3	0	6	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	3	3	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	3	0	0	3	6	0	0	12	3	0	9	3	0	3	0	0	0	3	0	0	0	3	3	0	0	0	0
23	3	0	0	0	0	0	0	0	3	0	3	0	6	3	0	0	3	3	0	0	3	0	0	0	6	3	0	3	12	0	0	0	3	0	0	9	3	0	0	0	0	0	0	6	0	0
24	0	0	0	3	3	0	6	0	3	0	3	0	0	0	0	3	0	0	3	0	0	3	0	0	3	9	0	6	6	3	3	3	0	3	0	0	0	0	3	0	0	3	3	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	3	0	3	3	0	3	3	0	3	0	0	0	0	0	0	0	0	0	0	3
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	3	0	0	0	3	3	0	6	0	0	6	3	3	0	0	0	0	0	0	0	0	3	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	3	0	0	0	0	0	0	3	0	0	0	6	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	9	0	3	0	0	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	3	0	3	0	3	3	0	6	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	3	0	3	0	3	0	6	3	0	0	0	0	0	0	0	0	3	0	0	0	0
30	0	0	0	0	0	0	0	3	0	0	0	0	3	3	0	0	6	0	3	3	0	3	3	0	0	0	0	0	3	0	3	3	0	0	3	0	0	0	0	3	0	0	0	0	0	0
31	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	3	0	0	0	0	0	3	3	0	3	3	3	3	0	3	0	0	0	3	0	0	0	3	3	3	0	0	0	3	0	0	0	0	0	3	0	0	0	0	0	0
33	0	0	0	0	0	3	0	0	0	0	0	0	0	0	6	0	6	3	3	0	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	3	0	3	0	0	0	0	0	3	0	6	0	0	0	0	0	0	0	3	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	6	3	6	0	3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	3	3	0	0	0	0
36	0	0	0	0	3	0	0	0	0	0	3	3	0	0	0	3	0	6	0	3	0	0	3	3	0	3	3	0	0	3	0	0	0	3	0	0	3	0	3	0	0	0	3	0	0	0
37	0	0	0	0	0	0	3	3	3	0	0	0	0	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	3	3	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	3	0	3	0	0	6	0	0	6	0	3	3	0	0	0	0	3	0	0	3	0	0	0	0	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	3	0	3	3	3	3	3	0	0	0	0	0	3	0	0	0	0	3	0	0	3	0	0	0	0	0	0	0	6	0	0
41	0	0	0	0	0	3	0	0	0	0	0	0	0	0	3	0	3	3	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	0	3	0	0	0	0	0	0	3	0	0	0	3	0	0	3	0	0	6	3	0	0	3	0	0	0	0	3	0	0
43	0	0	0		TREES				0	3	0	0	0	0	6	0	0	3	3	0	0	3	0	0	0	0	0	6	0	0	0	3	0	0	0	0	0	3	0	3	0	0	0	0	0	0
44	0	0	9	0					0	0	0	0		0	3	3	0	3	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0					0	0	0	0	0	3	3	0	0	0	6	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	3	0	0	0	0
46	0	0	0	0	0	&	0	0	0	0	0	3	3	3	3	0	6	0	0	0	0	3	6	0	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0		0	3	0	0	0	0	0	3	0	3	3	0	0	0	6	3	3	0	0	3	6	0	0	3	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	3	3	0	0	0	3	3	0	6	6	3	3	3	0	0	0	0	0	3	0	0	0	0	0	0	0	0
49	0	0	0		HOUSE				0	0	0	0	0	0	0	6	0	0	0	3	3	0	3	0	0	3	3	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0						0	0	0	3	3	0	3	0	0	0	6	3	0	0	0	0	0	0	0	9	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0	3	3		0	0	3	3	0	0	3	3	0	0	3	9	0	0	0	0	0	6	0	0	3	3	0	3	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	6	3	0	0	0	6	0	0	0	0	3	0	3	3	0	0	0	0	3	0	0	0	0	0	0	0	0
53	0	0	3	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	6	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	3	0	6	3	0	3	0	0	3	0	0	3	0	0	0	0	3	0	0	0	0	0	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	3	6	3	0	0	0	3	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	3	0	0	0	3	3	0	0	6	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	3	0	0	3	0	3	3	3	3	0	0	0	3	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	3	3	3	3	0	0	0	0	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	3	0	0	0	3	3	3	0	0	3	0	3	0	0	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	9	3	9	0	3	3	12	6	15	9	6	12	18	21	15	24	9	15	9	0	12	9	9	3	15	6	3	6	6	0	3	3	0	0	0	0	0	0

Figure 8. Histogram of 3-Color Differences (y coordinate) versus Intensity (x coordinate)

more than one window width apart) must end up in the same cell or in adjacent cells at the next level in the cone; at intervals of more than one level, the window width is the size of one side of the subcone below a cell. We may therefore treat the histogram as a pseudo-image and insert it into the cone at the appropriate level; in the analyses to follow, the histogram is inserted at the 64^2 level since the values of the parameters are in the range 0 to 63. The reason for treating the histogram as a pseudo-image is that we can now apply, in parallel, the complete range of operators available in the cone; e.g., reduction, scaling, region growing, projection, etc.

Once the histogram has been inserted in the cone, it is reduced to the 16^2 level via a scaling and averaging operation. By only using the 2 x 2 window below each cell during reduction from level 2 to level 4, each cell at level 4 is an average of the 4 x 4 window in the original histo gram. This operation has the effect of eliminating small values bridging clusters and tightening the clusters themselves; e.g., significant clusters remain intact while small diffuse clusters and stray points are averaged out. The results of this operation are shown in Figure 9 below. The simple region grower may now be applied to this highly reduced histogram at the 16^2 level. Once the clusters have been labelled, as shown in Figure 10, the 16^2 level may be projected down to the 64^2 level and the region growing algorithm applied once again using the already labelled points as the seed regions to

0	0	0	0	0	0	0	0	0	0	3	7	11	0
0	0	0	0	1	0	0	0	0	0	0	5	4	0
0	0	0	0	0	0	0	0	0	0	0	2	1	0
0	0	0	0	0	0	0	2	4	0	0	0	0	0
0	0	0	0	0	0	2	4	8	2	0	1	0	0
0	0	0	0	0	0	0	2	3	1	1	0	0	0
0	0	0	0	0	0	0	0	2	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	2	4	2	2	1	0	0	0

Figure 9. Histogram of Figure 8 reduced to 16^2 level.

Note: the leftmost and rightmost columns are zero and do not appear in Figure 9. In addition, the values shown in this figure are a result of a multiplicature scale factor of 3 (already reflected in Figure 8).

the algorithm. This has the effect of adding connected fringe points to the clusters identified at the higher level. The results of this algorithm at the 16^2 level are shown in Figure 11. Simple additional processing results in most histogram points being labelled according to the cluster (texture) type to which they belong; randomly scattered points, however, are not labelled.

Returning to the original scene data at the 32^2 level, a cluster label may now be associated with each image point since an ordered pair (intensity, color variation) of values uniquely identifies the cluster into which it maps. The results of labelling the original 32 x 32 reduced image are shown in Figure 12.

V. Detection of Macro Texture

At no point has the micro-texture analysis utilized any spatial information which could bind textured regions together; this information is not reflected in the clusters. Now we must try to detect the regularly occurring spatial distributions of texture types which we call macro-texture. By forming an adjacency matrix of N clusters by N clusters, some sense of the spatial relationships between texture types may be determined. A homogeneous region composed of points from a single cluster i will cause large values in the adjacency matrix on the $(i,i)^{th}$ diagonal element. Macro-texture is then determined by examining the large off-diagonal entries (if any). That is, a large number of adjacencies between points of two cluster types might signal a cohesive region that has a repetitive mixture of cluster types. The light green and dark green of a tree in bright sunlight is one example, as discussed above.

Again, let us assume we have determined two micro-textures. Figures 13a and 13b

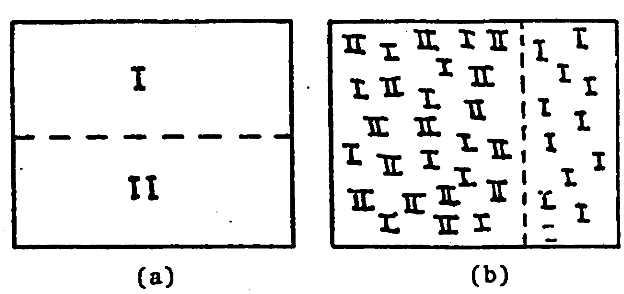


Figure 13. Example distributions of two texture-types and their adjacency matrices.

0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	2	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0
0	0	0	0	0	0	0	3	3	3	3	0	3	0	0	0
0	0	0	0	0	0	0	0	3	3	3	3	0	0	0	0
0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0
0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0
0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5	5	5	5	5	5	5	0	0	0	0

Figure 10. Labelled clusters at 16^2 level before including fringe points.

0	0	0	2	2	2	2	0	0	1	1	1	1	1	0	0
0	0	2	2	2	2	2	2	3	3	1	1	1	1	0	0
0	0	2	2	2	2	2	3	3	3	3	1	1	1	0	0
0	0	2	2	2	2	3	3	3	3	3	3	1	1	0	0
0	0	2	2	4	4	4	3	3	3	3	3	3	3	0	0
0	0	2	4	4	4	4	4	3	3	3	3	3	3	3	0
0	0	4	4	4	4	4	4	3	3	3	3	3	3	3	0
0	0	4	4	4	4	4	4	4	3	3	3	3	3	0	0
0	0	0	4	4	4	4	4	4	4	3	3	3	3	0	0
0	0	0	4	4	4	4	4	4	4	4	3	3	3	0	0
0	0	0	4	4	4	4	4	4	4	4	4	3	3	0	0
0	0	0	5	4	4	4	4	4	4	4	4	4	5	0	0
0	0	5	5	5	4	4	4	4	4	4	4	5	0	0	0
0	0	5	5	5	5	5	4	4	4	5	5	5	0	0	0
0	0	0	0	5	5	5	5	5	5	5	5	5	0	0	0
0	0	0	0	5	5	5	5	5	5	5	5	5	0	0	0

Figure 11. Final labelling of clusters at 16^2 level (after additional processing to retain fringe points).

	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240	248
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	3
24	3	4	4	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	4	4
32	3	3	5	5	4	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	4	4	5	4
40	4	4	5	5	3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	3	5	4	4	5	5	4
48	4	5	5	4	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	4	5	5	5	5	5
56	4	5	5	5	5	3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	5	5	5	5	5	4
64	5	4	4	4	5	5	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1	3	3	3	4	4	4	4	5	4	5
72	4	5	5	5	5	5	5	5	5	3	1	1	1	1	1	1	1	1	1	1	1	3	5	3	5	5	4	5	5	4	5
80	5	5	5	4	5	5	4	5	3	3	1	1	1	1	1	1	1	1	1	1	1	1	3	4	5	5	5	5	3	5	5
88	5	5	4	5	5	5	5	4	1	1	1	1	1	1	1	1	1	1	1	1	1	3	4	4	4	5	4	5	5	5	5
96	5	5	5	4	3	4	5	5	5	4	3	1	1	1	1	1	1	1	1	1	1	3	4	5	5	5	4	5	5	5	4
104	4	5	5	5	4	5	4	4	4	5	4	3	3	3	3	3	3	3	1	1	3	3	5	4	5	4	5	4	5	4	5
112	5	4	5	4	5	5	5	4	4	4	4	3	3	3	3	3	3	3	3	3	4	4	3	4	4	4	4	4	4	4	4
120	5	4	5	5	3	5	5	4	4	4	4	4	4	4	4	4	4	4	3	4	3	3	4	4	4	4	4	4	4	4	4
128	5	5	5	5	4	5	5	4	4	4	2	2	2	2	2	2	2	4	4	3	4	4	4	4	4	4	5	4	4	4	5
136	5	5	5	5	5	5	5	5	5	5	5	5	4	2	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4
144	4	5	5	5	5	5	4	4	2	2	2	2	4	5	5	4	2	2	4	2	2	4	4	4	4	4	4	4	4	5	5
152	5	5	5	5	4	4	4	4	5	3	1	3	3	2	4	4	4	4	4	4	4	4	4	3	3	5	5	4	4	5	5
160	4	5	4	4	4	4	4	4	4	4	3	4	3	4	4	2	5	4	2	4	2	3	4	4	5	4	4	3	4	4	1
168	3	1	4	4	4	4	3	3	4	4	4	4	3	4	4	4	5	4	2	5	2	3	4	4	5	4	5	3	4	4	3
176	4	4	4	4	4	4	3	3	3	3	3	5	3	4	4	4	5	2	4	5	2	3	4	4	5	3	5	5	4	5	3
184	3	3	4	4	4	4	4	4	5	5	4	5	5	4	4	4	4	4	4	4	2	3	2	4	5	5	5	4	4	5	4
192	4	4	4	4	4	4	5	4	4	4	3	3	3	3	3	3	4	4	4	3	3	4	3	4	4	5	4	4	4	4	4
200	4	4	4	3	3	3	3	3	3	4	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	4	4	4	4
208	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
216	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
224	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
232	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
240	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
248	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
256	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	3

Figure 12. Original 32^2 reduced image labelled by cluster membership.

illustrate two possible configurations of these texture types.

An adjacency matrix of these texture types can be formed in which a_{ij} represents the number of times texture-type i is adjacent to texture-type j (using four neighbor adjacency):

	I	II
I	a_{11}	a_{12}
II	a_{21}	a_{22}

In the case of Figure 13a, a_{11} and a_{22} are large because of the spatial distribution of textures; a_{21} is small and represents the boundary between the two regions. Since the adjacency matrix is symmetric, a_{21} may be ignored and set to 0. For Figure 13b, both a_{11} and a_{12} are large while a_{22} is small due to the absence of a homogeneous region of type II texture.

One of the problems we face is that some entries in the adjacency matrix are artificially inflated due to the interaction of texture types across the boundary of adjacent regions. In Figure 13b, it is not clear whether type I elements should be grouped with type II. If that is allowed, one big region will be formed as opposed to a homogeneous subregion of I's and a macro-textured subregion of I's and II's. The removal of the homogeneous region will make this distinction evident. In order to remove its contribution, a modified region growing algorithm may be applied to the texture type labels associated with the image points. This algorithm finds large connected homogeneous regions which are subsequently removed; construction of the adjacency matrix may then proceed without the contribution of boundary conditions. In the case of Figure 13b, this results in an adjacency matrix of the form

	I	II
I	x	O
II		x

where the circle represents a large value, the contribution due to macro texture, and the x's smaller values. Figure 14 shows the adjacency matrices for the 5 microtextures appearing in the labelled image of Figure 12 before and after the few major homogeneous regions (and their boundary effects) are deleted. The macro-texture of types 4 and 5 in the trees on both sides of the image are detected, but do not stand out as strongly as we would like.

There are a number of algorithms that are currently being examined to improve the

	1	2	3	4	5
1	367				
2	1	21			
3	67	10	294		
4	9	43	150	300	
5	4	14	47	216	178

14(a)

	1	2	3	4	5
1	2				
2	1	21			
3	6	8	71		
4	1	35	72	119	
5	2	14	40	122	111

14(b)

Figure 14. Cluster-type adjacency matrix for Figure 12.

performance of this portion of the analysis but no conclusions are available at the time of this writing. One significant weakness that must be dealt with is the problem of improper placement of the local windows (8 x 8 in our examples). Some of the windows fall entirely across a single type of micro-texture; others will overlap different types. If the boundaries of the trees (types 4 & 5) and sky (type 1) are examined in figure 12, one will note a line of points of type 3 which are a result of this effect. A number of ways to suppress these points are being examined.

VI. Conclusions

The algorithms discussed in the paper represent a portion of an on-going research effort. The power of the system resides in the maintenance of both a local and global perspective. The local extraction of texture and color features is blended with a global labelling of cluster types in the histogram. The experiments are incomplete and the results should be considered as preliminary. While we feel the results conclusively reveal the merits of this approach, several avenues of research are still open.

The results shown here are based on intensity versus color variation. While we have mentioned several other possibilities they have not been examined in any detail. One of the more exciting aspects here is the expectation that other pairs of parameters will yield additional information. Redundancies will produce regions with greater confidence. However, different feature pairs may extract different regions with different texture characteristics. The integration of this different information should yield a more complete characterization of the image.

The flexibility of the processing cones allow projection of an entire region down to lower levels of the cone. Examinations are underway of techniques for quickly growing regions at a relatively high level, say 32^2, and refining the regions by alternately projecting downward and region growing at lower levels.

All of the analyses to date have been derived from the original reduced image at the 32^2 level. While this level is sufficient for the detection of reasonably large regions, the quantification of both macro- and micro-texture clearly depends upon the resolution of the original image as well as the size and spacing of the texture elements. Regions containing small macro-texture elements or closely spaced regions (such as the house shown in Figure 5) may be analyzed at a lower level (64^2 or even 128^2) in precisely the same manner as described here. Furthermore, we can confine our processing on these lower levels to only those areas of the image which require further analysis. This type of low-level processing is being integrated into a full scene analysis system.

References

1. Y. Yakimovsky and J. A. Feldman, "A Semantics-Based Decision Theory Region Analyzer," Proceedings of Third Joint Conference on Artificial Intelligence, August 1973, pp. 580-588.

2. A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEETC, pp. 562-569, May 1971.

3. K. C. Hayes, Jr., A. N. Shah, and A. Rosenfeld, "Texture Coarseness: Further Experiments," IEEETSMC, pp. 467-472, September 1974.

4. B. Bullock, "The Performance of Edge Operators on Images with Texture," Hughes Research Laboratories Technical Report, Hughes Aircraft Company, Malibu, California, October 1974.

5. R. Haralick, "Textured Features for Image Classification," IEEETSMC, pp. 610-621, November 1973.

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

7. E. M. Riseman and A. R. Hanson, "Design of a Semantically Directed Vision Processor (Rev. & Updated)," Tech. Report 74C-1, Computer and Information Science, University of Massachusetts, January 1974.

8. A. R. Hanson and E. M. Riseman, "Pre-Processing Cones: A Computational Structure for Scene Analysis," Tech. Report 74C-7, Computer and Information Science, University of Massachusetts, September 1974.

9. D. Fishman, A. Hanson, E. Riseman, "Considerations in Model-Building for Scene Analysis," Tech. Report 75C- , Computer and Information Science, University of Massachusetts, January 1975.

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

Biographies

Dr. Allen R. Hanson received his Ph.D. degree from Cornell University in 1969. He taught in the Department of Computer Information and Control Sciences at the University of Minnesota from 1969-1973. He is currently an Assistant Professor at Hampshire College. His research interests have included a variety of topics in pattern recognition and artificial intelligence, particularly scene analysis.

Dr. Edward M. Riseman received his Ph.D. degree from Cornell University in 1969. Since then, he has been a member of the Computer and Information Science Department at the University of Massachusetts and is currently an Associate Professor. His research interests span the use of context in character recognition, AI systems for mechanized theory formation, and scene analysis.

Paul Nagin is a graduate student in the Computer and Information Science Department at the University of Massachusetts. He received his M.S. degree in 1974 and currently is working towards his doctorate. His research is in scene analysis systems, particularly the processing of texture and the segmentation of regions in complex scenes.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: COINS Technical Report 75C-3 / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): REGION GROWING IN TEXTURED OUTDOOR SCENES	5. TYPE OF REPORT & PERIOD COVERED: INTERIM
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): A. R. Hanson, E. M. Riseman, P. Nagin	8. CONTRACT OR GRANT NUMBER(s): ONR N00014-67-A-0230-0007
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computer and Information Science Dept. University of Massachusetts Amherst, MA 01002	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research Code 437 Washington, D.C.	12. REPORT DATE: February 1975
	13. NUMBER OF PAGES: 13
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

This paper was presented at the 3rd Milwaukee Symposium on Automatic Computation and Control, April 18 and 19, 1975, Milwaukee, Wisconsin.

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

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

Some progress has been made in scene analysis in finding boundaries between textured objects. However, standard line-finding and region growing algorithms have not been able to deal with strong local variations of intensity and color that repeat themselves more globally in images of natural scenes. These problems are significant with respect to both micro-texture, say the leaves of a tree, as well as macro-texture, the light green of leafy branches vs. dark green or black shadows between branches in a tree. How can local variations

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) be globally bound together into a cohesive region for perceptual identification.

The approach that is being examined involves the reduction in size of the image while extracting features such as intensity, color, and color variation in local windows across the image. Two-dimensional histograms are being used to determine clusters of points with similar characteristics. Major clusters are labelled and the corresponding points in the reduced image are also labelled. This can directly detect regions of homogeneous micro-texture. Now an adjacency matrix of N x N clusters will denote the types of image points adjacent to each other. Peaks in the adjacency matrix can determine macro-texture.

All of this information can be used to grow regions in the reduced image and then in the detailed image. Experimental results on color images of natural outdoor scenes will demonstrate our results.