TEXTURED IMAGE SEGMENTATION

by

Kenneth Ivan Laws

A Dissertation Presented to the

FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF SOUTHERN CALIFORNIA

In Partial Fulfillment of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

(Electrical Engineering)

January 1980

UNIVERSITY OF SOUTHERN CALIFORNIA
THE GRADUATE SCHOOL
UNIVERSITY PARK
LOS ANGELES, CALIFORNIA 90007

This dissertation, written by

Kenneth Ivan Laws

under the direction of his Dissertation Committee, and approved by all its members, has been presented to and accepted by The Graduate School, in partial fulfillment of requirements of the degree of

DOCTOR OF PHILOSOPHY

L S Grant

Dean

Date January 18, 1980

DISSERTATION COMMITTEE

William K. Pratt

Alexander A. Sawchuk *Chairman*

Alan Schumitzky

ACKNOWLEDGEMENTS

The guidance of Drs. William K. Pratt and Alexander A. Sawchuk is acknowledged. Programming support has been generously provided by Dr. Keith Price. Ray Schmidt and his staff at the USC Image Processing Institute provided photographic services. Special thanks are due to my wife, Lily, for supporting me through many years of graduate studies.

This research was supported by the Defense Advanced Research Projects Agency and was monitored by Wright Patterson Air Force Base under Contract F-33615-76-C-1203, DARPA Order No. 3119.

Views and conclusions in this document are those of the author and should not be interpreted as necessarily representing official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

Throughout this document a monochrome image will be denoted by brightness function I(r,c), where r and c are discrete row and column coordinates. I(r,c) is assumed nonzero only for the square region $0 \leq r \leq N$ and $0 \leq c \leq N$, although extension to other image shapes and coordinate systems is trivial. Image windows are similarly indexed nxn blocks. The image function may be considered a non-negative matrix. It can take either discrete values called gray levels or continuous values called luminance, brightness, density, or transmissivity. Individual image elements will be called pixels. Elements of texture-feature planes will also be called pixels. They may take negative values, but will be rescaled to a positive range for display as images.

This dissertation is the record of a search for fast, effective texture measures. Fortunately, the search was successful. Details of the search will not be of interest to all readers, however. Chapters 1 and 2 introduce the problem of texture segmentation and the historical approaches to texture analysis. Chapter 3 documents our

method of evaluating texture models. Chapter 4 applies this experimental paradigm to the co-occurrence method of texture measurement; this establishes a benchmark for evaluating other texture models. Correlation methods are investigated in Chapter 5. Chapter 6 traces the failures and partial successes of various "spatial-statistical" models. Chapter 7 presents the "texture energy" approach to texture measurement, and Chapter 8 develops it into an image segmentation system. Those interested only in the final analysis system should read Section 1.2 and Chapter 8. Chapter 9 contains a brief summary and suggestions for further research. Three appendices document the techniques used in this study.

Kenneth I. Laws

Los Angeles, California

November, 1979

ABSTRACT

The problem of image texture analysis is introduced, and existing approaches are surveyed. An empirical evaluation method is applied to two texture measurement systems, co-occurrence statistics and augmented correlation statistics. A "spatial-statistical" class of texture measures is then defined and evaluated. It leads to a simple class of "texture energy" transforms, which perform better than any of the preceding methods. These transforms are very fast, and can be made invariant to changes in luminance, contrast, and rotation without histogram equalization or other preprocessing.

Texture energy is measured by filtering with small masks, typically 5x5, then with a moving-window average of the absolute image values. This method, similar to human visual processing, is appropriate for textures with short coherence length or correlation distance. The filter masks are integer-valued and separable, and can be implemented with one-dimensional or 3x3 convolutions. The averaging operation is also very fast, with computing time independent of window size.

Texture energy planes may be linearly combined to form a smaller number of discriminant planes. These principal component planes seem to represent natural texture dimensions, and to be more reliable texture measures than the texture energy planes.

Texture segmentation or classification may be accomplished using either texture energy or principal component planes as input. This study classified 15x15 blocks of eight natural textures. Accuracies of 72% were achieved with co-occurrence statistics, 65% with augmented correlation statistics, and 94% with texture energy statistics.

CHAPTER 1

INTRODUCTION

Many tasks can be performed better by mechanical means than by biological systems. Not only are physical systems faster, more sensitive, and more attentive than any human, but also more quantitative. Image analysis is a task ripe for automation. This study will develop methods for extracting texture information from aerial photographs and images of natural scenes.

The goal of image analysis is extraction from an image of all the useful information it contains. Only through image analysis does photographic film become a useful medium for data acquisition. Most analysis is now accomplished by human interpreters, but mass screening applications are growing so fast that automation is essential.

Scene analysis is the extraction of region or object description from a given picture. The description may be numerical or it may be a data structure representing properties and relationships of the scene components. The following are important steps in the development of a scene analysis system:

1. Determine the purpose of the analysis.
2. Model the data source.
3. Analyze the model to determine useful features.

4. Preprocess data to remove known effects.

5. Extract features or segment the image.

6. Edit, resegment, or improve features.

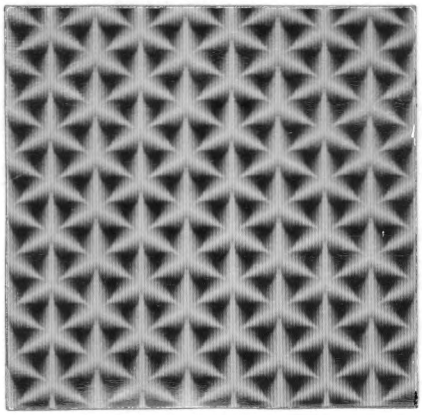
7. Code and/or display regions and boundaries.

8. Use extracted information for semantic scene analysis.

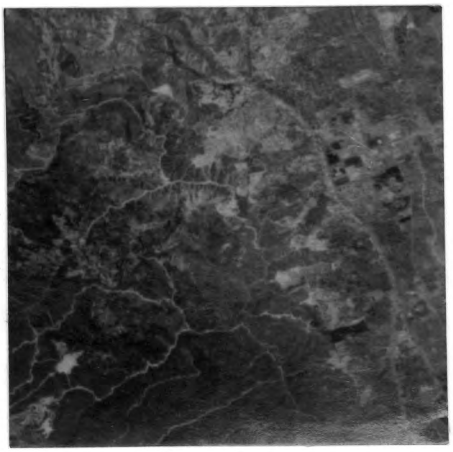
Texture analysis is fundamental to some applications, such as metal surface analysis and geologic fault identification. Appropriate theories of texture generation are required. In other applications, such as radiographic diagnosis, texture recognition is more important than knowledge of the physical generating mechanism. General image analysis systems, such as the human visual system, use texture as an aid in segmentation and interpretation of scenes.

Figure 1-1 illustrates two fundamental texture types. The first image is a "macro-texture," or high-resolution repetitive pattern. Structural analysis methods are adequate to describe such textures, although more than one type of description is possible. The other three images in Figure 1-1 are scenes which might be of interest in aerial reconnaisance and vehicle guidance. The scene components are differentiated by their textures, but description in terms of repetitive structural elements is impossible. This dissertation will develop methods of isolating and identifying small textured regions in natural scenes.

This study is not limited to any one application area



(a) A Structural Texture



(b) A LANDSAT Image

(c) An Aerial Image

(d) A Natural Scene

Figure 1-1. Examples of Textured Scenes

or data type, although it is biased toward the analysis of aerial images. Military and security applications of scene analysis are reconnaissance, night vision, mapping and terrain classification, target detection and tracking, traffic monitoring, personnel identification, fingerprint matching, and airport screening. Industrial and scientific applications include thermal analysis, parts inspection, particle counting, automation and robot vision, crop monitoring, remote sensing, geological analysis, cell classification, chromosome analysis, and radiological diagnosis. Scene analysis techniques might also be of use in pattern recognition and document processing.

1.1 Visual Texture Perception

Visual textures arise from many sources. Cellular textures are composed of repeated similar elements called primitives. Examples are leaves on a tree or bricks in a wall. Other texture types include flow patterns, fiber masses, and stress cracking. A complete analysis of any texture would require modeling of the underlying physical structure.

The human visual system is capable of discriminating and classifying all of these textures. It is obvious that spontaneous discrimination does not require built-in models of physical texture generators, although such models may be used by trained observers.

Texture is generally taken to mean whatever structure exists within a semantic region (one to which a name can

be assigned). One component of this structure is detail, small image regions that are identifiable but not semantically important. A second component is noise, taken to be any artifact of the imaging and quantizing process. The third component resembles noise, but is a property of the imaged object or scene. It arises from detail just beyond the perceptual resolving power of the analysis process, and seldom possesses a recognizable pattern or dominant repetition frequency. We shall call this component stochastic texture, micro-texture, or just texture.

Texture is both structured and random. It is common to speak of a uniform texture or a homogeneous texture, despite the apparent contradiction. This homogeneity is a perceptual phenomenon. Somehow the human visual system analyzes images and measures texture properties. Some texture fields are seen to be equivalent, others to differ in coarseness, linearity, or other texture dimensions. All, however, are unified by their perception as texture fields. We generally know a texture field when we see one.

Perception of related elements as a whole is known as grouping. Grouping is more fundamental than recognition, as demonstrated by figure-ground reversals and by ambiguous figures that cannot be recognized until parts are grouped [1]. We use contour, brightness, color, and texture for grouping, as well as stereopsis and relative motion.

Texture perception is itself a grouping phenomenon. Julesz [2] showed that spontaneous texture discrimination can occur even when recognition is prevented, and that a small amount of noise can disrupt texture perception if it destroys connectivity of texture elements. He comments that

> Instead of performing complex statistical analyses when presented with complex patterns, the visual system wherever possible detects clusters and evaluates only a few of their relatively simple properties. [p. 43]

If true, it does not necessarily follow that the eye segments an image before evaluating texture. This study will concentrate on an alternate hypothesis that local segmentation and texture description are performed at each pixel, with no global agreement on exact region boundaries.

The chief characteristic of texture is shift-invariance. Perception of a texture field does not change as its position on the retina changes. This seems to be the very definition of a texture field: an image that is not significantly changed by shifting. A region or object, on the other hand, is position dependent.

We shall define texture to be that which remains constant as a window (or fovea) is moved across an image. This presupposes that the image is a single texture field. Note that texture may change as a function of window size.

There is an ambiguity in the common meaning of texture. Let two texture fields be identical except for a difference in luminance. Most observers will say that the

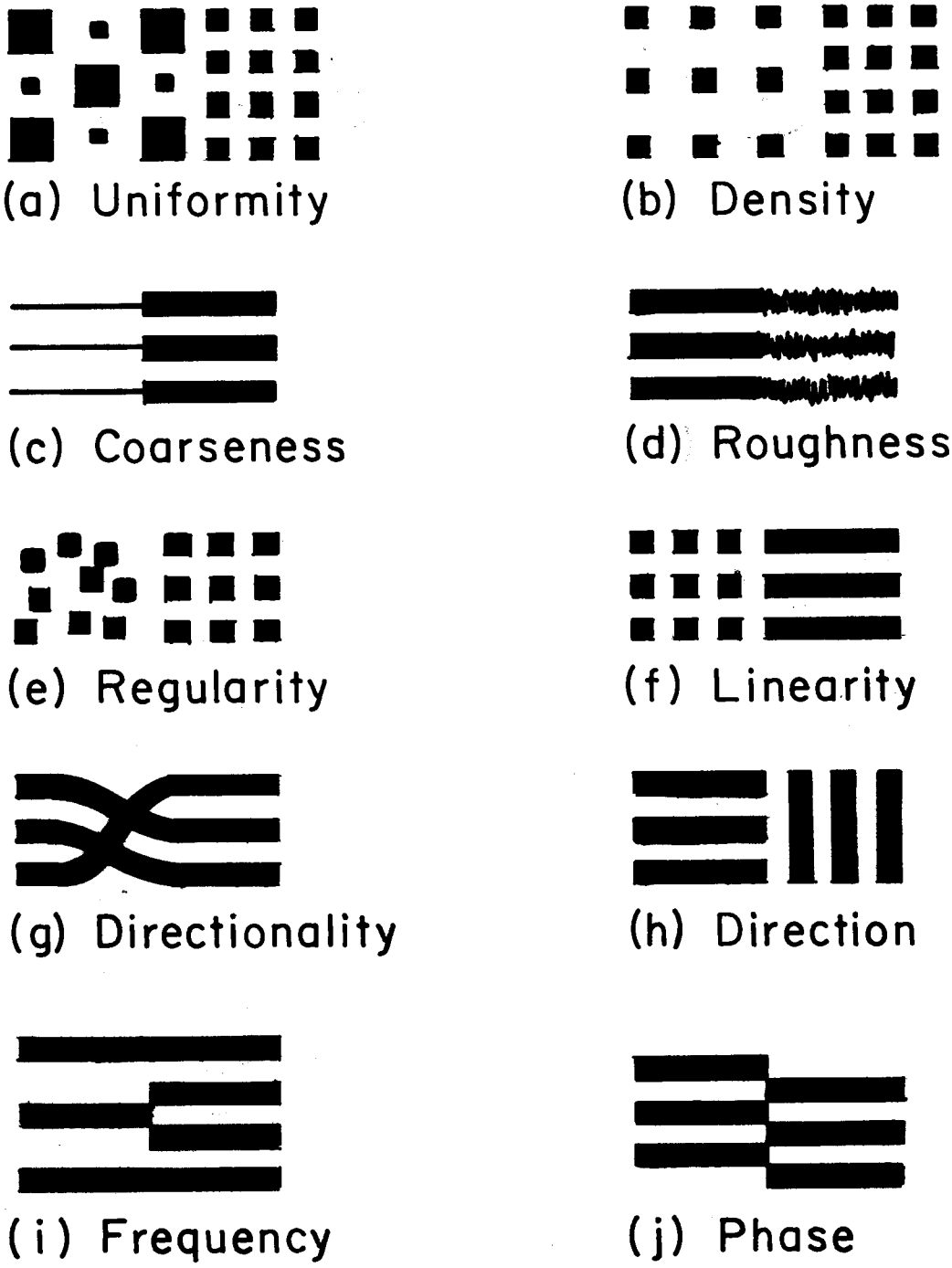


Figure 1-2. Perceptual Texture Dimensions.

textures are identical, although the two fields are easily distinguished. Similar results will be obtained with texture fields differing in contrast, color, size, rotation, or geometric warp. Texture is perceived to be invariant to changes in illumination or camera position.

We shall consider all of these differences to be differences in texture, although ones easily measured or compensated. Experimental work for this study uses monochrome images quantized to have nearly uniform gray level histograms. This compensates for any differences in illumination, sensor type, or film developing parameters.

One goal of texture analysis is discovery of texture measures that correlate well with human perception. Figure 1-2 illustrates commonly proposed structural texture dimensions. The illustrated scales are not independent: frequency is much the same as density, and coarseness is related to density and to element size (not shown). Perceptual contrast is correlated with several of these scales. Linearity is an attempt to describe element shape quantitatively. Direction clearly applies only to directional textures.

Julesz [2] has shown that the eye uses adaptive level slicing. It may group white with gray or gray with black, but it cannot group white with black. The eye can also group red with yellow and green with blue, but not red with green or yellow with blue. It seems reasonable that texture scales should have the same property.

It is debatable whether direction and phase are

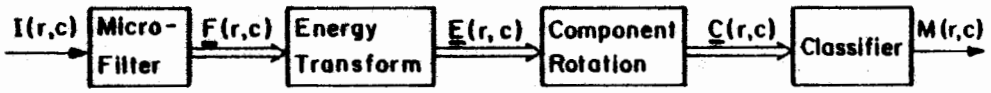
texture scales, although the texture fields are clearly discriminable. Using the criterion of shift invariance, we shall consider direction to be a texture dimension; phase is excluded. Note that phase discriminability might be due to distinctive texture properties of the region interface.

Perceptual scales such as these are useful for region description, but may have little relation to texture measures computed in the human eye or in an artificial vision system. Directionality and regularity may be high-level descriptions generated long after texture segmentation has taken place. The same may be true of shape descriptions and of color transformations such as hue and saturation.

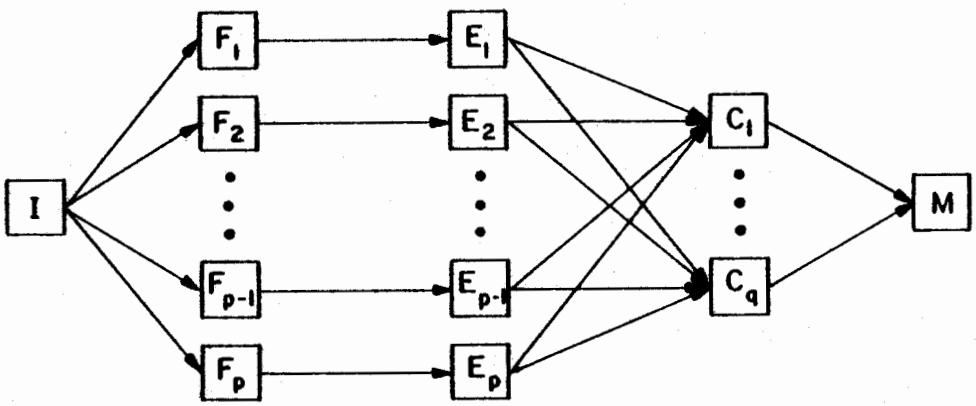
1.2 A Practical Texture Analysis System

This dissertation presents a set of "texture energy" transforms that provide texture measures for each pixel of a monochrome image. The transforms are fast, requiring only one-dimensional convolutions and simple moving-average techniques. The method is more accurate than gray level co-occurrence methods. It is local, operating on small image windows in much the same manner as the human visual system. It can be made invariant to changes in luminance, contrast, and rotation without histogram equalization or other preprocessing.

Figure 1-3 shows the sequence of images, or image blocks, used in measuring texture. The original image is first filtered with a set of small convolution masks,



(a) Operator Sequence



(b) Image Plane Sequence

Figure 1-3. Texture Measurement

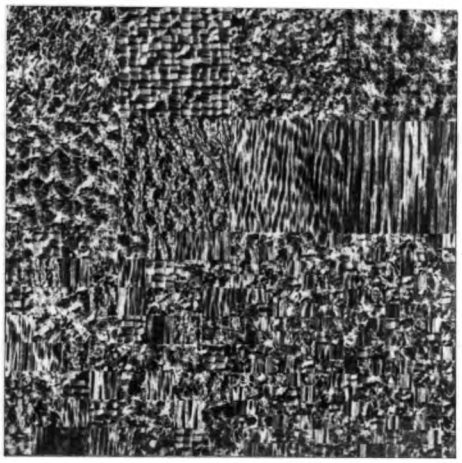
typically 5x5 masks with integer coefficients. Only one-dimensional convolution is required, since the masks are separable. The filtering could also be accomplished with multistage 3x3 convolutions.

The filtered images are then processed with a nonlinear "local texture energy" filter. This is simply a moving-window average of the absolute image values. Such moving-window operations are very fast even on general-purpose digital computers. The best window size depends on the size of image texture regions. This study has concentrated on 15x15 windows. Even smaller windows might be useful if color information were available.

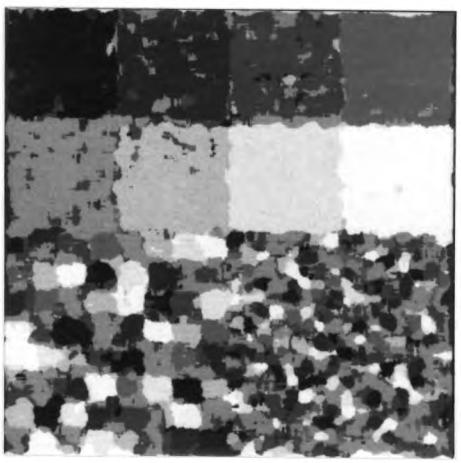
Figures 1-3a and 1-3b show a one-to-one mapping between filtered images and texture energy planes. Twelve measures per pixel were used in preliminary research. Experience has shown that either variance or standard deviation alone is sufficient to extract texture information from the filtered images.

Variance is an average squared deviation from the mean. For a zero-mean field, it is an energy measure. The standard deviation is the square root of this local energy. It may be considered a "texture energy" measure. A faster energy transform is the average of absolute values within a window. All of these texture measures give equivalent performance.

These statistics are more local than previously studied frequency-domain texture measures. Frequency components are measured with very small convolution masks.



(a) Composite



(b) Pixel Classification

Fige 1-4. Texture Segmentation Example

Phase relationships within each window are measured without regard to any global origin. This method, similar to human visual processing, is appropriate for textures with a short coherence length or correlation distance.

The next step in Figure 1-3 shows the linear combination of texture energy planes into a smaller number of principal component planes, typically three or four. This is an optional data compression step. It is tempting to call the final images "perceptual planes," but it has not yet been proven that they relate to human texture perception. They do seem to represent natural texture dimensions, and to be more "reliable" than the texture energy planes.

The final output is a segmented or labeled image. A classifier assigning texture labels to the image pixels can take either texture energy planes or principal component planes as input. Classification is simple and fast if texture classes are known a priori. Clustering or segmentation algorithms must be used if texture classes are unknown.

Figure 1-4a is a composite of the natural textures used in this study. The first two rows of 128x128 blocks are from images of grass, raffia, sand, wool, pigskin, leather, water, and wood. The lower-left quadrant is composed of 32x32 blocks, and the lower-right quadrant of 16x16 blocks. The 128x128 blocks have been individually histogram equalized; the other blocks have been equalized by quadrant. The textures were chosen precisely because

they are difficult to discriminate. They are a worst case dataset.

We have applied a simple set of texture energy transforms to the texture composite in Figure 1-4. Each pixel was then classified into one of the eight texture categories. Average classification accuracy is near 87% for interior regions of the 128x128 blocks. The 32x32 blocks are well separated, and the 16x16 blocks are differentiated to an extent. We believe this perfomance to be unmatched by any other texture classifier or image segmentation system.

CHAPTER 2

REVIEW OF TEXTURE ANALYSIS APPROACHES

Despite its importance, there is no generally accepted definition of texture. There are many models for the generation of particular texture classes [3], [4]. There are numerous ad hoc texture discrimination techniques. Yet there is no agreement on how to measure texture.

The eye must use the same feature extraction methods on every texture field, regardless of source. We do not know what these methods are, although there is indirect evidence that edge detection is involved. We do know that any retinal transform must retain enough information to distinguish different textures and suppress or ignore information distinguishing equivalent textures (as identified by human observers).

If computers could achieve the same processing results as humans, it would not matter how low-level data reduction was accomplished. It is unlikely, however, that we can ever simulate the activity of the human cortex without first learning the type of data it uses as input.

Julesz developed a basic test of human texture perception [5]-[7] in which split images of two computer generated texture fields are displayed. He found that viewers can spontaneously discriminate between textures

differing sufficiently in first- or second-order probability densities. They cannot easily discriminate between stochastic textures differing only in third-order statistics. Julesz conjectured that second-order statistics are sufficient determinants of human texture perception. This has led to the widespread belief that second order moments or spatial frequency spectra are sufficient measures of perceived texture.

The experiments were persuasive, but not conclusive. Julesz's texture fields had only four gray levels and were highly constrained. Because they were generated line by line there could be no vertical correlation. First- or second-order densities held constant for both fields had to be uniform, and when both were held constant there could be no spatial correlation whatever.

Recently Pratt, Faugeras, and Gagalowicz [8] extended this work to texture fields with multiple gray levels and controlled correlation in both spatial dimensions. Such fields can mimic natural textures reasonably well. Their experiments have supported Julesz's conjecture. Observers can discriminate such textures differing sufficiently in first- or second-order densities, but not those differing only in third order density. Furthermore, discriminable textures can be generated having common mean, variance, and autocorrelation function. Thus first-and second-order statistics may be sufficient descriptors of texture, but the mean, variance, and autocorrelation function are not.

Tamura et al. [9] have developed features correlating

well with human perceptions of natural textures. They have successfully measured coarseness, contrast, and directionality. It should be understood, however, that human observers do not interpret these words uniformly or repeatably. The texture measures are not computationally simple, and the measured concepts themselves cannot be defined independently of the observer's culture and experience.

Another perceptual modeling experiment has been devised by Zobrist and Thompson [1]. Three artificially generated textures are displayed. The viewer decides whether the first and second or the second and third are more similar. This protocol gets closer to the mechanics of texture perception, but the quantity being measured is left uncertain. Even simple changes in the spacing or shape of texture elements can alter many statistical properties of an image.

Many other types of texture measures have been proposed [10], [11]. The remainder of this section surveys the commonly used features. Later chapters will elaborate on the texture measures chosen for this study.

2.1 Statistical Features

The most powerful and appropriate statistics for a particular type of texture are those estimating parameters of the generating process. A general vision system, however, must use features common to many types of texture. One way to find such features is to model the human visual system.

Natural texture dimensions can also be discovered by studying homogeneous texture fields. Each field contains variation inherent to that texture type. Different fields have different types of variation. Discriminant analysis is an appropriate tool for identifying which are the significant variations. It is only necessary that we propose a set of texture measures; the analysis determines which linear combinations are useful.

The simplest texture properties are those based on single-point statistics. In monochrome imagery the only point property is luminance. Color images originate with an infinite number of degrees of freedom, commonly reduced to three primary responses by modern sensors. Some sensor systems record as many as 24 spectral bands.

The three primary responses are by no means the only way to record and use color data. There is a bewildering array of information-preserving color transformations [12]. Standard color coordinates systems have nonremovable singularities that can interfere with numerical analysis [13]. The human visual system seems to perform a complex mapping from spectral input to perceived color [14]. It is not known whether this transformation occurs before or after texture recognition.

A multispectral image is a vector function of a two-dimensional domain. Statistical methods may be used to classify the pixel vectors to a known set of source classes, or to cluster the vectors to determine a posteriori classes. Pointwise transformations of the pixel vectors may be used to reduce complexity of the

classifier.

Such pointwise statistical analyses lack spatial context, the essence of texture. It is true that first-order statistical properties satisfy the criterion of shift-invariance, but they are also invariant to any rearrangement of the image pixels. It is not surprising that such methods have failed to match the classification accuracy of trained humans.

Moving-window or convolution methods may be used to compute texture planes. These are continuously applied region-to-point transformations. The texture planes may be treated as additional spectral bands, introducing spatial dependencies into the analyses. We shall study these "spatial-statistical" methods in Chapters 6 and 7.

2.2 Autocorrelation Features

Texture is both spatial and statistical. It is spatial since texture is the relationship of groups of picture elements. Nothing can be learned about texture from an isolated pixel, and little from a histogram of pixel values. Monotonic transformations leave texture largely unchanged.

There is good evidence that the human visual system does not respond to spatial dependencies of higher than second order. The relationship between any two pixels may be significant, but their joint relationship with any third pixel in an image field is not. This suggests the digital autocorrelation function as a matrix of texture descriptors.

Mathematically this function is defined as

$$C(i,j) = \frac{\sum_{r,c} I(r,c)\, I(r+i,c+j)}{\sum_{r,c} I^2(r,c)}$$

It is convenient to restrict r and c, the row and column indices, to lie within the window; this is equivalent to assuming that the image function is zero outside the window. Note that i and j, the shift indices, may take negative values; the function is symmetric about the origin.

The autocorrelation function of an image measures how well the image matches a shifted version of itself. Autocorrelation is nonnegative (for nonnegative images) and takes its maximum value of 1.0 at shift (0,0). Correlation drops off exponentially with increasing shift. Typical photographs have nearest-neighbor (or single-pixel shift) correlations above 0.95. Texture blocks used in this study have nearest-neighbor coefficients near 0.70, with coefficients as low as 0.30 for some 15x15 blocks.

The autocorrelation function contains two types of information. One is texture coarseness, as revealed by the slope of the central peak. Autocorrelation of a coarse texture decays very slowly with increasing pixel separation. The other type of information concerns periodicity. Any regularity in size or spacing of texture elements will be revealed as an energy peak within the autocorrelation function. Man-made orchards and fields,

for instance, have regular spacings appearing as periodic amplitudes in the auto-correlation function.

The relationship between correlation and coarseness in seven Arctic aerial photographs was investigated by Kaizer [15]. He measured the image distance at which autocorrelation dropped to 1/e. (Circular symmetry of the autocorrelation function was assumed.) Then 20 subjects ranked the pictures in terms of coarseness. He found almost perfect agreement between 1/e distance and perceptual coarseness.

Unfortunately the autocorrelation function of most natural textures are very similar. Description of the correlation function by its first few spatial moments has little power unless correlations are measured over very large windows. This would be inappropriate in image analysis, since relatively small regions of texture must be identified.

The autocorrelation function is still being proposed as a source of texture features [8], however, and as the basis for linear-predictive texture synthesis and segmentation [16]-[18]. Usefulness of autocorrelation texture features will be explored further in Chapter 5.

A generalized autocorrelation measure is reported by Haralick [11]. It is based on the "mathematical morphology" binary filtering theory of Serra and Matheron as used in the Leitz texture analysis system [19]. Instead of summing terms of the form $I(r,c)I(r+i,c+j)$, texture is measured by summing $G(r,c)H(r+i,c+j)$, where

G(r,c) and H(r,c) are functions of the neighborhood of image point (r,c). Another way of producing the same result is to convolve functions G and H with the image, then cross-correlate the resulting feature planes. If G and H are identical, this reduces to autocorrelation of a single feature plane.

Some textures have regular structure best identified in the frequency domain. One could transform the autocorrelation function and use Fourier coefficients as texture measures. The autocorrelation function, however, is usually computed in the frequency domain by Fourier transforming the image itself. Further, the Fourier transform can be obtained optically. For both theoretical and computational reasons, frequency methods have largely supplanted correlation methods.

2.3 Spatial Frequency Features

Textures composed of repeated, regularly spaced elements are well described by their Fourier components. Natural textures are seldom so regular, but can also be descriminated by frequency domain features.

It has been shown [20] that Fourier features provide useful information for aerial classification and for identification of texture gradients. Performance of other transforms has also been investigated. Hadamard and slant transforms, for instance, have been found [21] to work as well as the Fourier for aerial classification.

Lendaris and Stanley [22] did the pioneering work in Fourier texture discrimination. They illuminated circular

sections of aerial imagery and sampled the Fraunhofer diffraction patterns cast by a lens. This diffraction pattern corresponds to the magnitude of the Fourier transform. (Neither they nor subsequent researchers seem to have investigated the Fourier phase component as a texture measure.) They integrated the transform energy over radial wedges and over concentric rings, a sampling scheme still used in some commercial systems.

Wedge features measure directionality in the original image. Linear classifiers using these features have performed well in recognition experiments, although their ability to handle rotated texture fields is open to question. Annular features have proven to be less valuable; apparently all natural images have similar spatial frequency spectra. Bajcsy and Lieberman [23] found annular components valuable for measuring element size in "blob-like" textures.

Other experimenters [24]-[26] have used digital techniques to transform texture fields. Special FFT algorithms and hardware make large transforms practical, and moving-window techniques [27] reduce the cost of repeated small transforms.

The chief difficulty with transform methods is that they must be computed over large windows. Small window transforms reveal only high-frequency information, negating the theoretical justification of the transform. Further, single frequencies are seldom important or reliable. The spectrum must usually be reduced to a

smaller number of features by computing functions of the spectrum.

2.4 Co-occurrence Features

Frequency-domain measures have little theoretical justification for randomly spaced texture elements or for small window sizes. They are also inappropriate for nonstationary textures or mixed textures within a sampling window. All of these problems exist in the segmentation of natural scenes. Correlation techniques are one way to analyze texture in the spatial domain; co-occurrence techniques are another.

A co-occurrence matrix is an estimate of the joint probability density function for pixels separated by a particular row and column shift. The i,j-th element is the number of times pixels with the luminance values i and j occur in a specified spatial relationship. Often this matrix is normalized by dividing each count by the total number of pixel pairs.

Transition probabilities are sensitive to contrast and average luminance of an image. It is therefore necessary to standardize each image or window by scaling or histogram modification. This will be discussed further in Section 3.4.

Co-occurrence approaches are an outgrowth of the Markov model of texture generation [28]-[30]. Julesz [5] was the first to use higher order transition matrices for texture synthesis. These matrices are equivalent to nearest-horizontal-neighbor co-occurrence matrices,

although normalization is applied to each row separately instead of to the matrix as a whole. Similar texture statistics have been used by Darling and Joseph [31] and by other researchers to discriminate cloud types, cell types, and textures.

Co-occurrence matrices for arbitrary row and column shift were first proposed by Rosenfeld and Troy [32] and by Haralick et al. [33], [34]. Many subsequent studies [35]-[39] have proven the value of these measures for aerial, X-ray, and microscopic imagery. Comparative studies [40], [41] have verified the superiority of co-occurrence statistics over spatial frequency and other early texture measures.

The number of co-occurrence matrices that can be computed is very large. Row shift can vary from zero to almost the number of window rows; column shift can vary over a similar range. Negative shifts are also permissible, although there are symmetry considerations. Each combination generates an entire co-occurrence matrix. For texture segmentation by pixel classification, each matrix must be computed around each image pixel. Clearly it is necessary to choose some small subset of these matrices to be computed. The best set is undoubtedly a function of the texture discrimination task.

The size of each co-occurrence matrix is also a problem. Most images are recorded with eight bits per pixel, or 256 gray levels. A few optical systems provide twelve bit resolution, or 4096 gray levels. Joint

probability matrices, however, are unreasonably large for images with more than 16 gray levels. Requantization to this number of levels conceals low contrast textures.

Haralick uses symmetric co-occurrence matrices (equivalent to averaging the matrix with its transpose). In some studies, he has reduced storage further by assuming rotational isotropy, i.e. by averaging all matrices computed for the same relative pixel shift in different directions. It has been shown [41], [42] that even the symmetry assumption is too strong for a simple Markov model of texture.

There may be an adaptive quantization scheme which retains the character of low-resolution textures. One approach is iterative histogram modification [43]. Another is to bypass the co-occurrence matrix itself. The matrix is usually reduced to a vector of features by computing two-dimensional moments. Moments that are linear functions of the matrix elements can be computed directly from the texture image. Examples are sums of probability mass along the major and minor diagonals. For such moments, the co-occurrence matrix is simply a theoretical intermediary; it need not be computed.

Individual elements of a co-occurrence matrix do not make good features: matrix elements are subject to large fluctuations due to sampling variation, the number of matrix elements is large, and sampling or unraveling of the matrix ignores the two-dimensional structure of the data. These objections can be met by using spatial

moments of the matrix as features.

Many weighted moments have been suggested. Haralick et al. [34] proposed a set of 14 moments, some later parameterized to form families of moments [11]. Pressman [38] suggested seven more moments; none were found useful. Chang [44] has suggested a principal components approach to extracting the significant information.

An entropy or conspicuousness transform has also been proposed by Haralick [45], [11]. This is one way of generating a texture plane without computing co-occurrence matrices for each point. Co-occurrence matrices are computed for pixels in a large area, possibly the entire image. Likelihood of each pixel is computed by looking up its gray level and that of its neighbors in the matrices. The likelihood, or some related function, can then be used in texture segmentation. "Common" pixels are removed as one segment, and co-occurrence statistics are then recomputed for the remaining pixels. The segments are thus identified without the necessity of classifying pixels as to texture type, much in the manner of the Ohlander segmenter [46]. These likelihood measures are similar to the conspicuousness transform of Winkler and Vattrodt [47] and the linear prediction techniques of Deguchi and Morishita [18].

2.5 Structural Features

A composite texture is one composed of primitive elements. A description of such a texture, in terms of observed primitives and their relationships, is called a

structural description. The description should be sufficiently flexible that a class of equivalent textures can be generated by using similar primitives in similar relationships.

A texture primitive is a maximal connected set of pixels having some property. Very complicated primitives have been used: Lu and Fu [48], [49] derive sets of primitives from arbitrary image windows. At the other extreme, individual pixels may be considered texture primitives.

Simple texture fields can be completely characterized by a set of primitives and a placement rule. Examples are characters of text or uniformly spaced polka dots. Sometimes the placement rule may be stochastic, as with irregularly spaced polka dots. Sometimes primitives may overlap, as with tree leaves; sometimes they add or "show through."

Primitive elements may also have stochastic attributes. They may differ in size, shape, orientation, color, or texture. These attributes may be independent or interrelated. They may be correlated with attributes of nearby primitives, and the relationships may change slowly across a nonstationary texture field.

Even in uniform texture fields, it is difficult to infer the primitive types and the placement rule. Some textures are ambiguous, with more than one choice of primitive. The most appropriate primitives are those corresponding to physical properties of the the image

source. A general vision system, however, cannot be strongly linked to a particular image source. Universal primitives must be those occurring in nearly all texture fields. Examples are maxima, saddle points, lines, edges, and regions of uniform luminance. Such "sub-primitives" are also useful in structural analysis of untextured image regions [50].

It is plausible that these elementary texture primitives are the correct level at which to define texture. Many biological visual systems contain spot and edge detectors. In fact, there is evidence that the human visual system transmits only edge information to the brain [14], [51]. It seems reasonable, then, to describe a texture by relationships of edges within it, or by relationships of lines, local maxima, etc.

The structural approach to image understanding is to locate primitives and link them together into larger structures. A less rigid approach to texture description is often used; it might be called "structural-statistical." Texture elements are identified and their properties measured, then spatial distribution of the primitive properties is described statistically.

The simplest texture measures record the observed mixture of primitives, without regard to their spatial relationships. These measures are appropriate for textures generated by randomly placed or randomly selected texture elements. It is assumed that each element is independent of its neighbors; the texture may thus be

described by its mixture density.

More complicated texture measures are needed when the primitives themselves have variable properties. We may still assume independence between primitives, but must now use a more complex probability model. It becomes very difficult to estimate the multidimensional density function of a texture field unless primitives are very numerous and simple.

We may also have to measure the spatial relationships between primitives. Variables which may be mutually dependent are the texture element types, properties, orientations, and relative spacings or relationships. It is believed that only pairwise relationships are of importance to human perception [7].

It may be sufficient to record the observed mixture of element pairs. Zucker [52] has suggested estimation of the joint probability distribution for primitive pairs in a particular spatial relationship, e.g. nearest neighbors. More powerful methods are required when texture element properties and spacings are related. It is not known how much power is needed for analysis of natural textures.

One primitive form is the maximal connected region of constant gray level. Maleson et al. [53] suggest using ellipsoidal approximations to such regions to simplify shape description. Measurable properties are size, elongation, orientation, and tonal statistics.

Galloway [54] described coarsely quantized textures

in terms of gray level run lengths. Runs were measured in several directions, each generating a matrix of gray level versus run length counts. This is similar to co-occurrence techniques. Comparative studies have shown co-occurrence measures to be superior for terrain classification [40] and characterization of Markov textures [41].

Intensity extrema are the basis of several popular texture measures. An extremum is an image pixel brighter or darker than any neighboring pixel. Several researchers [55], [56] have analyzed scan-line extrema. Measurable qualities include peak height and width, valley depth and width, inter-peak distances, and density of extrema. These quantities are not trivial to measure; several definitions are in use. The desirability of extracting features at several resolutions has led to hierarchical decompositions of scan-line waveforms [57], [11].

Texture is a two-dimensional phenomenon; it makes sense to seek two-dimensional extrema. Associated with each peak is a "mountain" or connected region that may be reached by a monotonically descending path from that peak alone. Such "reachability sets" can be computed by iterative algorithms. Texture features which may be extracted from these mountains include height, area, circularity, elongation, and direction of elongation [32].

One way to record these distributions is with generalized co-occurrence matrices [58]. Each measurable property is quantized to a small number of levels. Then

the observed traits are tabulated for all pairs of adjacent texture elements, adjacent texture elements in a given direction, or elements within a given radius of each other.

Generalized co-occurrence methods suffer from computational complexity. It is not easy to locate texture primitives and to measure their attributes, nor is it trivial to identify an element's nearest neighbors. Another weakness is that the co-occurrence matrices are quite difficult to update if the image window is shifted. This makes it difficult to compute texture properties around each image point.

2.6 Texture Segmentation

A texture measure should only be defined over a uniformly textured region. Measures computed over a multi-textured region will often be a weighted average of component texture measures, but this is not guaranteed. A homogeneity measure, for instance, will be very different for a mixed texture than for any of its components. Texture classifiers can be tricked into completely erroneous identifications by composite textures.

A texture classifier must be given regions of uniform texture over which to compute feature vectors. A segmenter must be able to find these regions without *a priori* knowledge of the textures or their context. The puzzle of how to combine these two has yet to be solved. A solution must exist, however, since biological vision systems are able to segment textured images.

Existing segmentation methods all require that region interiors be smoother than border neighborhoods. They are thus unsuitable for locating textured regions unless textures can be transformed to one or more feature planes with the property of region homogeneity. Chapter 8 will present a good method of computing such feature planes.

The constituents of texture are so many and so varied that it is difficult to combine them in a segmentation algorithm. One method [46] is to segment on the cheapest or most effective feature first, then on the next best feature. This can lead to sequence-dependent results, but is particularly effective in purposive vision systems.

A method particularly suited to texture segmentation is pixel classification, long used in analysis of multispectral LANDSAT images. Each pixel has an associated vector of spectral luminance responses. This vector can be augmented with any number of texture features computed over the immediate neighborhood of the pixel. A classification algorithm then assigns a class label to the pixel. Texture classes are usually known *a priori*, but may also be derived from the image by cluster analysis.

Suppose that we wish to classify an 8x8 image window as one of several texture types. The method of maximum likelihood could be used if we had enough information about the texture classes. We would estimate likelihood of the observed pattern under each hypothesis, then choose the texture class with highest likelihood. The trouble

with this approach is that the required probability distributions are 64-dimensional. Even for binary textures it is nearly impossible to estimate such large distributions. ($2^{64} \simeq 10^{19}$ coefficients are required for a full histogram.) The same amount of storage is needed for 4x4 blocks of 16 gray levels.

Nonparametric methods have been proposed for estimating and storing large distributions; see, for example, set covering procedures of Read and Jayaramamurthy [59] and McCormick and Jayaramamurthy [60]. It seems sensible, however, to assume a parametric form for the distributions whenever it is possible to do so.

Image gray levels seem to be well characterized by statistical moments. Ahuja et al. [61] show that the first few moments are as useful as the entire distribution for classifying image regions. Some classification procedures require that a particular parametric model be chosen (e.g. Gaussian or Poisson), but nearest-centroid techniques require only statistical moments. Chapter 8 of this dissertation will develop a texture analysis method based on nearest-centroid pixel classification.

2.7 Summary

No one has yet developed a completely adequate theory of texture analysis. Indeed, no such theory can be developed independent of the myriad physical processes producing textures. It is possible, however, to correctly segment and identify image textures using ad hoc measures and simple algorithms.

Some sets of texture measures are of more interest than others. The set used by the human visual system is of paramount importance, but not yet identified. Theoretically tractable and computationally simple feature sets are also important. Any useful set must be computable and sufficiently complete to characterize textures found in a given application area. Other desirable properties are feature independence and the ability to synthesize a texture from its feature values.

Structural methods first locate primitive elements, then analyze spatial relationships. The texture must have identifiable primitives, and the vision system must be able to determine which primitives are present. It is much harder to analyze such textures than it is to generate them. In natural images, adjoining texture fields may be obscured by noise and blur. Even with complete knowledge of texture types, it may be difficult to locate the primitives. We may have no *a priori* knowledge, making it necessary to jointly estimate the segmentation boundaries and the texture model within each segment. Such methods are too knowledge-dependent for a preliminary texture segmentation system.

The other texture models of this chapter are worthy of investigation as micro-texture measures. We shall test the efficacy of correlation, co-occurrence, and statistical methods in Chapters 5 through 6. In Chapter 7 we shall introduce several sets of texture measures which may be considered either statistical or an unusual frequency-domain approach. Chapter 8 will develop the

best of these texture measures into a texture analysis system.

CHAPTER 3

EXPERIMENTAL METHODS

An optimal vision system would have components that are jointly optimal rather than individually optimal. Unfortunately texture segmentation is too poorly understood to allow even componentwise optimization. We are faced with a chicken-and-egg puzzle: each step must be developed in the context of all others. The best we can do under the circumstances is to fix those components for which we have a rationale, and to iteratively improve all other components. Fixed choices are discussed in this chapter; experimentally determined results are given in following chapters.

3.1 Segmentation

We desire a segmentation method that is fast, insensitive to noise, and theoretically tractable. It should use little storage, work with any texture type, detect both large and small regions, and adjust for *a priori* probabilities or external knowledge.

Any segmentation method might be made to work. We shall restrict our attention to pixel classification. It satisfies all of the above requirements, provided that suitable texture measures can be found.

Two cases must be considered: true classification and blind segmentation. True classification requires that the

possible region types be known beforehand; we need simply assign a region type to each pixel. Blind segmentation is the grouping of pixels into regions without *a priori* knowledge of region characteristics. The classification approach to blind segmentation uses cluster analysis to determine the region types present, then classifies each pixel to one of these types. This could be followed by an editing phase that would attempt to assign meaningful labels to the regions.

Either case requires a classification algorithm. There are many to choose from, including nearest-neighbor, k-nearest-neighbor, maximum likelihood, and sequential decision methods. For true classification, we shall choose one of the simplest: nearest centroid classification. This algorithm is fast, easy to implement, and requires little storage. Its theoretical basis is documented in Appendix C.

The nearest centroid algorithm works well providing that suitable texture dimensions can be found. It is necessary that texture samples form well-separated globular clusters in the feature space. Elongated clusters, classes with multiple clusters, and dense clusters within sparse ones would all cause errors avoidable with more sophisticated techniques. Fortunately, the statistical technique of discriminant analysis is available to identify good features. We shall assume that optimization of the feature space is a sufficient substitute for joint optimization of the feature space and classification algorithm.

Statistical analyses are of two types: those with a known objective function and those analyzing the structure of data without regard to an objective function. The former type is characterized by work of Tamura et al. [9], in which perceptual scales for coarseness, directionality, and other features are constructed from observers' ranking of images. These scales are then matched by linear combinations of measured features. Another example is the work of Zobrist and Thompson [1], in which perceptual effects of known texture transformations are measured and modeled. The limitations of these methods lie in the experimenter's ability to invent scales measuring fundamental textural or perceptual dimensions.

The other statistical approach seeks fundamental texture dimensions in the correlation structure of the input data. This study uses discriminant analysis to identify useful features for texture classification. Discriminant analysis is a fairly well developed statistical method for choosing linear combinations of features which best classify data from a set of source classes.

Available methods are all linear analyses. Nonlinearities may be introduced by including products and quotients of texture measures, but such terms are seldom fundamental and are difficult to interpret. Of course, the analysis can be no better than the data. After studying one analysis, it is often possible to compute better features as input to the next.

Useful texture features may correspond to human visual measures or to natural texture dimensions. It has not been proven that natural texture dimensions exist, but there is evidence that humans and some lower animals have very similar perceptions of texture. It seems likely that natural texture dimensions exist and that natural vision systems have been selected and trained to use them.

Research presented here incorporates perceptual factors in three indirect ways. First is the choice of images to be used. This study uses a number of images that are visually similar, yet differing in some obvious, unspecified manner. This comes as close to a controlled experiment as can be managed with natural textures. The purpose of the experiment is to learn what features make the images visually distinct.

Second is the choice of texture measures to be computed. Some of these may be chosen for theoretical reasons, but most simply seem plausible. Some measures attempt to model anatomical processing, such as edge and spot detectors. Others are chosen to measure hypothesized differences in the selected texture images.

Third is the analysis of statistical results. Here the experimenter's subjective knowledge enters. Statistical analysis will eliminate many bad features, but may discover chance combinations of features with significant discriminating power. It is up to the experimenter to decide what is being measured by feature combinations, which of several correlated features are

most fundamental, and how to modify features to make them better.

3.2 Feature Selection

Classification accuracy is a function of the number of features available and the joint information of those features. It is also a function of the method used to select or combine features.

The primary tool of this research is discriminant analysis. Feature vectors computed over image windows are fed to the discriminant routines of the Subroutine Package for the Social Sciences (SPSS). These descriminant algorithms are documented in Appendix C. Source textures are known, so that cluster analysis is unnecessary. The goal is similar, however: to find linear combinations of features that separate data vectors into compact groups.

One could search for fundamental texture features by analyzing differences between pairs of images. It is likely, however, that each pair differ along a combination of fundamental dimensions. The analysis might identify some discriminating features, but would leave unclear the nature of the true texture dimensions.

Analyzing many textures at once is more likely to discover fundamental dimensions, if they exist. Discriminant routines identify the best axis, then the best orthogonal axis, and so on. The axes are best in the sense of a Karhunen-Loeve or eigenvector coordinate transform. It is quite likely that the human visual system uses correlated feature measures, but the expense

of such an analysis is not justified by the quality of our present texture descriptors.

Discriminant functions, computed as eigenvectors of certain statistical matrices, serve three purposes. They identify natural data dimensions, permit data reduction for simpler classification functions, and provide natural axes for visual display of clusters. A display of data points in the primary discriminant plane conveys a great deal of intuitive information difficult to discern in tables of numbers.

A more quantitative description is provided by the weights of features used to compute the axis values. These coefficients are given for input variables normalized to zero mean and unit variance. The coefficients thus show relative weight or importance of each component feature.

Computed texture dimensions must be judged by their ability to classify the input vectors. Technically it would be better to classify an independent set of texture vectors, but classification of the training set is a useful experimental tool. More rigorous validation need be applied only to the final texture model.

Two data clusters in a multivariate space are maximally separated along a single axis. Three clusters can be discriminated in a plane, i.e. along two axes. The number of possible discriminant axes is one less than the number of groups. The number of useful discriminant functions may be even smaller if data clusters tend to

line up or occupy low-dimensional subspaces.

Classification functions, one for each texture group, can be derived from the discriminant functions. A data vector may be classified by evaluating each function and assigning the vector to the group with the highest score. The method assumes multivariate normal distributions with identical covariance structure. Prior probabilities for the classes are usually assumed equal.

3.3 Texture Data

In an experimental study, the results can be no better than the input data. We require a set of uniform texture fields large enough to provide adequate samples of each texture. Ideally this training set should come from a target application area. For a general vision system, however, each texture must be a "natural" one, and the set must include a range of natural texture dimensions. We avoid artificially generated textures, such as sinusoidal gratings, because they would favor the Fourier transform and other frequency domain measures.

The texture images we have chosen are from an album by Brodatz [62]. High quality prints obtained from the photographer were scanned and digitized at the USC Image Processing Institute. The images are 510x512 pixels quantized to 256 gray levels. This is sufficient for extraction of 256 nonoverlapping 31x31 blocks from each texture field. Most of the texture samples in this study will be 15x15 feature plane blocks computed within 17x17 or 19x19 blocks of image data. The larger image window is

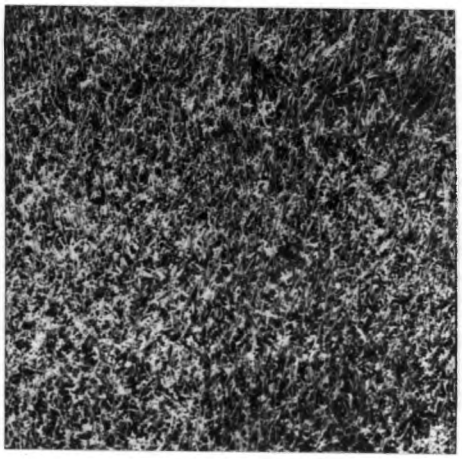
used only to prevent contamination of the samples by border effects, and is unnecessary when texture measures are computed for every pixel in an image.

Initial data analyses for this study were carried out on the four Brodatz textures in Figure 3-1: Grass, Raffia, Wool, and Sand. Pratt et al. used 64x64 blocks of these same images for visual discrimination experiments [8] and for theoretical discriminability studies [63]. Ashjari [64] has investigated singular value decomposition as a tool for discriminating 32x32 blocks of these textures. Additional texture dimensions have been introduced with the textures in Figure 3-2: Pigskin, Leather, Water, and Wood.

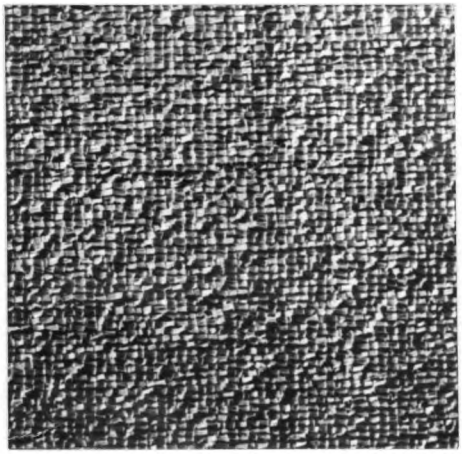
The textures have been chosen precisely because they are difficult to discriminate. They are a worst case dataset. Raffia, Wool, and Sand may be considered cellular textures with similar cell sizes. Grass and Sand have similar statistics, with the main difference being the extended edges in Grass. Pigskin has statistics similar to those of Sand, but lacks the cellular edge structure. Leather has edge structure similar to Grass, although the textures are perceptually quite different. The Wood and Water images have much stronger vertical structure than Grass.

3.4 Preprocessing

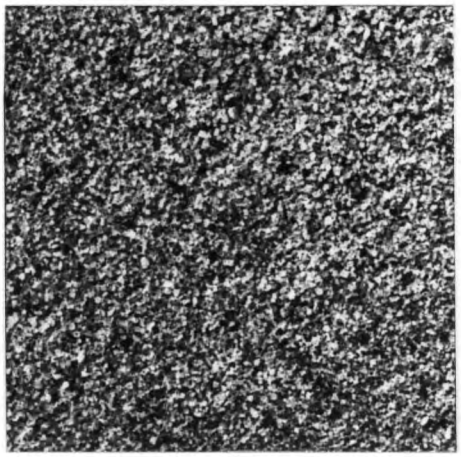
The texture images were not taken under completely controlled conditions. They differ in illumination, contrast, and possibly film type or developing process.



(a) Grass



(b) Raffia



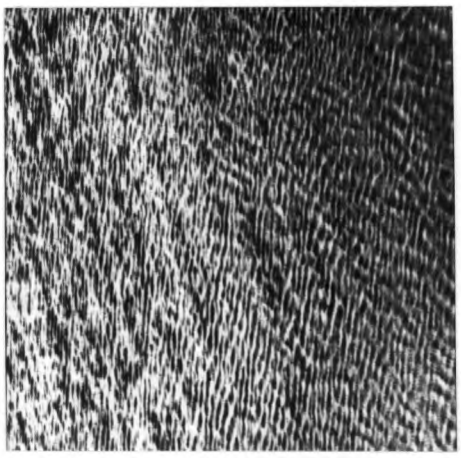
(c) Sand

(d) Wool

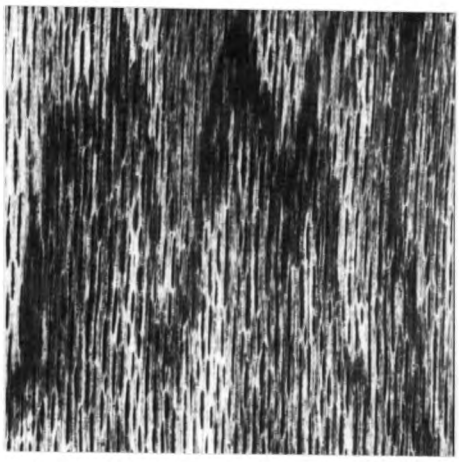
Figure 3-1. Experimental Textures

(e) Pigskin

(f) Leather



(g) Water



(h) Wood

Figure 3-2. Additional Textures

These differences introduce monotonic transformations of the image function, and we must design our texture analysis system to be invariant to them. We shall not worry, however, about spatial transformations such as geometric warp and linear filtering. The removal of known warps is well understood, but estimation of spatial transformations from texture data awaits a better understanding of texture.

There are two approaches to compensating for unknown monotonic transformations. One is to alter the entire image, reducing it to some canonical form. The other is to develop texture measures invariant to monotonic transformations.

We have chosen a compromise technique: histogram equalization [65], [66] of the entire image coupled with texture measures compensating for local mean and standard deviation. This partially corrects for an effect noted by Sklansky [67]:

> Most images are dominated by low frequencies that carry little information about the scene. These low frequencies consume a large range of gray level quantization cells with little benefit to the viewer. Hence before any histogram transformations are carried out it is useful to suppress (but not eliminate) the low spatial harmonics... [p. 240]

The texture fields used in this study are sufficiently uniform that prior filtering would gain little.

There are several rationales for histogram equalization. Sklansky sees it as an equalization of local contrast across an image. Other authors have

considered it a maximum entropy transform since it maximizes the amount of information conveyed by a given number of gray levels. Certainly the transformation improves the appearance of low contrast images, but this is true even if the number of gray levels (hence the information content) is greatly reduced. Frei [68] found histogram hyperbolization even more visually pleasing; it is believed that this shape is converted to a uniform histogram by the logarithmic response of the human eye. Ashjari [64] uses histogram Gaussianization to prepare texture data for classifiers based on Gaussian assumptions.

It should be noted that such standardization sacrifices information. Sklansky [67] reports:

> We have found that certain diagnostically significant textural features in xeromammograms are strongly related to infrequently occurring gray levels in the tails of certain shapes of histograms. Because these gray levels occur infrequently, histogram equalization inhibits rather than enhances the extraction of these features. [p. 243]

Conners and Harlow [69] found, however, that histogram equalization was essential for proper analysis of radiographic images.

Images normalized to a common mean and standard deviation are easily discriminated by their skewness and kurtosis measures. We have applied histogram equalization to remove all first-order differences. This also finesses the problem of whether to measure image luminance or density, since the standardization will give the same result for either.

Our histogram equalization routine is given in Appendix A. It follows Conners' algorithm [41], modified to fit new quantization levels to a constant percentage of total probability rather than a percentage of remaining probability. For natural images this algorithm works well, although it will give slightly different results when starting from one end of a histogram than it would if started from the other end. It is possible to construct pathological cases for which the mean square error compared to a true uniform histogram is much greater than for optimal equalization as found by a search algorithm [70].

Global equalization is valid for experimental studies on reasonably homogeneous texture images. A general vision system, able to identify textures in scenes with varying illumination, requires stronger equalization. Either the computed texture measures must be invariant to luminance and contrast changes, or adaptive local equalization must be used.

This study uses a simple adaptive equalization. First global equalization is used, then each sampled texture window is scaled to have a constant mean and standard deviation. The method is not suitable for moving-window equalization around each image point, but the same effect could be achieved with luminance-invariant and contrast-invariant texture measures. Texture discrimination results will be reported for both the globally equalized and the adaptively equalized texture samples. There should be little difference if the source

images are homogeneous.

3.5 First-Order Statistics

A texture field is an extended entity composed of repetitions of similar local primitives. We require, therefore, global measures of local properties. These global measures must be statistical since they must be shift-invariant and insensitive to random texture variations. They should also be easy to compute since large windows are involved.

Global features characterize the whole texture rather than its elements. The computing window must be large enough to enclose a representative sample of the texture, so that feature values change little as the window is shifted within a texture region.

The set of statistical moments are particularly good global measures. Consider a window placed on an image, or on any feature plane computed as a transform of the image. One likely texture measure is the average value within the window. Another is the standard deviation. Skewness and kurtosis are also good candidates, although somewhat harder to explain. It is known that the histogram of an 8-bit feature plane can be completely characterized by a set of 256 such statistics. Statistical moments above the fourth, however, are likely to be unreliable and to have little energy or importance. This study will determine whether the first four moments are useful.

The basic statistical moments of a window are

$$Mk = E\,[I^k(r,c)]$$

where E denotes the expectation operator. The moments may be estimated by

$$Mk = (1/n^2) \sum_{r,c} I^k(r,c)$$

It is convenient to standardize higher moments to remove the effect of mean and standard deviation. Statistical moments used in this study are of the form

$$AVE = E\ [I(r,c)]$$

$$VAR = E\ [(I(r,c)-AVE)^2]$$

$$SKW = E\ [(I(r,c)-AVE)^3 / VAR^{3/2}]$$

$$KRT = E\ [(I(r,c)-AVE)^4 / VAR^2]$$

These corrected moments may be estimated by

$$AVE = M1 \tag{3-1}$$

$$VAR = M2-M1^2 \tag{3-2}$$

$$SKW = (M3-3(M1)(M2)+2M1^3) \ / \ VAR^{3/2} \tag{3-3}$$

$$KRT = (M4-4(M1)(M3)+6(M1^2)(M2)-3M1^4) \ / \ VAR^2 \tag{3-4}$$

The following transformations and block statistics will also be used as first-order statistics:

$$SDV = \sqrt{VAR} \tag{3-5}$$

$$ACV = SDV \ / \ |AVE| \tag{3-6}$$

$$ASK = |SKW| \tag{3-7}$$

$$AKR = |KRT-3.0| \tag{3-8}$$

$$MIN = \min_{r,c} I(r,c) \tag{3-9}$$

$$MAX = \max_{r,c} I(r,c) \tag{3-10}$$

$$RNG = MAX-MIN \tag{3-11}$$

$$MID = (MAX+MIN) / 2 \qquad (3\text{-}12)$$

The most fundamental first-order statistic is the average. Histogram-equalization renders it useless on the original image, but it is useful on feature planes computed from the image. Computing the moving-window average is equivalent to blurring or lowpass filtering the feature plane.

Variance and standard deviation measure the irregularity in a feature plane. These are important features, and it is not known *a priori* which form is more fundamental. Using both forms permits a linear analysis to approximate nonlinear functions of the standard deviation. Absolute coefficient of variation (**ACV**) is also provided; for nonnegative distributions it is often a better dispersion measure than the standard deviation.

Other moments may also be useful. Skewness measures the extent to which outliers favor one side of the main distribution. Kurtosis measures peakedness or the presence of outliers: the kurtosis of a uniform distribution is 1.8, that of a Gaussian is 3.0. Absolute skewness and the absolute deviation of kurtosis from the Gaussian value (sometimes known as the "excess") are also computed. Care has been taken to prevent computational problems when the standard deviation is near zero: skewness is set to zero and kurtosis to three. Large values are also prevented by clipping both measures at plus and minus six.

The last four first-order features are the minimum,

maximum, range, and midrange of the window. Although common descriptors of uniform distributions, these statistics are included primarily because of their computational simplicity.

Computation of the twelve statistics at every picture point can be done in a single pass. On a PDP KL/10 this takes two minutes for a 512x512 image, regardless of the moving window size. The number of image rows kept in core is equal to the number of rows in the window. Each pixel is examined only twice. A similar algorithm for computing moving absolute averages is documented in Appendix B.

3.6 The F-Ratio Feature Strength Measure

Throughout this dissertation, it will be necessary to compare the discriminating powers of different features. We could compare classification accuracies for the individual features, but an impractical amount of computing time would be needed. A simpler comparison statistic is the F-ratio. It is the ratio of inter-class variance to intra-class variance.

A good feature will have a cluster of values for samples from one texture field, and a different cluster of values for another texture field. Good features therefore have high F-ratios. Actual values will not be important here, but ratios with the same degrees of freedom (i.e. sampled populaton sizes) may be compared.

F-ratios listed in this and following chapters are for 256 15x15 samples from each of the eight textures. The F-ratios have degrees of freedom 7 and 2040, making

the probability less than 0.001 that a feature with no discriminating power will have a ratio above 3.47. In practice we find that ratios below 100 are of little value. All discriminant functions and classification accuracies cited in this study will be based on variables with F-ratios of at least 40 after adjusting for all other variables in the model. The probability of a variable having a ratio this large by chance is less than 10^{-50}.

3.7 Image Block Statistics

Table 3-1 shows the effects of various standardization procedures on first-order information. The table lists the F-ratio for each statistic, a measure of its discriminating power for this set of textures. F-ratios in the first column are for the original images, before any type of standardization. It is apparent that the texture fields are easily discriminated by their means, variances, ranges -- in fact, by any of the first-order statistics.

The last entry in the column shows that all twelve features used together provide 85% classification accuracy. It can be seen that even F-ratios above 2000 do not guarantee perfect classification. A high ratio shows that class means are separated along the feature dimension. It does not mean that all classes are separated, however. Classification accuracy is a better indication of multiclass separation.

The second column is for adaptively standardized images. The pixels in each window were adjusted to have

TABLE 3-1. IMAGE STATISTIC F-RATIOS

Feature	Original	Original Adaptive	Global	Adaptive
IMGAVE	651	593	0	3
IMGVAR	1555	497	42	58
IMGSKW	625	595	6	9
IMGKRT	439	376	57	63
IMGSDV	1882	477	47	57
IMGACV	1593	554	5	54
IMGASK	502	461	40	30
IMGAKR	152	196	28	66
IMGMIN	1449	400	12	2
IMGMAX	386	619	59	10
IMGRNG	2004	473	68	7
IMGMID	575	637	34	7
Accuracy	84.81%	50.39%	19.82%	22.27%

mean 127.5 and standard deviation 73.9, then were clipped to the range 0.0 - 255.0. The table shows that this standardization reduces discriminability of the textures, although the power of some first-order features is increased. Joint classification accuracy is reduced to 50%. This adaptive algorithm apparently does not work well for grossly different first-order distributions. The clipping step emphasizes differences in skewness and kurtosis; it also translates them into differences in mean, variance, and other first-order features.

The third column shows results of histogram equalization on the original images. The procedure has

little effect on perceived texture[1], but reduces first-order discriminability. Classification accuracy for the set of 12 features drops to 20%. Equalization has removed nearly all first-order differences among the images. Texture information is evidently contained in second-order statistics of the equalized images.

The fourth column corresponds to histogram equalization followed by adaptive standardization. This is a form of adaptive histogram equalization. The discriminating power of several features increases slightly, apparently because of the nonlinear clipping effect. Joint classification accuracy remains nearly the same, 22%.

The above statistics show that histogram equalization is a useful preprocessing technique for removing first-order image differences. Such processing may not be needed in a calibrated texture recognition system, but is essential for texture research with uncalibrated images. All images used in this study have been histogram equalized. Texture measures have also been computed for the adaptively equalized case since this additional standardization is likely to be needed when classifying small texture patches within natural scenes. Here the adaptive standardization has been performed by brute force scaling of the image windows. It could also be

---

[1]All pictures in this document have been equalized. The only perceptual changes are an increase in contrast and possibly a change in average brightness.

accomplished by algebraic adjustment of computed texture measures.

Note that the minimum classification accuracy under this experimental paradigm is about 20%. Random classification of eight textures would produce 12.5% accuracy, but classification using random features may do better. This is because the best combination of features is chosen *a posteriori*. These features must give at least 12.5% accuracy, and will do significantly better if training images have exploitable differences. Even identically distributed random fields can appear statistically discriminable if the number of samples per texture field is less than three times the number of independent features. This study guards against false significance by using 256 samples per texture and a minimum F-ratio of 40.

3.8 Comparative Measures

To judge the quality of newly developed texture measures, it is desirable to apply them to the same data used by other investigators. Unfortunately no common database exists. We have implemented co-occurrence and correlation texture measures and have applied them to the Brodatz textures, each technique using the same 15x15 window size. Each algorithm has been optimized to a reasonable extent, but there can be no guarantee that a faster or more powerful version could not be found.

CHAPTER 4

CO-OCCURRENCE METHODS

This chapter investigates co-occurrence texture measures, seemingly the most effective and widely used of existing texture analysis techniques. The relative discriminating power of individual co-occurrence features will be measured, which is itself an important contribution. We will also determine joint classification accuracy on our dataset using all of the co-occurrence features; this will establish a lower bound for acceptable performance of other approaches.

4.1 Co-occurrence Measures

Co-occurrence matrices are a popular source of texture features. For this study we generate each co-occurrence matrix from a 15x15 source window requantized to 32 gray levels. Each matrix is thus 32x32. Nine of these matrices are used, corresponding to horizontal and vertical spacings of zero, one, and seven pixels. The chosen spacings correspond to horizontal, vertical, and top-left to bottom-right diagonal directions. The P00 matrix records first-order information: all the entries are on the diagonal. The other eight matrices record second-order information. The matrices are not symmetric, nor is there any averaging across different co-occurrence angles.

Many ways have been proposed for extracting texture information from co-occurrence matrices. The commonly studied moments are called contrast[2], inverse difference moment, angular second moment, entropy, and correlation. The formulas are

$$CON = \sum_{r,c} (r-c)^2 \, P(r,c) \tag{4-1}$$

$$IDM = \sum_{r \neq c} \frac{P(r,c)}{(r-c)^2} \tag{4-2}$$

$$ASM = \sum_{r,c} P^2(r,c) \tag{4-3}$$

$$ENT = - \sum_{r,c} P(r,c) \log P(r,c) \tag{4-4}$$

$$COR = \sum_{r,c} \frac{(r-AVE_r)(c-AVE_c)P(r,c)}{(SDV_r)(SDV_c)} \tag{4-5}$$

where

$$AVE_r = \sum_{r,c} (r) P(r,c)$$

$$SDV_r = \sqrt{\left(\sum_{r,c} (r-AVE_r)^2 P(r,c) \right)}$$

Rectilinear and diagonal moments of the matrices will be used as texture measures, as well as the ad hoc moments of Equations 4-1 through 4-5. The rectilinear (horizontal and vertical) moments of a matrix are

---

[2]Tamura et al. [9] found no correlation between Haralick's CON moment and perceptual contrast. The designation has become standard, however.

$$Mij = (1/n^2) \sum_{r,c} r^i \, c^j \, P(r,c) \qquad (4\text{-}6)$$

where P is the co-occurrence matrix and row and column indices are computed relative to the matrix center.

Co-occurrence matrices have diagonal structure. It makes sense to measure energy distribution relative to the diagonals. Spatial moments in this orientation can be measured by

$$Dij = (1/n^2) \sum_{r,c} (r + c)^i \, (r - c)^j \, P(r,c) \qquad (4\text{-}7)$$

Diagonal moments may also be computed from the rectilinear moments. For instance:

$$D22 = M40 - 2(M20)(M02) + M04$$

Both rectilinear and diagonal moments will be tested as texture features. Each spatial power will take values from zero to two. Since the M00 and D00 moments are identical, there are 17 moment features. The Haralick, rectilinear, and diagonal moments computed for each of nine co-occurrence matrices generate 172 independent features.

4.2 Co-occurrence Results

Table 4-1 lists F-ratios for the common Haralick moments of Equations 4-1 to 4-5. Only angular second moment and entropy features are listed for the P00 matrix, since the others are identically zero. It is interesting that P70 features have much more discriminating power than P07 features. Evidently this texture set differs more in its vertical statistics than in its horizontal statistics.

TABLE 4-1. HARALICK STATISTIC F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
P00ASM	60	46	**P00ENT**	103	65
P01ASM	17	30	**P10ASM**	55	99
P01CON	168	275	**P10CON**	744	681
P01COR	297	297	**P10COR**	644	632
P01IDM	290	326	**P10IDM**	687	292
P01ENT	71	71	**P10ENT**	278	239
P11ASM	15	17	**P77ASM**	45	43
P11CON	38	32	**P77CON**	12	5
P11COR	34	36	**P77COR**	6	6
P11IDM	31	41	**P77IDM**	10	3
P11ENT	62	44	**P77ENT**	68	62
P07ASM	65	68	**P70ASM**	65	101
P07CON	24	15	**P70CON**	241	304
P07COR	14	14	**P70COR**	267	264
P07IDM	16	6	**P70IDM**	355	213
P07ENT	123	105	**P70ENT**	157	143
P17ASM	64	64	**P71ASM**	41	43
P17CON	23	11	**P71CON**	82	80
P17COR	8	8	**P71COR**	57	58
P17IDM	16	4	**P71IDM**	64	35
P17ENT	117	97	**P71ENT**	83	65

This may be due to vertical structure of the Leather, Wood, and Water images. **P77** moments are also weak, probably because this training set has no diagonally streaked textures. Note the power of **P01** and **P10** features. Weszka *et al.* [40] also reported the dominance of local co-occurrence features, and of local features in general. They found that large-lag co-occurrence features work best if computed on blurred images, but we have not used blurred images in this study.

Table 4-2 shows classification accuracies available with various feature sets. The first analysis uses only the ad hoc Haralick moments. Together, the 32 features perform better than the best combination of the last chapter. The globally equalized textures have two dominant discriminant functions using **P10CON**, **P01IDM**, **P70IDM**, **P11CON**, **P01CON**, **P10IDM**, **P10COR**, and **P11COR**. Discriminant functions for the adaptively equalized textures use **P10CON**, **P01IDM**, **P70CON**, **P11CON**, **P01CON**, and **P71COR**. Angular second moment, correlation, and entropy features apparently carry little texture information.

TABLE 4-2. CO-OCCURRENCE CLASSIFICATION ACCURACY

Feature Set	Global	Adaptive
Haralick Moments	70.85	67.58
Rectilinear Moments	63.04	65.92
Diagonal Moments	56.60	63.04
Combined Moments	72.07	68.16

The second and third analyses in Table 4-2 use the rectilinear and diagonal moments, respectively. These are the same moments computed on the autocorrelation matrices of the previous section. Neither set is as powerful as the Haralick moments. The first set of discriminant functions are built primarily of **M11** and **M22** moments, the second uses only **D22**, **D02**, and **D20** moments. These facts apparently reflect the diagonal symmetry of the co-

TABLE 4-3. CO-OCCURRENCE MOMENT F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
P00M02	1	59	P00D02	2	1
P00M11	1	59	P00D11	0	0
P00M20	1	59	P00D20	1	59
P01M02	1	45	P10M02	1	50
P01M11	30	281	P10M11	30	238
P01M20	1	43	P10M20	1	49
P01M22	8	202	P10M22	8	148
P11M02	1	40	P77M02	1	6
P11M11	10	41	P77M11	13	8
P11M12	1	41	P77M12	1	3
P11M22	1	51	P77M22	1	14
P07M02	1	45	P70M02	1	14
P07M11	25	281	P70M11	187	257
P07M22	1	40	P70M22	19	97
P17M11	24	8	P71M11	71	65
P01D02	168	275	P10D02	744	681
P01D12	34	40	P10D12	33	50
P01D20	8	195	P10D20	7	92
P01D22	151	287	P10D22	719	552
P11D20	3	47	P77D20	7	13
P07D02	24	15	P70D02	241	304
P07D20	12	21	P70D20	73	187
P07D22	6	18	P70D22	185	138
P17D02	23	11	P71D02	82	80
P17D20	12	11	P71D20	29	48

occurrence matrices. Note that D02 moments are identical to the Haralick CON moments. Table 4-3 shows the discriminating power of individual rectilinear and diagonal moments computed on the co-occurrence matrices. Only those moments with ratios above 40 are listed. It is possible, but rare, for features with lower individual F-ratios to enter the discriminant model after the first step.

The fourth analysis uses all of the co-occurrence features together[3]. Classification accuracy is improved slightly. The strongest of the globally equalized features, P10CON, is later dropped from the model. The remaining features are P01IDM, P70IDM, P11CON, P01CON, P10COR, P10D22, and P01COR. The adaptively equalized features are P10CON, P01IDM, P70CON, P11CON, P01CON, and P71M11. Both sets identify two dominant texture dimensions. Scatter diagrams of sample points against the first two principal axes look very similar to plots for the different moment types individually. The patterns are also similar to those found with Laplacian and Sobel features, although clusters are better separated. The first discriminant function separates the directional textures, Wood and Water, from the rest. The second function separates Raffia from Wool and Leather. Least

---

[3]Some features had to be omitted from the analysis because of an SPSS limit of 100 variables. All features with F-ratios above 40 and all features appearing in previous discriminant functions were made available, as well as the maximum allowable number of less important features.

separated textures are Grass, Sand, and Pigskin.

4.3 Summary

Joint classification accuracy for these measures is 68%, or 72% for globally equalized textures. This is far better than the 33% achieved with the correlation and Markov statistics of the last chapter, and somewhat better than the 65% possible with Laplacian and Sobel statistics.

The features of greatest use are the Haralick **CON**, **IDM**, and **COR** moments. The strength of these measures is not surprising, considering their evolution over nearly a decade. It is surprising that the full set of 172 co-occurrence features has no more power than the 42 Haralick moments. Evidently there is nothing to be gained by studying new ways of extracting texture from co-occurrence matrices.

CHAPTER 5

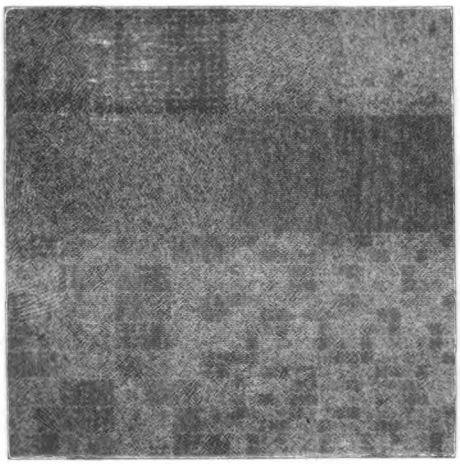
CORRELATION METHODS

This chapter presents a particular method of texture measurement based on autocorrelation statistics. The model will be developed only as far as seems necessary to determine the efficacy of correlation statistics as texture measures. Classification accuracies achieved with correlation methods will be cited in later chapters as standards of comparison. The best individual features will be carried forward into the texture models of Chapter 6.

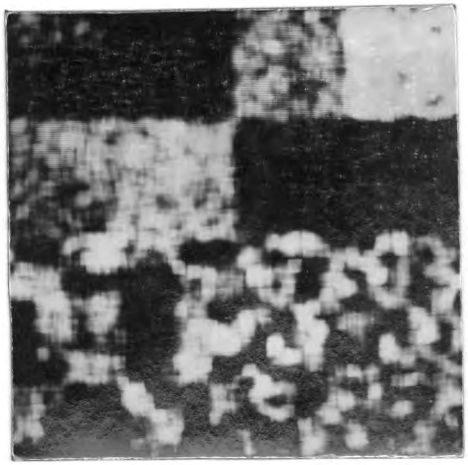
5.1 Correlation Measures

It was mentioned in Section 2.2 that the autocorrelation function is not a sufficient texture descriptor. Discriminable textures can be constructed with identical first-order statistics and autocorrelation functions.

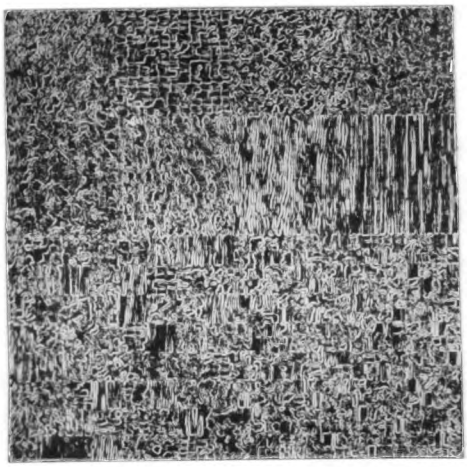
Faugeras and Pratt [63] have developed a new class of texture measures that go beyond autocorrelation information. They apply a whitening filter to the texture field, then measure the first-order statistics of the decorrelated image field. These statistical moments and moments of the original autocorrelation function form a set of texture features. It is possible to mimic the original texture by generating a random field with the



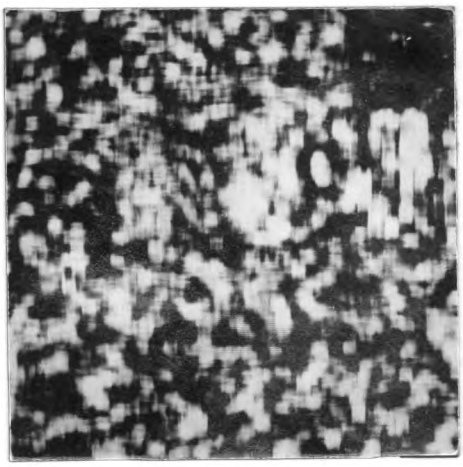
(a) Laplacian



(b) **LPLSDV**



(c) Sobel Magnitude



(d) **SBLSDV**

Figure 5-1. Transformation Examples

same moments and applying the inverse of the whitening filter. The features extracted from several natural textures have been compared using a Bhattacharyya measure; results imply good classifying power with a very small number of features.

The full whitening operation is very expensive to compute. Faugeras and Pratt suggest that the image be convolved with the Markov process whitening mask (**MKV**):

$$\frac{1}{(1-R^2)(1-C^2)} \begin{bmatrix} RC & -C(1+R^2) & RC \\ -R(1+C^2) & (1+R^2)(1+C^2) & -R(1+C^2) \\ RC & -C(1+R^2) & RC \end{bmatrix} \qquad (5\text{-}1)$$

where **R** and **C** are the horizontal and vertical nearest-neighbor correlation coefficients. This operator will completely decorrelate a Markov field for all lags greater than one. Nearest-neighbor coefficients are scaled by -0.5 and diagonal-neighbor correlations are scaled by 0.25.

The **R** and **C** coefficients for 15x15 blocks of the Brodatz textures range from 0.30 to 0.95, with the average near 0.70. As the correlation coefficients approach unity, the numerator of the whitening operator approaches a Laplacian operator:

$$\mathbf{LPL} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad (5\text{-}2)$$

Figure 5-1a is the convolution of this mask with the

composite texture image of Figure 1-4b. Figure 5-1b is the result of computing the standard deviation in a 15x15 window around each pixel in the Laplacian image. This and other feature planes will be evaluated in the next section.

Another 3x3 operation suggested by Faugeras and Pratt is the Sobel gradient magnitude. It is considered an edge detector rather than a whitening operator, but empirical evidence supports its use in texture discrimination. The Sobel gradient is a 3x3 nonlinear operator weighted toward the window center but omitting the actual center pixel. The Sobel masks are

$$x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

For each image position the Sobel magnitude is computed as the root-mean-mean-square of the two weighted pixel sums:

$$SBL = \sqrt{x^2 + y^2} \qquad (5\text{-}3)$$

This measure has been shown [71] to locate gray level step edges about as well as any other popular edge detector.

Figure 5-1c shows the Sobel gradient magnitude for the composite image. This operator emphasizes edge structures in the texture fields. The 15x15 standard deviation, shown in Figure 5-1d, is obviously less useful than the Laplacian standard deviation.

5.2 Correlation Results

The texture feature set we shall use consists of moments of the autocorrelation function plus first-order statistics of the Markov whitened image. Laplacian and Sobel gradient magnitude operators will also be tried in place of the Markov decorrelation operator. Texture features based on these 3x3 operators should be less powerful than the adaptive Markov features.

We shall extract texture information from the correlation matrices by computing spatial moments. The rectilinear and diagonal moments are of the same form as in Equations 4-6 and 4-7. Since the **M00** and **D00** moments are identical, there are 17 correlation features. The twelve first-order statistics will also be computed for each texture block "whitened" with the Markov, Laplacian, or Sobel operators, for a total of 53 independent features per texture block.

Table 5-1 shows the discriminating power of individual features. It can be seen that moments of the correlation function are very weak texture measures. The Laplacian operator generates some very powerful texture measures. Statistics of Markov whitened fields have much less discriminating power, although kurtosis and absolute kurtosis features are moderately good.

Table 5-2 shows classification accuracies achieved with various subsets of these texture features. The first three rows correspond to features extracted from autocorrelation matrices of the 15x15 windows. Each

TABLE 5-1. CORRELATION STATISTIC F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
CORM00	15	15	CORD00	15	15
CORM01	15	17	CORD01	10	11
CORM02	22	21	CORD02	6	6
CORM10	16	14	CORD10	18	17
CORM11	23	23	CORD11	47	46
CORM12	7	10	CORD12	7	5
CORM20	14	14	CORD20	24	24
CORM21	9	9	CORD21	5	4
CORM22	17	17	CORD22	15	15
MKVAVE	65	74	LPLAVE	1	1
MKVVAR	24	31	LPLVAR	707	609
MKVSKW	23	17	LPLSKW	22	16
MKVKRT	240	242	LPLKRT	251	250
MKVSDV	51	70	LPLSDV	851	700
MKVACV	1	1	LPLACV	1	1
MKVASK	29	29	LPLASK	48	46
MKVAKR	248	253	LPLAKR	261	263
MKVMIN	44	60	LPLMIN	429	374
MKVMAX	38	53	LPLMAX	512	444
MKVRNG	42	58	LPLRNG	571	488
MKVMID	6	7	LPLMID	13	11
SBLAVE	84	64	IMGAVE	0	3
SBLVAR	53	156	IMGVAR	42	58
SBLSKW	79	77	IMGSKW	6	9
SBLKRT	54	49	IMGKRT	57	63
SBLSDV	55	159	IMGSDV	47	57
SBLACV	112	105	IMGACV	5	54
SBLASK	79	77	IMGASK	40	30
SBLAKR	20	34	IMGAKR	28	66
SBLMIN	14	9	IMGMIN	12	2
SBLMAX	50	29	IMGMAX	59	10
SBLRNG	48	33	IMGRNG	68	7
SBLMID	50	25	IMGMID	34	7

correlation matrix is computed for horizontal and vertical lags ranging from minus seven to plus seven. It is thus a 15x15 matrix, although symmetry reduces the number of independent elements to 113. Correlation coefficients for larger lags would be based on too few pixel pairs for reliability.

TABLE 5-2. CORRELATION CLASSIFICATION ACCURACY

Feature Set	Global	Adaptive
COR (Rectilinear)	-	-
COR (Diagonal)	19.63	19.92
COR	19.63	19.92
COR+MKV	31.71	33.11
COR+LPL	54.83	47.80
COR+SBL	32.47	38.67
COR+MKV+LPL+SBL+IMG	63.62	65.23
MKV+LPL+SBL+IMG	63.62	65.23

The first row of Table 5-2 is based on rectilinear moments of the correlation matrix, as described in Equation 4-6. Discriminant functions could not be computed because none of these features have an F-ratio above 40. The second row uses diagonal moments as given in Equation 4-7. These are little better than the rectilinear moments, although CORD11 has sufficient power to generate a classification function. The third analysis combines both sets of moments; again only CORD11 is

useable. It is clear that moments of small-window correlation functions have little discriminating power on this texture set. They might perform better on directional textures or textures differing strongly in coarseness.

The next analysis combines autocorrelation features with first-order statistics of the whitened block. The plus sign represents the union of texture feature sets rather than addition. Each block was whitened with the Markov decorrelation operator of Equation 5-1. The operator is adaptive since it uses the nearest-neighbor correlation coefficients of each window in decorrelating that window. Two discriminant functions were found, with joint classification accuracy of almost 32%. The principal component is essentially **MKVAKR**. No autocorrelation feature is strong enough to contribute.

The next two analyses use nonadaptive 3x3 operations in place of the whitening filter. The Laplacian of Equation 5-2 works very well, identifying three texture dimensions related to **LPLSDV**, either **LPLKRT** or **LPLAKR**, and **LPLVAR**. The strong discriminating power of these features contradicts the theoretical basis of this section, which predicts superiority of the **MKV** features.

Faugeras and Pratt [63] proposed the Sobel gradient magnitude, an edge detector, as an ad hoc replacement for the decorrelation operation. As a texture detector, it works little better than the Markov whitening filter. For the globally equalized texture set, it identifies three

texture dimensions related to **SBLACV**, **SBLAVE**, and **SBLRNG**. For the adaptively equalized set it identifies four dimensions based on **SBLSDV**, **SBLASK**, **SBLVAR**, and **CORD11**.

The final two analyses made all of the preceding features available, with and without the correlation moments. The **IMG** features of the last section are also included: by themselves they have little discriminating power, but they could be important in combination with other features. Results of both analyses are identical since the correlation moments are not strong enough to enter into the model. The globally equalized textures produce six discriminant functions using **LPLSDV**, **LPLAKR**, **LPLVAR**, **SBLAVE**, and **SBLVAR**. The adaptively equalized textures generate seven functions using **LPLSDV**, **LPLAKR**, **LPLVAR**, **SBLSDV**, **IMGAKR**, **MKVSDV**, **SBLSKW**, and **SBLVAR**. In each case, the first three texture dimensions are much stronger than the rest. They are based almost entirely on standard deviations and variances of Laplacian and Sobel features. Scatter diagrams, pairwise F-ratio tables, and classification (or confusion) matrices show that texture dimensions computed for the two cases are similar. The least separated textures are Sand, Pigskin, and Leather. The chief texture dimensions seem to be Wool versus Raffia and Wood, and Water versus Raffia and Wool.

5.3 Summary

It is clear that the local autocorrelation function does not discriminate these eight textures, although it may measure texture dimensions not represented in this training set. This casts doubt upon the autocorrelation

texture model, and on the correlation-based linear predictive methods of texture segmentation [18]. The success of Laplacian and Sobel texture transforms will be explored further in Chapter 6.

CHAPTER 6

SPATIAL-STATISTICAL METHODS

Structural texture measures share a common weakness: discrete texture elements must be located, classified, and studied before texture itself can be measured. This is a severe computational problem even for simple artificial textures, and is nearly impossible for noisy, blurred, undulating, or stochastic textures. It would be difficult, for instance, to identify a reasonable texture primitive for the Pigskin image. Further, structural methods inherently classify a texture field as a whole, or at best classify discrete texture elements. They are unsuited to the task of segmenting an image by classifying each pixel.

We now introduce a more suitable class of texture features, called "spatial-statistical." The name is new, but many of the techniques are well known. Indeed, they would be claimed by researchers in both the statistical and structural camps.

The basic approach is to compute statistics of various local image functions. These measures are spatial because they depend upon local window functions rather than single pixels. They are statistical in the sense that statistical moments of an image window are invariant to relative pixel positions: pixels of the intermediate

functions could be shuffled without changing the composite texture measure.

To recapitulate: we compute functions of an image, e.g. by convolving with 3x3 masks, then compute the mean and other statistics in a window around each pixel. The number of texture features measured at each point is the number of image functions times the number of statistics.

Two window sizes are actually used. The "micro" window, used to compute spatial functions, is typically 3x3 or 5x5 pixels. The "macro" window for computing statistical moments is typically 15x15 pixels, possibly 31x31 or larger. Odd window sizes are convenient because they have well-defined center pixels.

The simplest micro-feature is the pixel value itself. One may regard this as the average luminance over a 1x1 region of the original image source. In calibrated imagery the pixel value has quantitative meaning, but pixels in typical images have only a relative meaning. This can invalidate some macro-statistics. One "cure" for this is to standardize each input image to a particular mean and contrast. The images used in this study have all been requantized to have uniform first-order statistics.

Two popular measures of texture coarseness are edge per unit area [32] and extrema per unit area [72]. Each is found by convolving a spatial operator with the image. The resulting feature plane may or may not be subjected to thresholding (hard limiting), thinning, or adaptive binarization. Then the response around each point is

integrated and assigned to that point as a texture measure. This last operation is equivalent to blurring of the feature plane.

A measure similar to a local standard deviation has been used by Hsu [73]. He computed the average deviation of neighborhood pixels from the neighborhood average and also from the intensity of the central pixel. These operations will locate image edges, but will also locate areas of high noise or high-frequency texture variations.

Recent evidence indicates that spot information is the only data transmitted to the brain by the optic nerve. The visual cortex then locates edge and line features from the spot response plane [74]. These edge features seem to be the principal determinants of perception [51], [14]. It is possible that a single type of primitive is sufficient to explain the myriad varieties of perceived texture, but it seems more sensible to use a larger set of texture primitives. One set, borrowed from terrain description, consists of peaks, pits, ravines, hillsides, passes and saddles [75], [76]. Measures similar to these will be investigated in this chapter.

Edge per unit area is generally considered a structural-statistical texture measure. Indeed it is, if the feature is computed by finding and counting discrete edge elements. The spatial-statistical paradigm includes this approach, but permits another: to compute the average (and other statistics) of an "edgeness" measure computed at each pixel. This saves having to determine a suitable

threshold level. It is not known which method is more powerful. Throughout this study the term spatial-statistical will refer to the second approach.

In a sense, the micro-windows themselves are used as primitive elements, but we shall reserve the terms texture primitive and texture element for structures inherent to the source texture. Properties related to these primitives, such as edge per unit area, can be measured without identifying the primitives themselves. The methods are thus purely statistical despite any theoretical dependence on structural elements.

Spatial-statistical methods are particularly appropriate for noisy or blurred imagery where texture elements cannot be identified with certainty. Very little work has been done on the identification of structural textures in the presence of noise, but effects of noise and blur on spatial-statistical features are relatively easy to model. A particularly tractable set of micro-features, spatial moments, will be discussed later in this chapter.

6.1 Window Size

This research uses micro-texture and macro-texture measures. Micro-texture measures are computed within very small overlapping windows. The windows are typically 3x3 or 5x5, small enough to make it unlikely that more than a single texture region exists within the window. Macro-texture measures are large-window summaries of the micro-features. Macro-windows must be large enough to include a

representative sample of the image texture. A method for dealing with windows overlaping more than one texture region has been suggested by Laws [77].

There is no theoretical reason for limiting the micro-window to 5x5; it could even be larger than the macro-window. The micro-window is typically small because:

- Micro-features are often very expensive to compute, taking time $O(n^2 \log n^2)$ or greater for a window of size nxn. The macro-statistics we propose are less costly and can be applied to larger windows. They can be computed in constant time regardless of the macro-window size.

- Micro-texture features are designed to measure local texture properties, while the macro-statistics measure properties of the texture field as a whole. The contrast between their sizes is essential for characterizing all but the simplest textures.

- There is no guarantee that any particular resolution or window size will be optimal for a given analysis. Still, there is a tendency for humans to request analyses requiring the finest resolution available from an image, and to obtain imagery with resolution just sufficient for the desired analysis. We may thus assume that very small windows can produce texture features as powerful as the highest resolution features used by the human retina.

- Small window features work very well. Rosenfeld and his co-workers [32], [78] achieved good results computing edge per unit area with the 2x2 Roberts gradient. This study will further support the power of such local operators.

It could be argued that micro-textures should be computed over several window sizes. This is not a great computational problem, but multiple window sizes quickly

create a large number of features. Five micro-features at five resolutions described by five macro-statistics would be 125 features to be computed, stored, and analyzed for each of perhaps 250,000 image pixels.

Further research may prove that many window sizes must be used simultaneously for proper texture identification. This approach has been used [79], [9] in edge detection and measurement of texture coarseness. It seems plausible, however, that a macro-scale characterization of micro-features is sufficient for preliminary texture classification, with syntactic, semantic, and special-purpose detectors invoked for detailed analysis of interesting regions.

The size of the optimal macro-window clearly depends on texture coarseness or regularity, as well as the quality of the available micro-features. It is to be hoped that one size will be found adequate within any given application. Multiple or adaptive window sizes could be implemented only at much greater expense.

6.2 Window Shape

When using Fourier descriptors, it is common practice to multiply window elements by a shaping function. This gives the most weight to center elements, progressively less to pixels near window edges. Such weighting functions have also found implicit use in the more sophisticated edge detection operations, as in the Hueckel operator [80], and even in simple operators such as the Sobel gradient function. The technique deserves

examination.

Weighted windows are used with transform methods because digital transforms are inherently cyclic. Each image block "wraps around" so that its left and right sides are adjacent, as are its top and bottom. One way of visualizing this is to imagine that the image block is surrounded by replicas of itself. Weighting functions which fall off toward the block edges reduce the sharp transitions, or aperture effects, that may occur there.

The other reason for using weighted windows is to reduce the effect of boundary overlap. A window covering more than one texture region will produce hybrid or even unpredictable texture measures. Window shaping reduces the effect of contrasting regions near the window edges.

For non-transform applications, the best weighting function depends on the average region size and shape relative to the window size. Exact criteria are in the realm of estimation theory. If it is known that the window covers a single texture, there is no reason to reduce the weight of any data. The most accurate classification will be possible if the largest computable window is used. Window shaping reduces the effective window size and hence the classification accuracy. It also adds to the computational burden, particularly since moving-window update techniques cannot be used. This study will not use weighted windows.

6.3 Statistical Moments

The first-order statistics of Section 3.5 may also be used as micro-features. We can, for instance, compute the standard deviations within moving 3x3 windows and then compute macro-window statistics within this feature plane. Resulting texture measures would be called **SDVAVE**, **SDVSDV**, etc. The name of a texture measure is composed of the micro-statistic name followed by the macro-statistic name.

This section compares the local statistical features with the **IMG**, Laplacian, and Sobel features discussed in Sections 3.7 and 5.2. The **AVE**, **SDV**, **SKW**, and **KRT** micro-features are simply small, continuously shifted versions of the corresponding macro-features. They are computed for each 3x3 or 5x5 window in the image, with the computed value assigned to the center pixel. Macro-features are then computed for 15x15 windows in the feature planes.

Individual features with F-ratios above 100 are listed in Table 6-1. The micro-window **AVE** features have little power. **SDV**, **SKW**, and **KRT** do better, about as well as **SBL** micro-features. None of these methods approaches the Laplacian in power, although jointly the 3x3 statistical features have about the same power as the **IMG**, **LPL**, and **SBL** sets together. The 5x5 measures perform less well, presumably because they contrast less with the 15x15 macro-statistics.

Joint classification accuracies are listed in Table 6-2. The largest feature set, using Laplacian, Sobel, and 3x3 statistical features together, performs far better

TABLE 6-1. LOCAL STATISTIC F-RATIOS

Feature	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
LPLVAR	707	-	609	-
LPLKRT	251	-	250	-
LPLSDV	851	-	700	-
LPLAKR	261	-	263	-
LPLMIN	429	-	374	-
LPLMAX	512	-	444	-
LPLRNG	571	-	488	-
SBLVAR	53	-	156	-
SBLACV	112	-	105	-
SDVVAR	50	73	166	193
SDVSKW	97	37	105	34
SDVSDV	52	74	166	195
SDVACV	158	134	179	136
SKWVAR	207	81	125	77
SKWSDV	245	99	164	94
SKWMAX	134	56	25	68
SKWRNG	244	151	49	130
KRTAVE	497	117	474	130
KRTVAR	181	54	178	55
KRTSKW	118	35	134	39
KRTSDV	234	67	228	69
KRTACV	150	57	144	52
KRTASK	118	35	134	39
KRTMIN	6	100	16	80
KRTMAX	184	83	157	90
KRTRNG	117	75	92	78
KRTMID	130	92	84	102

than any previous texture measures. Neither type of measure alone approaches this accuracy of 84%.

TABLE 6-2. LOCAL STATISTIC CLASSIFICATION ACCURACY

Micro-Feature Set	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
LPL	54.83	-	47.80	-
SBL	32.47	-	35.84	-
IMG+LPL+SBL	63.62	-	64.16	-
AVE	19.82	19.73	19.43	21.88
SDV	39.94	29.54	28.66	34.62
SKW	31.59	23.68	29.20	21.92
KRT	39.50	33.89	37.26	33.45
AVE+SDV+SKW+KRT	59.81	48.44	61.62	46.88
IMG+LPL+SBL+AVE +SDV+SKW+KRT	84.57	65.63	82.52	67.82

It is apparent from the scatter diagrams (not shown) that the two combined 3x3 feature sets, **IMG+LPL+SBL** and **AVE+SDV+SKW+KRT**, are measuring slightly different texture dimensions. This is confirmed by the much greater classification accuracy when both sets are combined. Principal components of the globally equalized textures are based on **LPLSDV**, **KRTAVE**, **LPLAKR**, (**SBLACV**), **LPLVAR**, **SDVAVE**, **SBLAVE**, **SDVSDV**, (**IMGASK**), **IMGMAX**, **SKWAVE**, **IMGAVE**, **AVEACV**, **SKWVAR**, and **SBLVAR**. Terms in parentheses were dropped from the model as other terms were found to be jointly more powerful. The adaptively equalized textures generate principal components using **LPLSDV**, **KRTAVE**,

LPLAKR, **SBLAVE**, **SDVAVE**, **LPLVAR**, **SDVSDV**, (**IMGRNG**), **SBLSDV**, **SKWAVE**, **IMGMAX**, and **IMGVAR**.

Surprisingly, the joint classification accuracy is lower when the 5x5 statistical moments are combined with the 3x3 Laplacian and Sobel. Principal components for both texture sets require **LPLSDV**, **LPLAKR**, and **LPLVAR**. The globally equalized set adds **KRTAVE**, **SBLACV**, and **SKWRNG**; the adaptive set requires **SKWSDV**, **SBLACV**, **SDVAVE**, and **IMGRNG**. The 5x5 statistical moments add almost nothing to the information in the 3x3 Laplacian and Sobel.

It is difficult to draw conclusions from the data presented here. A set of simple 3x3 texture measures evaluated over 15x15 blocks has been found to have extraordinary discriminating power. The first two texture dimensions are slightly rotated versions of those found with co-occurrence methods. The least separated textures are still Grass, Sand, and Pigskin. The first principal component separates Wood from Wool, the second separates Raffia from the other seven textures. The number of terms, however, makes it difficult to say just what is being measured. We shall continue our search for a set of fast, effective texture measures.

6.4 Spatial Moment Masks

Since texture is a locally spatial phenomenon, we must use local spatial operators to generate our feature planes. Computation of spatial moments is equivalent to multiplying an image window by a mask and then summing. This is exactly what is done in convolution. It seems

reasonable to convolve small spatial moment masks with an image to produce a set of feature planes.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

M00 **M01** **M02**

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

M10 **M11** **M12**

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

M20 **M21** **M22**

Figure 6-1. 3x3 Spatial Moment Masks

The spatial moments of a local window are

$$Mij = (1/n^2) \sum_{r,c} r^i \, c^j \, I(r,c) \qquad (6\text{-}1)$$

It is assumed that row and column indices are relative to the window center, and that the computed moments are assigned to this center point as a feature vector. The

3x3 and 5x5 spatial moment masks are shown in Figures 6-1 and 6-2.

When spatial moments are computed over a probability density, such as a co-occurrence matrix, it is often desirable to relate higher moments to the center of the probability mass, **(M10/M00,M01/M00)**. For instance,

$$M20' = (1/n^2) \sum_{r,c} (r - M10/M00)^2 \, I(r,c)$$

or

$$M20' = M20 - M10^2 / M00$$

The same normalization is often used in character recognition systems to achieve shift invariance. For small texture windows, however, such standardization makes little difference. It is not worth the extra computation, and may not even be appropriate.

Table 6-3 lists local moment features with F-ratios above 150. **M10SDV**, **M11SDV**, and **M12SDV** features are seen to be extremely powerful. Several **RNG** features are also outstanding, but will be found less important in conjunction with the other texture measures. **M00**, **M02**, **M20**, and **M22** features are seen to have very little power. Note that the **M00** moment is identical to the **AVE** micro-feature.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 4 & 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 & 4 \end{bmatrix}$$

M00 **M01** **M02**

$$\begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \quad \begin{bmatrix} 4 & 2 & 0 & -2 & -4 \\ 2 & 1 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -2 & 0 & 2 & 4 \end{bmatrix} \quad \begin{bmatrix} -4 & -2 & 0 & -2 & -4 \\ -2 & -1 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 2 \\ 4 & 2 & 0 & 2 & 4 \end{bmatrix}$$

M10 **M11** **M12**

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \quad \begin{bmatrix} -4 & -2 & 0 & 2 & 4 \\ -2 & -1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -2 & 0 & 2 & 4 \end{bmatrix} \quad \begin{bmatrix} 4 & 2 & 0 & 2 & 4 \\ 2 & 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 2 \\ 4 & 2 & 0 & 2 & 4 \end{bmatrix}$$

M20 **M21** **M22**

Figure 6-2. 5x5 Spatial Moment Masks

TABLE 6-3. LOCAL MOMENT F-RATIOS

Feature	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
M01VAR	258	177	587	427
M01SKW	221	52	229	54
M01KRT	183	21	249	38
M01SDV	274	183	618	424
M01ASK	198	24	208	26
M01MAX	153	122	248	203
M01RNG	136	148	249	292
M10VAR	797	601	909	797
M10SDV	1490	1038	1486	1195
M10MIN	944	698	956	725
M10MAX	765	594	722	579
M10RNG	1388	1062	1407	1144
M11VAR	837	609	713	603
M11KRT	185	61	196	69
M11SDV	1245	892	977	804
M11AKR	183	22	194	25
M11MIN	543	491	434	408
M11MAX	506	429	418	363
M11RNG	773	678	628	581
M12VAR	769	584	865	746
M12SDV	1428	1003	1396	1120
M12MIN	883	633	885	634
M12MAX	704	541	644	506
M12RNG	1270	952	1259	982
M21VAR	266	177	621	431
M21SKW	206	49	212	51
M21KRT	167	27	229	47
M21SDV	284	182	652	429
M21ASK	180	24	188	25
M21MAX	154	103	249	178
M21RNG	141	119	263	244

TABLE 6-4. MOMENT CLASSIFICATION ACCURACY

Feature Set	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
Mij	81.05	65.67	77.00	67.72

Table 6-4 shows classification accuracies on each of the texture sets. The first analysis, with 81% classification accuracy, uses **M10SDV, M11SDV, M10VAR, M12VAR, M01SDV, M21SDV, M01SKW,** and **M11KRT.** Scatter diagrams for the first two texture dimensions are visually different from those of previous texture sets, but the pattern of group centroids is much the same. The first dimension separates Wood and Water from the rest; the second separates Raffia from Wool and Leather. The 3x3 adaptive case gives very similar results with **M10SDV, M11SDV, M12SDV, M10VAR, M01SDV, M21SDV, M01SKW,** and **M11KRT.** The dominance of **SDV** and **VAR** macro-statistics is obvious. micro-window moments containing odd powers are also dominant; they are the ones with zero-sum masks.

Model features for 5x5 moments are similar to those for 3x3 moments. A large decrease in classification accuracy occurs with the larger micro-features. This trend has been noted before. It may be an artifact of the texture set, or an interaction of micro-window and macro-window sizes. It may also indicate that the perimeter-weighted moments are not as appropriate as center-weighted statistics such as the Laplacian. The larger micro-window brings out the perimeter weighting of the spatial moments.

6.5 Rotation-Invariant Moments

Most investigators have chosen texture measures that are invariant to rotation of the texture field. This is partly because perceived texture, particularly perceived coarseness, is little changed by rotation. The assumption of rotational isotropy has also been used to reduced the

number of measured texture features and to increase statistical reliability of texture features by averaging measurements in different directions.

There is a need for directional texture features. Humans are able to distinguish horizontal line textures from vertical ones, and left gradients from right gradients. One application of directional texture measures is the segmentation and interpretation of rock strata in seismic images. There is also a need for nondirectional texture measures, such as the Laplacian. This section describes two methods of generating nondirectional features from the directional spatial moments of the previous section.

Assume that the image texture has a dominant direction, such as a global gradient or a major Fourier component. Let the camera or texture field be rotated through an angle A, and let a = cos(A), b = sin(A). The new moments can be computed from the original window as

$$Mij(A) = (1/n^2) \sum_{r,c} (ar + bc)^i \, (ac - br)^j \, I(r,c)$$

Haralick computes several features of this form to measure energy along co-occurrence matrix diagonals. Using the binomial expansion it can be seen that these moments are linear combinations of the Mij. For instance,

$$M11(A) = -abM20 + (a^2 - b^2)M11 + abM02$$

A better method of normalization has been developed by Hu [81]. He derives the following orthogonal set of rotation-invariant moments:

$$RI1 = M20+M02$$

$$RI2 = (M20-M02)^2+4M11^2$$

$$RI3 = (M30-3M12)^2+(3M21-M03)^2$$

$$RI4 = (M30+M12)^2+(M21+M03)^2$$

$$RI5 = (M30-3M12)(M30+M12)[(M30+M12)^2-3(M21+M03)^2]$$
$$+(3M21-M03)(M21+M03)[3(M30+M12)^2-(M21+M03)^2]$$

$$RI6 = (M20-M02)[(M30+M12)^2-(M21+M03)^2]$$
$$+4M11(M30+M12)(M21+M03)$$

$$RI7 = (3M21-M03)(M30+M12)[(M30+M12)^2-3(M21+M03)^2]$$
$$-(M30-3M12)(M21+M03)[3(M30+M12)^2-(M21+M03)^2]$$

Maitra [82] suggests a set of ratios of these functions which are invariant to contrast and scale changes as well as rotation. We will call them "full invariants," although they are not invariant to changes in luminance level. In theory, they are also invariant to scale changes, but this may not hold when the sampling rate and window size remain constant. The moments, modified to avoid negative roots, are:

$$FI1 = \sqrt{|RI2|} \;/\; RI1$$

$$FI2 = (RI3 * M00) \;/\; (RI2 * RI1)$$

$$FI3 = RI4 \;/\; RI3$$

$$FI4 = \sqrt{|RI5|} \;/\; RI3$$

$$FI5 = RI6 \;/\; (RI4 * RI1)$$

$$FI6 = RI7 \;/\; RI5$$

Both sets of invariant moments present computational difficulties. Rotation-invariants RI5 and RI7 tend to

"blow up" because of the high powers involved. We have corrected for this by scaling the Mij terms by 1/255, in effect scaling the input data to the range zero to one. Full-invariants give trouble because the denominators can approach zero. We have set the quotient to zero if the magnitude of the denominator is less than 0.001.

Note that these invariant moments, like the spatial moments of the last section, are used only as micro-features. They are computed on 3x3 or 5x5 windows, not on the larger macro-windows. Application of the twelve macro-statistics generates 84 rotation-invariant texture features and 72 full-invariant features.

The invariants are nonlinear transformations of the moment feature planes. They are rotation invariant in the same sense as the statistical moments of the last section: the output of each micro-window is theoretically unaffected by rotation of the texture field around the center of that micro-window. In practice, this is only approximately true because of discretization and aperture effects. Global effects of rotation are removed by the macro-statistic computation, which is invariant to the rotation or translation of the micro-windows.

Tables 6-5 and 6-6 show the individual powers of the rotation-invariant moments and full-invariant moment ratios. The tables show that RI2, RI5, FI3, and FI4 micro-features are the most useful for texture description.

Table 6-7 shows that discriminating power always

TABLE 6-5. ROTATION-INVARIANT MOMENT F-RATIOS

Feature	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
RI2VAR	837	603	713	596
RI2KRT	187	59	198	66
RI2SDV	1244	875	976	789
RI2AKR	185	21	197	23
RI2MIN	547	501	437	419
RI2MAX	502	412	413	344
RI2RNG	773	673	627	575
RI3VAR	42	33	134	70
RI3SDV	30	29	124	65
RI3ACV	140	147	135	138
RI4AVE	62	64	91	132
RI4VAR	66	62	200	208
RI4SDV	65	63	204	225
RI4ACV	120	149	120	157
RI5AVE	193	153	481	368
RI5VAR	69	60	173	145
RI5SDV	74	70	199	203
RI5ACV	118	6	122	7
RI5MAX	376	278	283	244
RI5MID	85	99	144	172
RI6MAX	114	101	80	75
RI6RNG	128	103	99	94
RI7SDV	82	86	41	107

TABLE 6-6. FULL-INVARIANT MOMENT F-RATIOS

Feature	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
FI1AVE	53	63	379	772
FI2SDV	69	24	114	23
FI3AVE	247	46	177	42
FI3VAR	234	9	250	11
FI3SKW	197	12	147	8
FI3KRT	159	0	89	1
FI3SDV	386	29	381	27
FI3ACV	319	36	378	32
FI3ASK	180	12	137	8
FI3AKR	149	0	86	1
FI3MAX	159	21	153	19
FI3RNG	159	21	153	19
FI3MID	159	21	153	19
FI4VAR	245	18	264	19
FI4SKW	258	19	220	12
FI4KRT	272	6	176	5
FI4SDV	314	41	331	36
FI4ACV	202	56	306	48
FI4ASK	187	19	168	12
FI5AKR	184	6	140	5
FI4MAX	147	24	133	20
FI4RNG	147	24	133	20
FI4MID	147	24	133	20
FI5VAR	173	65	196	184
FI5SDV	227	81	240	204
FI5MIN	175	88	99	58
FI5MAX	186	102	111	113
FI5RNG	265	124	140	121
FI6SDV	151	8	106	5
FI6RNG	106	8	67	5

decreases as more invariance is added. The 3x3 rotation-invariant features still perform very well, better than co-occurrence measures. Adaptive equalization has little effect on the classification accuracies; surprisingly, it has less effect on rotation-invariants than on full-invariants. Globally equalized textures use **RI2SDV, RI2VAR, RI4AVE, RI3AVE, RI2AKR, RI5AVE, RI6ACV,** and **RI6AVE. AVE** macro-features are apparently of use because of the nonlinear product terms involved in computing these moments. Discriminant functions for the adaptively equalized textures use **RI2SDV, RI5AVE, RI4SDV, RI3SDV, RI2KRT, RI2VAR, RI4AVE, RI6SDV, RI6AVE,** and **RI1VAR.**

TABLE 6-7. INVARIANT CLASSIFICATION ACCURACY

Feature Set	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
RI	74.17	54.25	74.17	57.37
FI	53.27	30.47	56.69	37.26

Full-invariants are nearly invariant to texture as well as to rotation and contrast. It must be concluded that contrast invariance is better achieved by global or macro-window equalization than by micro-window equalization. Rotation invariance, when required by a particular application, can be obtained at little cost with the **RI** features or the local statistical measures of

the last section.

6.6 Joint Moments

Nonlinear functions can be introduced by squaring or otherwise transforming window elements prior to computing moments. Let

$$Mijk = (1/n^2) \sum_{r,c} r^i \, c^j \, I^k(r,c) \qquad (6\text{-}2)$$

This reduces to the spatial moments when k = 1 and to the statistical moments when i = j = 0. It is possible that the joint moments are more powerful descriptors than the spatial and statistical features together.

Preliminary trials proved that the texture features of Equation 6-2 are of no use for $k \neq 1$. This prompted the correction of higher moments for the k = 1 and k = 2 moments. The correction formulas are exactly analogous to Equations 3-1 through 3-4. This section will investigate the 432 features generated by the twelve macro-statistics applied to the corrected Mijk for i and j ranging from zero to two and k ranging from one to four.

Table 6-8 shows that only the **AVE**, **VAR**, and **SDV** macro-features are very strong, and then only for 3x3 micro-features. The only class of micro-features worth computing is the Mij1 set, which is identical to the Mij set of Section 6.4. It is surprising that better classification accuracies are not achieved, considering the enormous computational resources thrown at the problem.

TABLE 6-8. JOINT MOMENT CLASSIFICATION ACCURACY

Feature Set	3x3 Global	5x5 Global	3x3 Adaptive	5x5 Adaptive
M00k	56.05	48.44	62.30	46.88
M01k	54.30	36.43	48.19	38.43
M02k	56.79	19.63	53.27	-
M10k	48.54	44.19	47.41	45.41
M11k	45.85	39.01	41.65	36.23
M12k	48.10	43.31	46.39	45.17
M20k	36.77	24.71	45.17	20.75
M21k	52.20	36.91	49.02	36.91
M22k	48.10	21.53	41.31	-
Mij1	81.05	65.67	77.00	67.72
Mij2	66.75	57.52	70.26	58.64
Mij3	62.50	45.75	58.20	48.05
Mij4	62.40	57.86	63.33	55.18
MijkAVE	83.01	65.14	69.63	56.88
MijkVAR	78.03	64.26	80.37	68.99
MijkSKW	54.98	42.04	57.86	40.92
MijkKRT	53.56	41.85	54.69	38.96
MijkSDV	80.96	67.14	83.54	69.48
MijkACV	54.20	42.77	61.52	43.41
MijkASK	42.92	33.25	58.11	32.47
MijkAKR	37.89	38.38	38.57	39.06
MijkMIN	62.06	57.23	54.74	56.64
MijkMAX	52.54	50.54	53.27	49.27
MijkRNG	60.79	60.06	59.42	59.91
MijkMID	60.01	47.75	55.57	41.02

6.7 Combined Moments

This section combines the IMG, LPL, and SBL micro-features with the 3x3 and 5x5 AVE, SDV, SKW, KRT, and Mij micro-features. Twelve macro-statistics are computed for each of the 29 micro-feature planes, generating 348 texture measures.

The first section of Table 6-9 shows that 3x3 moments

TABLE 6-9. COMBINED MOMENT CLASSIFICATION ACCURACY

Feature Set	Adaptive
3x3	88.67
5x5	73.73
3x3+5x5	85.25
3x3 VAR+SDV	84.08
3x3 VAR	82.37
3x3 SDV	86.04

contain more texture information than 5x5 moments. In fact, when both are used none of the 5x5 measures enter the discriminant functions. They contain no information which is not more easily extracted from 3x3 measures. This does not mean that a particular 5x5 feature measures exactly the same thing as the corresponding 3x3 feature, but that the set of 5x5 features contains the same texture information as the set of 3x3 features.

The second section shows that standard deviation macro-statistics of the 3x3 moment planes contain nearly as much information as all twelve macro-statistics. Variables required for 86% classification accuracy are **M10SDV, LPLSDV, M11SDV, M01SDV, M12SDV, M20SDV, KRTSDV, SDVSDV,** and **M02SDV.** If some of these variables were unavailable it is quite likely that others among the 348 could be found to provide the same information. A scatter plot of the eight texture classes against the first two principal axes looks very similar to those produced with co-occurrence and other texture features.

6.8 Ad Hoc Masks

Many researchers have suggested texture measures based on edge per unit area or average Laplacian. Our experimental results, documented in the next chapter, show that these are wise choices. Standard deviations of 3x3 spot and edge measures are very powerful features. Averages computed within thresholded feature planes would be very similar.

The quality of these measures suggests further experimentation. The following convolution masks have been chosen as spot and ring detectors:

$$\mathbf{SPT1} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad \mathbf{SPT2} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\mathbf{SPT3} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad \mathbf{SPT4} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

$$\mathbf{SPT5} = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \qquad \mathbf{SPT6} = \begin{bmatrix} -1 & 1 & -1 \\ 1 & 0 & 1 \\ -1 & 1 & -1 \end{bmatrix}$$

Note that the **SPT1** mask is the Laplacian of previous sections. The coefficients of these masks sum to zero, making computed texture measures invariant to luminance shifts. Otherwise there was no particular theory behind the selection of these masks.

Elongated spots can appear as thin lines. These may be detected with the following masks:

$$\mathbf{LNE1} = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \qquad \mathbf{LNE2} = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\mathbf{LNE3} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{LNE4} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

LNE4 texture measures are the same as the **M11** measures suggested in Section 6.4.

Large spots, lines, and regions may be sensed by edge detectors. We shall use

$$\mathbf{EDG1} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \mathbf{EDG2} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{EDG3} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \mathbf{EDG4} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The first two masks are identical to the **M01** and **M10** spatial moment masks.

There is anatomical evidence that the eye contains separate detectors for bright spots and for dark spots. There may also be neurons which respond similarly to both positive and negative spots. We can test such texture

features by measuring response magnitude. Using magnitudes is also a way of introducing nonlinearities in the discriminant functions. Absolute values of the micro-features will be denoted by

$$\mathbf{ASPi} = |\mathbf{SPTi}|$$

$$\mathbf{ALNi} = |\mathbf{LNEi}|$$

$$\mathbf{AEDi} = |\mathbf{EDGi}|$$

The notation is meant to indicate absolute response to a mask rather than response to an absolute mask. Micro-feature **ALN4** has not been computed.

Edge detectors in common use respond equally to edges in different directions. Rotation-invariant micro-features used for this study will be

$$\mathbf{ILN1} = \sqrt{(\mathbf{LNE1})^2 + (\mathbf{LNE2})^2}$$

$$\mathbf{ILN2} = \mathbf{ALN3}$$

$$\mathbf{IED1} = \sqrt{(\mathbf{EDG1})^2 + (\mathbf{EDG2})^2}$$

$$\mathbf{IED2} = \sqrt{(\mathbf{EDG3})^2 + (\mathbf{EDG4})^2}$$

Again, the notation represents feature plane operations rather than operations on the convolution masks. **IED1** and **IED2** are commonly known as the Prewitt and Sobel edge detectors.

It turns out that these local moments provide exceptionally strong texture measures. When run with the combined moments of the previous section, these features are the only ones entering the discriminant functions. (Of course, some of these features are duplicates of the

LPL, SBL, M01, M10, and M11 features.) The statistical moments of Section 6.3 were not made available, but they have been shown less powerful than the spatial moment features.

TABLE 6-10. AD HOC MOMENT CLASSIFICATION ACCURACY

Feature Set	3x3 Global	3x3 Adaptive
SPT	76.81	74.07
LNE	75.68	67.82
EDG	68.46	64.60
ASP	74.51	72.17
ALN	71.68	68.85
AED	69.58	67.53
ILN	47.61	54.64
IED	56.15	55.18
SPT+ASP	75.83	72.46
LNE+ALN+ILN	73.78	74.61
EDG+AED+IED	66.50	69.29
AVE+VAR+SDV+ACV	-	86.52
AVE+VAR+SDV	87.16	87.16
AVE+VAR	-	86.28
AVE+SDV	-	87.50
VAR+SDV	88.92	84.77
VAR	80.42	82.14
SDV	87.45	85.79

Table 6-10 shows the classification results with various subsets of the ad hoc texture measures. None of the single-type subsets perform well. Even the combined subsets, such as **SPT+ASP**, do not perform well. Other

experiments (not shown) indicate that several Spot and Line features are needed. Edge features are also useful. Absolute features are important, but rotation-invariant line and edge features are of little use.

The final section of Table 6-10 is based on the combined set of 30 micro-features, but with various subsets of the macro-statistics. The first line, **AVE+VAR+SDV+ACV**, is essentially equivalent to the entire set of macro-statistics. The following lines show that very little discriminant power is lost by using only the **SDV** statistics. The differences between pairs of very similar features, such as **(EDG1SDV - EDG3SDV)**, are of great importance, apparently because the difference forms a feature nearly orthogonal to the originals. Tables 6-11 through 6-13 show the discriminating powers of individual features. Lines with no F-ratios above 200 have been omitted. Interestingly, none of the **SPT3**, **SPT4**, **SPT5**, or **SPT6** features were of this strength, nor were the absolute versions of the same features. Only the **SPT3** features even came close. It is difficult see why this should be so.

Also missing are the rotation-invariant Line features and most of the rotation-invariant Edge features. Only the Prewitt operator, **IED1**, has a ratio above 200. Evidently, edge per unit area texture measures should be based on directional gradients rather than gradient magnitude.

The difference in strength between **LNE3** and **LNE4**

TABLE 6-11. AD HOC SPOT F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
SPT1VAR	707	609	**SPT2VAR**	280	268
SPT1KRT	251	250	**SPT2KRT**	144	178
SPT1SDV	851	700	**SPT2SDV**	293	274
SPT1AKR	261	263	**SPT2AKR**	41	47
SPT1MIN	429	374	**SPT2MIN**	192	183
SPT1MAX	512	444	**SPT2MAX**	316	285
SPT1RNG	571	488	**SPT2RNG**	376	336
ASP1AVE	849	690	**ASP2AVE**	252	233
ASP1VAR	665	607	**ASP2VAR**	364	382
ASP1SKW	216	225	**ASP2SKW**	128	156
ASP1SDV	784	671	**ASP2SDV**	359	355
ASP1ACV	318	354	**ASP2ACV**	172	219
ASP1ASK	216	225	**ASP2ASK**	128	156
ASP1MAX	519	449	**ASP2MAX**	327	291
ASP1RNG	518	499	**ASP2RNG**	327	291
ASP1MID	519	450	**ASP2MID**	327	291

TABLE 6-12. AD HOC LINE F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
LNE1VAR	182	292	**LNE2VAR**	581	499
LNE1KRT	387	455	**LNE2KRT**	31	30
LNE1SDV	231	364	**LNE2SDV**	1068	810
LNE1AKR	271	299	**LNE2AKR**	25	25
LNE1MIN	52	82	**LNE2MIN**	599	515
LNE1MAX	122	163	**LNE2MAX**	501	438
LNE1RNG	94	144	**LNE2RNG**	766	650
LNE3VAR	40	96	**LNE4VAR**	837	713
LNE3SDV	44	96	**LNE4SDV**	1245	977
LNE3MIN	58	68	**LNE4MIN**	543	434
LNE3MAX	49	51	**LNE4MAX**	506	418
LNE3RNG	66	74	**LNE4RNG**	773	628
ALN1AVE	285	432	**ALN2AVE**	1009	730
ALN1VAR	132	222	**ALN2VAR**	502	475
ALN1SKW	285	328	**ALN2SKW**	20	17
ALN1KRT	201	219	**ALN2KRT**	10	7
ALN1SDV	147	240	**ALN2SDV**	980	819
ALN1ACV	573	681	**ALN2ACV**	82	91
ALN1ASK	285	328	**ALN2ASK**	20	17
ALN1MAX	77	117	**ALN2MAX**	616	530
ALN1RNG	76	116	**ALN2RNG**	615	529
ALN1MID	77	117	**ALN2MID**	617	530

TABLE 6-13. AD HOC EDGE F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
EDG1VAR	258	587	EDG2VAR	797	909
EDG1SKW	221	229	EDG2SKW	62	62
EDG1KRT	183	249	EDG2KRT	16	24
EDG1SDV	274	618	EDG2SDV	1490	1486
EDG1ASK	198	208	EDG2ASK	23	23
EDG1MIN	100	125	EDG2MIN	944	956
EDG1MAX	153	248	EDG2MAX	765	722
EDG1RNG	136	249	EDG2RNG	1388	1407
EDG3VAR	243	545	EDG4VAR	810	917
EDG3SKW	237	247	EDG4SKW	61	61
EDG3KRT	202	276	EDG4KRT	31	46
EDG3SDV	256	573	EDG4SDV	1510	1492
EDG3ASK	217	230	EDG4ASK	26	26
EDG3MIN	91	108	EDG4MIN	945	967
EDG3MAX	147	242	EDG4MAX	776	730
EDG3RNG	123	222	EDG4RNG	1396	1415
AED1AVE	289	586	AED2AVE	1343	1245
AED1VAR	200	534	AED2VAR	788	1058
AED1SKW	168	226	AED2SKW	21	29
AED1SDV	202	527	AED2SDV	1430	1664
AED1ASK	167	226	AED2ASK	21	29
AED1MAX	92	159	AED2MAX	1206	1252
AED1RNG	91	158	AED2RNG	1202	1250
AED1MID	92	160	AED2MID	1209	1254
AED3AVE	276	551	AED4AVE	1356	1232
AED3VAR	180	480	AED4VAR	804	1101
AED3SKW	182	246	AED4SKW	36	50
AED3SDV	180	475	AED4SDV	1452	1714
AED3ASK	182	247	AED4ASK	36	50
AED3MAX	82	140	AED4MAX	1205	1254
AED3RNG	81	138	AED4RNG	1202	1252
AED3MID	83	141	AED4MID	1208	1255
IED1VAR	63	214	IED2VAR	53	156
IED1SDV	64	218	IED2SDV	55	159

features should be noted. The two micro-operators are similar, being essentially rotated versions of each other. For some reason the diagonal line detector is much more powerful. This could be due to anisotropy of the data set, but results to be presented in the next chapter show much stronger discrimination for vertical and horizontal features than for any diagonal feature. The only other explanation which presents itself is the separable nature of the **LNE4** mask. All of the masks which work well can easily be expressed as the product (or convolution) of a vertical vector and a horizontal vector. None of the masks which work poorly have this property.

Separability into vertical and horizontal features might well be of importance in biological vision systems. Octopi and rats have great difficulty discriminating diagonals in different directions. Rabbits, cats, and humans are known to discriminate stimuli near the vertical and horizontal more accurately than those which are nearly diagonal. The apparent diagonal structure of the **LNE4** mask could thus be less important than its horizontal and vertical decomposition. It is difficult to see, however, how this separable structure could be important in a mathematical discriminant analysis.

6.9 Summary

This chapter presented many sets of texture measures, all fitting the spatial-statistical paradigm. Local statistical moments were found useful only when combined with spatial moments such as the Laplacian. Spatial moments alone are also lacking, although more powerful

than co-occurrence texture measures. Rotation-invariant moments are somewhat weaker, but possibly useful. Full-invariants and joint spatial moments are nearly invariant to texture differences. Some of the ad hoc 3x3 operators work well, others do not.

A few other lessons have been learned:

- Texture can be measured with very local operators.
- The 5x5 spatial moments are jointly less powerful than the 3x3 moments, and contain no additional texture information; this may be an inherent fault of perimeter-weighted masks.
- Convolution masks which are zero-sum and separable seem to work best.
- Statistics of rotation-invariant measures work less well than linear combinations of directional statistics.
- The only macro-statistic needed is the standard deviation.

We shall use these lessons in the next chapter to develop even better texture analysis methods.

CHAPTER 7

TEXTURE ENERGY MEASURES

This chapter develops our final spatial-statistical texture model, one incorporating the best of our previous models. We shall measure texture in much the same way as in the previous chapter, convolving small center-weighted filter masks across the image and then computing statistics within a window around each pixel. The responses to several such transforms will then be combined in discriminant and classification functions for a set of known textures.

7.1 Center-Weighted Filter Masks

Figure 7-1 shows three sets of one-dimensional convolution masks. We suggest that these be called the Lattice Aperture Waveform Sets of orders three, five, and seven. The names of the vectors are mnemonics for Level, Edge, Spot, Wave, Ripple, Undulation, and Oscillation. Vectors in each set are ordered by sequency[4]. The vectors are weighted toward the center, all are symmetric or antisymmetric, and all but the Level vectors are zero-sum. The vectors in each set are independent, but not orthogonal.

---

[4]Number of zero crossings: zero for L7, six for O7.

$$L3 = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$E3 = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$S3 = \begin{bmatrix} -1 & 2 & -1 \end{bmatrix}$$

$$L5 = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$E5 = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$

$$S5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

$$W5 = \begin{bmatrix} -1 & 2 & 0 & -2 & 1 \end{bmatrix}$$

$$R5 = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

$$L7 = \begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix}$$

$$E7 = \begin{bmatrix} -1 & -4 & -5 & 0 & 5 & 4 & 1 \end{bmatrix}$$

$$S7 = \begin{bmatrix} -1 & -2 & 1 & 4 & 1 & -2 & -1 \end{bmatrix}$$

$$W7 = \begin{bmatrix} -1 & 0 & 3 & 0 & -3 & 0 & 1 \end{bmatrix}$$

$$R7 = \begin{bmatrix} 1 & -2 & -1 & 4 & -1 & -2 & 1 \end{bmatrix}$$

$$O7 = \begin{bmatrix} -1 & 6 & -15 & 20 & -15 & 6 & -1 \end{bmatrix}$$

Figure 7-1. Center-Weighted Vector Masks

The 1x3 vectors form a basis for the larger vector sets[5]. Each 1x5 vector may be generated by convolving two 1x3 vectors. S5, for instance, can be generated as (L3)*(S3), (S3)*(L3), or (E3)*(E3). The 1x7 vectors can be generated by convolving 1x3 and 1x5 vectors, or by twice convolving 1x3 vectors. The sequency of a generated vector is the sum of the component sequencies.

Figure 7-2 shows the nine masks generated by convolving a vertical 3-vector with a horizontal 3-vector. This may be considered a cross-product or vector multiplication operation, but convolution has special significance here. We shall extract texture information from image data by convolving with the 3x3 masks, just as we did with spatial moment and ad hoc masks. Convolution with the component one-dimensional masks gives exactly the same result as convolution with a separable 3x3 mask.

The nine independent 3x3 masks form a complete set. Any 3x3 matrix can be expressed as a unique linear combination of the masks. This was also true of the perimeter-weighted spatial moment masks, but the center-weighted set contains the edge, line, and spot masks which were shown in Section 6.8 to be more powerful. Eight of the center-weighted masks are zero-sum, a property shown in Section 6.4 to be important.

The 5x5 masks and 7x7 masks (not shown) are similar,

---

[5]The 1x3 vector elements can be derived from coefficients of the polynomials (a+b)(a+b), (a+b)(a-b), and (a-b)(a-b). Indeed, any of the vector sets may be generated from coefficients of the binomial expansion.

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

L3L3

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

L3E3

$$\begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}$$

L3S3

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

E3L3

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

E3E3

$$\begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$

E3S3

$$\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix}$$

S3L3

$$\begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & 2 \\ 1 & 0 & -1 \end{bmatrix}$$

S3E3

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

S3S3

Figure 7-2. 3x3 Center-Weighted Masks

with even stronger weighting toward the center. The separable structure of these masks makes it feasible to apply them as spatial-domain filters. A 5x5 convolution, for instance, can be implemented as two 3x3 convolutions, a 5x1 and a 1x5 convolution, or two 3x1 and two 1x3 convolutions.

We have also investigated the discriminating power of one-dimensional masks. Previous experiments have shown

that rotation-invariant filters, such as the Sobel gradient magnitude, are only fair as texture measures. Better results are obtained by using directional masks separately and then combining the texture measures. We have applied horizontal and vertical masks in pairs, although the discriminant analyses have not been constrained to assign equal weights. Sets of 3x5, 3x7, and similar rectangular masks have not been tried.

7.2 Macro-Statistic Selection

It is time to re-examine our set of macro-window texture statistics. In the last chapter we used twelve measures. Experience has shown that either the variance or standard deviation alone is sufficient to extract texture information from the filtered images.

Variance is an average squared deviation from the mean. For a zero mean field, as produced by convolution with a zero-sum mask, variance is the average of squared signal values. It is thus an energy measure, in the formal sense of the word. It measures the total energy within a window. If the image has been filtered, it measures local energy within the pass band. The **SDV** macro-statistic is the square root of this local energy. It may be considered a "texture energy" measure.

These statistics are more local than previously studied frequency-domain texture measures. Frequency components are measured with very small convolution masks. Each micro-window is treated independently, without regard to its phase relationships with other micro-windows. This

is appropriate for textures with short coherence length or correlation distance. It is less powerful than Fourier methods for man-made textures with inherent synchronization of texture element spacings.

Energy and variance are both defined as sums of squares because such sums are analytically tractable. The physical world is under no constraint to be tractable. It is probable that the human visual system avoids root-mean-square computations, and quite possible that simpler statistics are more appropriate for texture analysis.

Tables 7-1 and 7-2 present three alternatives to the standard deviation. The first, **ABSAVE**, is computed as the average absolute value within a macro-window. For a zero mean field, it may be considered a fast approximation to the standard deviation. The table of F-ratios shows that it performs poorly only with **L3L3**, the 3x3 operator which is not zero-sum. The table of classification accuracies, which was computed for the adaptively equalized texture set using fifty 3-vector, 5-vector, 3x3, and 5x5 feature sets, shows that **ABSAVE** features are jointly more powerful than **SDV** features, and nearly as powerful as both sets together.

The **SDV** and **ABSAVE** macro-statistics share a common weakness. Neither can distinguish between a dark field with bright spots and a bright field with dark spots. In statistical terms, the two fields differ in skewness. In frequency terms, they differ in phase rather than in energy. A method of measuring local phase relationships

TABLE 7-1. MACRO-STATISTIC F-RATIOS

Micro-Feature	SDV	ABSAVE	POSAVE	NEGAVE
L3L3	63	2	2	2
L3E3	573	551	293	291
L3S3	345	415	378	392
E3L3	1492	1232	648	625
E3E3	977	933	887	880
E3S3	655	677	671	677
S3L3	811	727	666	672
S3E3	734	690	688	685
S3S3	700	690	688	691

TABLE 7-2. MACRO-STATISTIC CLASSIFICATION ACCURACIES

Feature Set	Global	Adaptive
SDV	85.99	85.60
ABSAVE	88.09	87.11
SDV+ABSAVE	89.16	87.55
POSAVE	85.79	87.06
NEGAVE	87.01	85.94
POSAVE+NEGAVE	85.79	87.21

is needed. One solution is to take averages of positive values instead of absolute values. We will call this the **POSAVE** statistic. It is reasonable that neurons in the visual cortex might perform such a clipping function. There might also be a balancing set of neurons responding only to luminances below average. We will compute **NEGAVE** as the negative average of macro-window values below zero.

Tables 7-1 and 7-2 show that the two one-sided measures perform slightly less well than the **SDV** and **ABSAVE** measures, although much better than the co-occurrence statistics of Section 4.2. For the present dataset there is no compelling reason to use these less powerful statistics. We shall restrict our attention to the **ABSAVE** statistic, keeping in mind that there will be some textures not discriminable by these measures. **ABSAVE** features are preferred to **SDV** features only because of their computational simplicity[6]. Both appear to be equivalent measures of texture energy for this dataset.

7.3 Micro-Feature Selection

It is desirable to reduce the feature set as much as possible. We shall begin by studying the one-dimensional features.

Table 7-3 presents individual F-ratios for the horizontal (H) and vertical (V) masks. The most striking pattern is the exceptional strength of the vertical measures contrasted with the moderate strength of corresponding horizontal measures. This reflects the presence of directional textures in the dataset. A more significant pattern is that Spot features are always the most powerful, with power gradually decreasing as the mask sequency increases. This despite the fact that Spot filters of different lengths pass different spatial frequency bands. Edge features are also strong texture

---

[6]An algorithm for computing **ABSAVE** statistics across a feature plane is documented in Appendix B.

discriminators. Level features are of no use because of the histogram equalization.

TABLE 7-3. 1-DIMENSIONAL ABSAVE F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
HL3	0	2	VL3	0	2
HE3	220	403	VE3	1335	1079
HS3	272	367	VS3	935	658
HL5	0	2	VL5	0	2
HE5	151	304	VE5	1210	1152
HS5	258	415	VS5	1385	1113
HW5	217	302	VW5	1032	737
HR5	282	337	VR5	742	543
HL7	0	2	VL7	0	3
HE7	94	178	VE7	1048	1076
HS7	240	412	VS7	1438	1292
HW7	245	356	VW7	1297	978
HR7	197	272	VR7	1044	760
HU7	205	271	VU7	847	608
HO7	291	336	VO7	695	527

Neurological studies [74] show that the visual cortex computes edge measures in approximately ten-degree increments. We have investigated diagonal one-dimensional features, although they are not properly members of the separable feature sets.

Table 7-4 lists F-ratios for one-dimensional features along the forward diagonal (**F**) and backward diagonal (**B**). The forward diagonal is from top left to bottom right. These features show far less power than corresponding

TABLE 7-4. DIAGONAL FEATURE ABSAVE F-RATIOS

Feature	Global	Adaptive	Feature	Global	Adaptive
FL3	0	2	BL3	0	2
FE3	64	95	BE3	49	62
FS3	68	70	BS3	130	131
FL5	0	2	BL5	0	2
FE5	73	119	BE5	48	79
FS5	75	107	BS5	59	67
FW5	48	46	BW5	70	67
FR5	133	121	BR5	219	197
FL7	0	2	BL7	0	2
FE7	71	102	BE7	41	69
FS7	88	144	BS7	64	92
FW7	74	98	BW7	55	58
FR7	45	45	BR7	60	56
FU7	71	65	BU7	121	115
FO7	164	144	BO7	254	224

horizontal and vertical measures. This was unexpected, even given that element spacing is somewhat wider for diagonal measures. The discriminating strengths do not even follow the same sequency pattern. The remarkable differences between rectilinear and diagonal responses must be taken as a warning that discriminating power of the separable masks may depend strongly on orientation of the training textures. Indeed, all results in this dissertation are derived from a particular dataset, and should be extrapolated with care.

Figure 7-3 presents F-ratios for two-dimensional features, rounded to the nearest hundred. The extreme discriminating power of vertical Edge and Spot features is apparent. The matrices would be symmetric if the textures

	L	E	S
L	0	6	4
E	12	9	7
S	7	7	7

	L	E	S	W	R
L	0	5	6	4	4
E	13	9	7	6	5
S	12	11	10	8	6
W	8	8	8	7	7
R	5	5	6	6	7

	L	E	S	W	R	U	O
L	0	3	7	5	4	3	4
E	11	8	6	6	5	4	4
S	13	12	10	9	8	7	5
W	10	10	9	9	8	7	6
R	7	8	8	8	7	7	6
U	6	6	6	6	6	6	6
O	5	5	5	5	6	6	6

Figure 7-3. Square Mask F-Ratios, in Hundreds

were non-directional or randomly directional. Evidently the F-ratios would then be largest along the diagonal, especially in the middle sequencies. The other important fact is the great discriminating power of even the weakest of these texture measures (excluding Level features). Very few of the co-occurrence F-ratios were as high as 300.

Joint classification accuracies for various feature subsets are given in Table 7-5. The first and second columns represent classification over globally equalized and adaptively equalized images, as in the previous chapter. The third and fourth columns are similar, but with discriminant and classification functions computed directly on the entire feature set instead of a selected subset. Stepwise analysis with the F-ratio threshold of 40.0 typically selects nine to twelve features. A lower threshold would increase the number of features, and slightly increase classification accuracy. Direct analysis usually achieves the highest possible classification accuracy, but at the cost of evaluating as many as 100 features for each pixel to be classified.

The first five rows of Table 7-5 are based on horizontal and vertical one-dimensional convolution masks. The six 3-vectors alone perform slightly better than the elaborate co-occurrence features of Chapter 4. This is amazing considering the simplicity of the texture energy method and the many experimental vindications of Haralick's co-occurrence statistics. The 5-vector statistics perform even better. Using 7-vectors or

TABLE 7-5. ABSAVE CLASSIFICATION ACCURACIES

Feature Set	Global	Adaptive	Direct Global	Direct Adaptive
H3+V3	76.51	74.76	76.90	75.34
H5+V5	82.42	81.45	83.11	81.69
H7+V7	82.57	81.54	83.98	82.28
H3+V3+H5+V5	82.08	81.59	85.45	84.38
H3+V3+H5+V5 +H7+V7	82.71	81.98	87.21	85.99
H3+V3+F3+B3	82.37	80.76	82.67	80.71
H5+V5+F5+B5	86.23	85.11	87.65	86.23
H7+V7+F7+B7	84.28	85.16	88.77	87.65
H3+V3+F3+B3 +H5+V5+F5+B5	86.62	86.43	90.48	87.94
H3+V3+F3+B3 +H5+V5+F5+B5 +H7+V7+F7+B7	85.64	86.52	84.38	90.09
3x3	84.67	82.67	84.33	83.15
5x5	86.77	86.18	88.96	87.84
7x7	87.65	86.67	89.65	88.43
3x3+5x5	88.43	87.40	90.53	89.50
3x3+5x5+7x7	88.33	86.62	92.77	92.53
H3+V3+3x3	84.91	83.06	86.47	85.25
H5+V5+5x5	86.62	85.89	90.09	88.92
H7+V7+7x7	87.70	86.91	90.87	90.33
H3+V3+3x3 +H5+V5+5x5	88.09	87.11	92.48	91.55
H5+V5+5x5 +H7+V7+7x7	88.04	86.57	93.80	93.21

combining more than one vector size gives no significant improvement.

The next five rows incorporate forward and backward diagonal statistics. Classification accuracies improve significantly. The 5-vector statistics alone are sufficient to achieve 86% classification accuracy, close to the maximum reached in this study. The combined feature sets have little more power, but provide insight into the selection process. Discriminant functions are based on vectors of all directions and sizes. Different subsets are selected in the globally equalized and adaptively equalized cases, yet all selected features are either Edge statistics or the symmetric Spot, Ripple, and Oscillation statistics. None of the antisymmetric Wave or Undulation features were found useful.

The third section of Table 7-5 shows the two-dimensional masks to be just as powerful. Length five masks are again best, although the evidence is less conclusive. The adaptively equalized **3x3+5x5** feature subset differs from the **5x5** feature subset only by inclusion of **L3S3**, the ninth and last feature to be added. The fifth analysis favors 5x5 and 7x7 features about equally. Selected statistics again differ from one analysis to another, but Wave features are rare and Undulation features are absent. The consistent inclusion of **R5R5** is somewhat surprising since matching image structures must be quite rare. This mask resembles a two-

dimensional sinc[7] or Bessel function. The similar S5S5 feature is individually very strong, but has little power when combined with other features.

The final section combines one-dimensional and two-dimensional features. It can be seen that classification accuracies improve very little. Two-dimensional features enter the models first, followed by a few of the longer vector features. Again there are few Wave and no Undulation features, despite their high individual F-ratios. Otherwise the selection seems somewhat arbitrary. Scatter diagrams show that the discriminant dimensions are the same ones found with co-occurrence features and with every other texture set we have tried. The chief difference is that there is slightly less discriminating power in the first two principal components and correspondingly more in the third component.

7.4 Summary

We have seen that one-dimensional and two-dimensional convolution masks generate powerful texture measures. Principal components analysis shows that all of the feature subsets are measuring the same texture dimensions. Several simple statistics are equally good at extracting the texture information. Further development of these methods would require a more extensive dataset.

---

[7] Sin(x)/x, an important function in image processing. It is the spatial-domain representation of a square low-pass filter. It approximates the circularly symmetric Airy pattern or Bessel function important in Fourier optics.

Perceptual studies and comparisons with known features of biological vision systems might also lead to new understanding.

In the next chapter, we will develop one set of texture energy measures into a working texture analysis system. Equivalent performance could probably be achieved with any of the feature sets presented in this chapter.

CHAPTER 8

SEGMENTATION AND CLASSIFICATION

This chapter develops a particular texture energy model into a useful texture analysis system. Coefficients are given for four principal component texture planes: these can be used as texture measures for any dataset. Classification coefficients for the eight training textures are also given. Segmentation examples show that the classifier can be used for blind segmentation of natural textures, although better coefficients for particular applications could be derived from appropriate training data or from the principal component planes.

8.1 Texture Energy Measures

Figure 1-3 shows the sequence of images used in measuring texture. The original image is first filtered with a set of small convolution masks. The filtered images are then processed with a nonlinear "local texure energy" filter. This is the **ABSAVE** moving-window average of absolute image values. Such moving-window operations are very fast even on general-purpose digital computers.

The next step in Figure 1-3 shows the linear combination of texture energy planes into a smaller number of principal component planes, typically four. This is an optional data compression step. The component images seem to represent natural texture dimensions, and to be more

"reliable" than the texture energy planes.

The final output is a segmented image or classification map. Classification is simple and fast if the texture classes are known *a priori*. Either texture energy planes or principal component planes may be used as input to the pixel classifier. Clustering or segmentation algorithms must be used if texture classes are unknown.

We saw in the last chapter that almost any set of texture energy transforms could be used to discriminate the eight textures of our dataset. 5x5 convolution masks are more powerful than 3x3 masks, and simpler than 7x7 masks. Separable square masks are easier to implement on a digital computer than rectilinear and diagonal masks. We shall therefore proceed with the 5x5 measures.

TABLE 8-1. TEXTURE ENERGY CLASSIFICATION ACCURACY

	Macro-Window Size			
Feature	3x3	7x7	15x15	31x31
LESWR	43.55	67.24	86.77	97.95
LESR	41.65	66.80	86.77	97.71
LSR	-	-	86.57	95.85
LER	-	-	86.57	-
ILESWR	35.99	58.06	85.11	97.17
ILESR	34.28	58.06	85.11	96.96
ILSR	-	-	83.89	94.97
ILER	-	-	84.30	-

Table 8-1 shows the classification accuracies achieved with different 5x5 micro-features and macro-window sizes. The letters in the feature set names stand for the vector masks of the last chapter. **LESWR**, for instance, is the set containing all two-dimensional masks made of Level, Edge, Spot, Wave, and Ripple convolutions. The letter I stands for contrast invariance. Features were made invariant by dividing pixel values in the texture energy plane by corresponding values in the **L5L5SDV** plane. **L5L5** features are otherwise excluded from all feature sets in the table. Other feature planes were computed with the **ABSAVE** macro-statistic. Tabulated values are based on 3025 samples per texture, except that 31x31 features are based on 1056 samples per texture. The table shows that classification accuracy drops rapidly as the macro-window size is reduced below 15x15. Nearly perfect classification of 31x31 blocks is possible, but we will see later that segmentation quality is poor at this resolution.

Contrast invariance has a very small effect on classification accuracy, but permits a big savings in computational cost. This is because histogram equalization is unnecessary. We shall use contrast-invariant features throughout the rest of this chapter.

All of the 15x15 feature sets perform well, even the eight-member **ILSR** and **ILER** sets. The antisymmetric Wave features are of little use. We shall confine our attention to the vector masks

$$L5 = [\ 1\ \ 4\ \ 6\ \ 4\ \ 1\ ]$$

$$E5 = [-1\ -2\ \ 0\ \ 2\ \ 1\ ]$$

$$S5 = [-1\ \ 0\ \ 2\ \ 0\ -1\ ]$$

$$R5 = [\ 1\ -4\ \ 6\ -4\ \ 1\ ]$$

Sixteen two-dimensional masks can be formed from these vectors. The number of masks could be reduced to nine or even six with little penalty, but we shall present coefficients and classification results for the full set of 15 zero-sum masks. The four most important masks for our experimental dataset are shown in Figure 8-1.

TABLE 8-2. STANDARDIZED COEFFICIENTS

Feature	Cmp 1	Cmp 2	Cmp 3	Cmp 4
IL5E5	-0.277	0.238	0.092	0.339
IL5S5	-0.105	-0.055	-0.065	-1.215
IL5R5	-0.269	0.284	0.179	1.210
IE5L5	0.204	0.331	-0.570	-0.413
IE5E5	0.011	-0.248	0.318	-1.264
IE5S5	0.188	-0.084	0.166	-0.122
IE5R5	0.123	-0.147	0.243	0.043
IS5L5	0.177	0.359	0.482	0.508
IS5E5	0.215	-0.185	0.161	1.011
IS5S5	0.026	-0.087	0.622	0.437
IS5R5	0.053	-0.313	-0.054	0.011
IR5L5	0.006	0.291	-0.371	-0.160
IR5E5	0.081	0.190	-0.265	-0.020
IR5S5	-0.168	-0.270	-0.315	-0.127
IR5R5	-0.171	-0.439	-0.693	-0.252

Relative strengths of the features may be estimated from Table 8-2. The principal component coefficients are

$$\begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

E5L5

$$\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

R5R5

$$\begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & -4 & 0 & 2 \\ 1 & 0 & -2 & 0 & 1 \end{bmatrix}$$

E5S5

$$\begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

L5S5

Figure 8-1. 5x5 Center-Weighted Masks

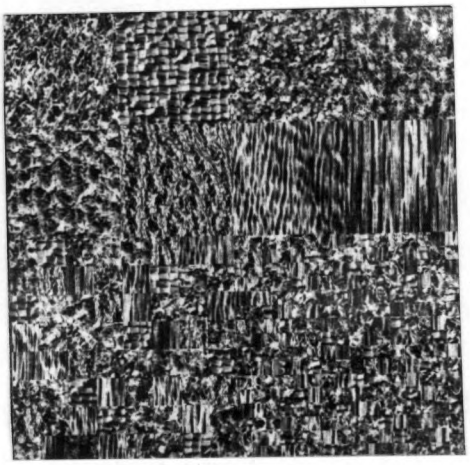
given for features reduced to zero mean and unit standard deviation. Table 8-3 gives the same coefficients for unstandardized features. These are more useful for actually computing the principal component images. Different sets of coefficients must be used for different sets of features or for different window sizes.

TABLE 8-3. UNSTANDARDIZED COEFFICIENTS

Feature	Cmp 1	Cmp 2	Cmp 3	Cmp 4
IL5E5	-4.266	3.658	1.416	5.214
IL5S5	-2.127	-1.110	-1.327	-24.721
IL5R5	-3.070	3.239	2.046	13.798
IE5L5	3.578	5.801	-9.986	-7.241
IE5E5	0.743	-17.515	22.427	-89.249
IE5S5	21.520	-9.650	18.975	-13.926
IE5R5	6.156	-7.398	12.193	2.168
IS5L5	5.466	11.079	14.891	15.721
IS5E5	25.569	-22.015	19.150	119.984
IS5S5	4.813	-16.332	117.367	82.408
IS5R5	3.936	-23.431	-4.057	0.834
IR5L5	0.128	6.609	-8.427	-3.632
IR5E5	5.995	14.112	-19.662	-1.464
IR5S5	-17.690	-28.349	-33.155	-13.345
IR5R5	-5.469	-14.050	-22.192	-8.069
Constant	-0.265	-0.148	-0.069	0.815

8.2 Pictorial Examples

Figure 8-2 shows two images which will be used to illustrate the texture energy transform. The first is a composite of the Brodatz textures. The first two rows of 128x128 blocks were taken from the centers of the Grass, Raffia, Sand, Wool, Pigskin, Leather, Water, and Wood



(a) Texture Composite

(b) House Image

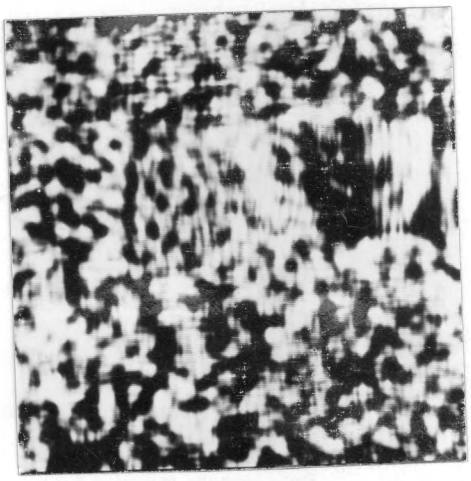
Figure 8-2. Original Images

images. Histogram equalization was applied to each block separately. The bottom-left quadrant is composed of 32x32 blocks of histogram-equalized images; the bottom-right quadrant of 16x16 blocks. The resolution is such that even trained observers would have difficulty identifying the 16x16 blocks.

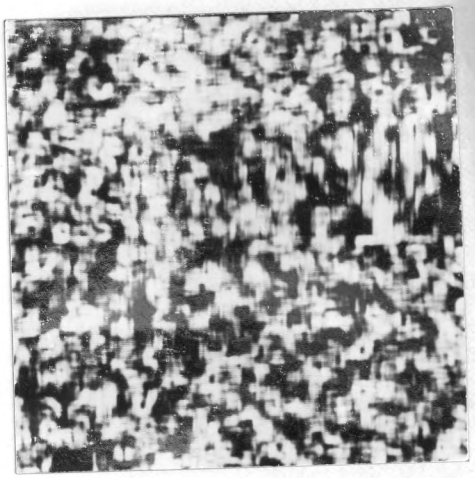
The second image is a street scene that has been used by other segmentation researchers. It is available in color, but this study is confined to monochrome segmentation. The luminance image has been subjected to histogram equalization for display. All texture transforms were computed on the unequalized version.

Figure 8-3 shows the result of convolving the original images with the **L5L5** mask. The **AVE** planes are just blurred versions of the originals. These images give some idea of the resolution actually available to a texture segmenter, since texture must be measured over a region around each pixel.

The **SDV** planes are more useful as texture feature planes. They measure local contrast. By itself this is not a good segmentation feature: it tends to locate edges rather than regions. Note how little difference there is in the **SDV** values of the different Brodatz textures. The importance of these feature planes is that they can be used to remove contrast and edge effects from other feature planes. We simply take the ratio of each feature value to the corresponding **SDV** value. This removes effects of variable scene illumination as well as reducing



(a) Composite L5L5AVE



(b) Composite L5L5SDV

(c) House L5L5AVE

(d) House L5L5SDV

Figure 8-3. Averages and Standard Deviations

the effect of edges. Even stronger normalization could be devised using the AVE image as well.

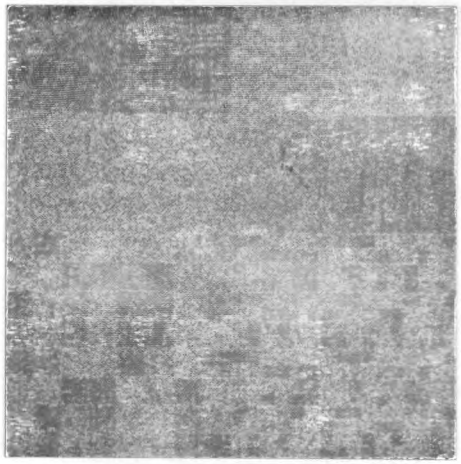
Figures 8-4 and 8-5 show the results of filtering each image with the four most important center-weighted masks. **E5L5** is a horizontal edge mask. It enhances the horizontal structure in Raffia, while hardly responding to the vertical edges in Wood. **R5R5** is a high-frequency spot detector: it produces a grainy feature plane which is very difficult to reproduce. **E5S5** is a peculiar V-shaped mask which responds best to textures with low correlation. In the House image it seems to enhance diagonal edges. **L5S5** is a vertical line detector. It enhances vertical edges, particularly repetitive ones such as in Water and Wood.

Figures 8-6 and 8-7 show the effect of the ABSAVE texture energy transform prior to normalization with the SDV plane. The separation of textures in the Composite image is obvious. Careful examination of the House images shows that different parts of the scene also have different relative brightnesses in the different texture energy planes. It should be remembered that only four of 15 texture planes are illustrated.

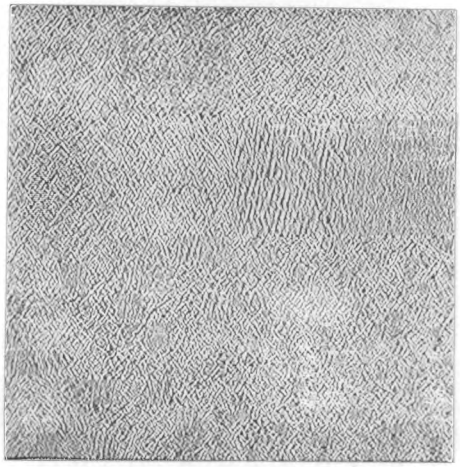
Figures 8-8 and 8-9 are particular linear combinations of the 15 texture energy planes (after normalization). The linear combinations are principal component transformations for the eight Brodatz textures. The Composite images look very similar to texture energy planes, but the bright and dark areas are more uniform.

The House images do not strongly resemble the texture

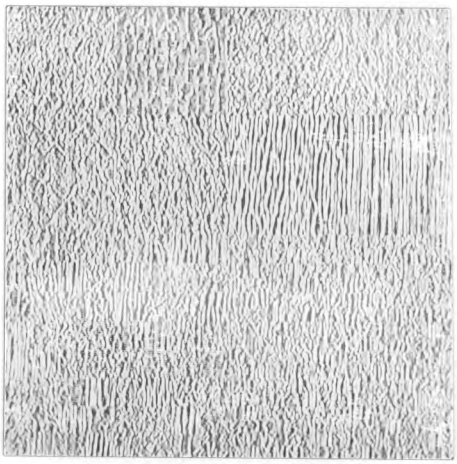
(a) E5L5



(b) R5R5



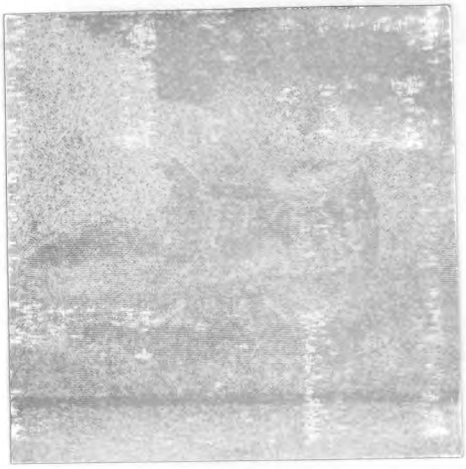
(c) E5S5



(d) L5S5

Figure 8-4. Filtered Image Planes, Composite

(a) E5L5

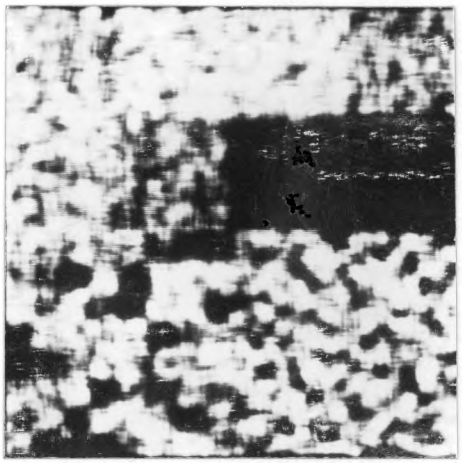


(b) R5R5

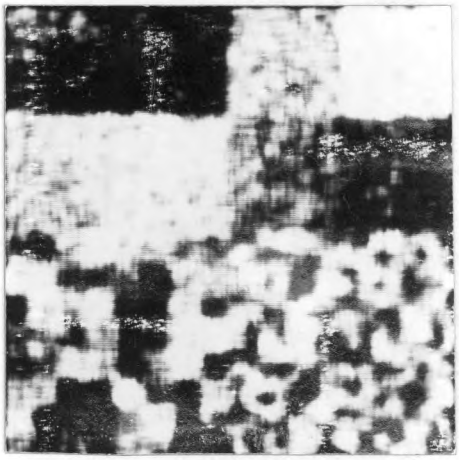
(c) E5S5

(d) L5S5

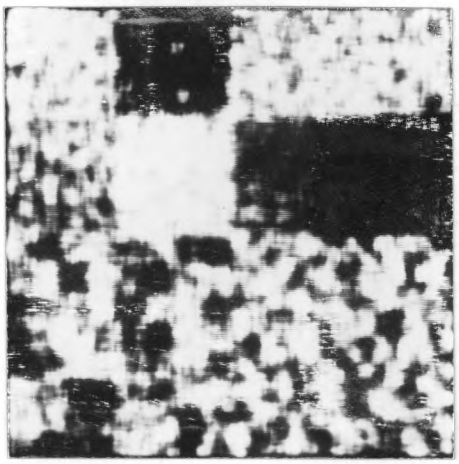
Figure 8-5. Filtered Image Planes, House



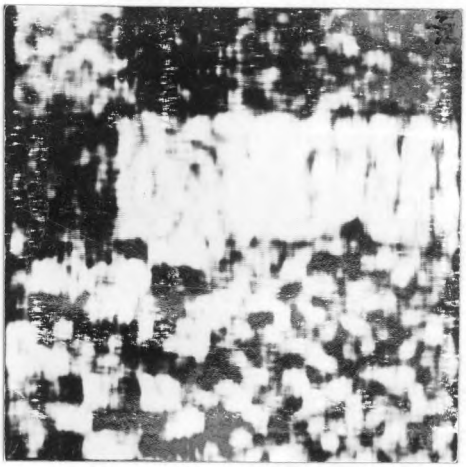
(a) E5L5



(b) R5R5



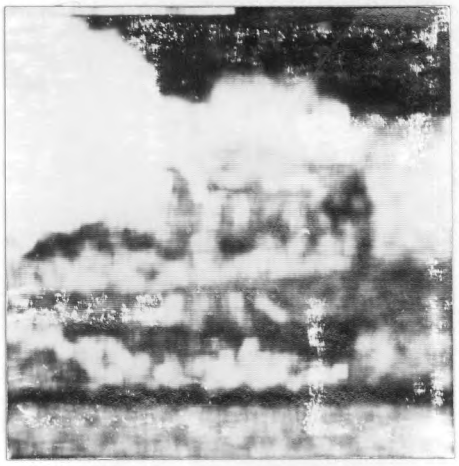
(c) E5S5



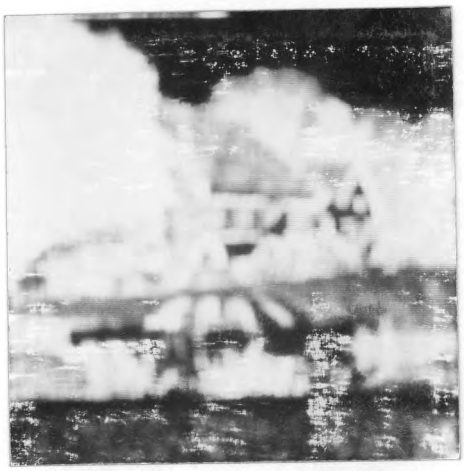
(d) L5S5

Figure 8-6. Texture Energy Planes, Composite

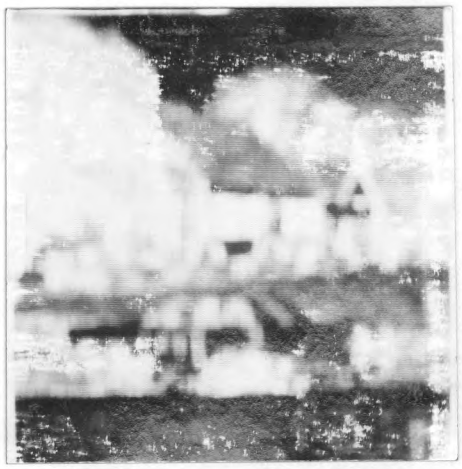
(a) E5L5



(b) R5R5

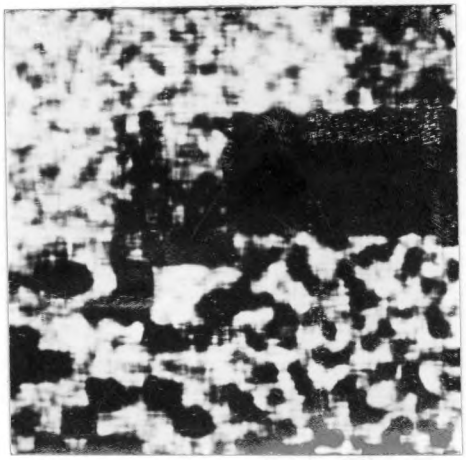


(c) E5S5

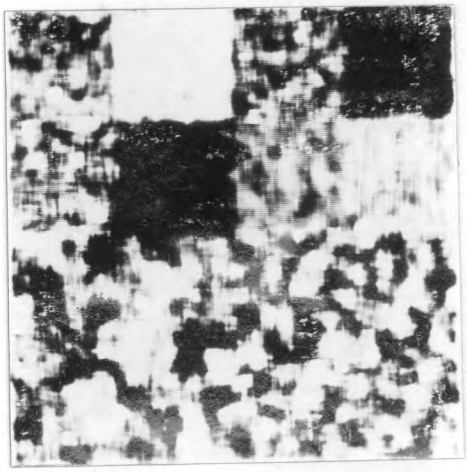


(d) L5S5

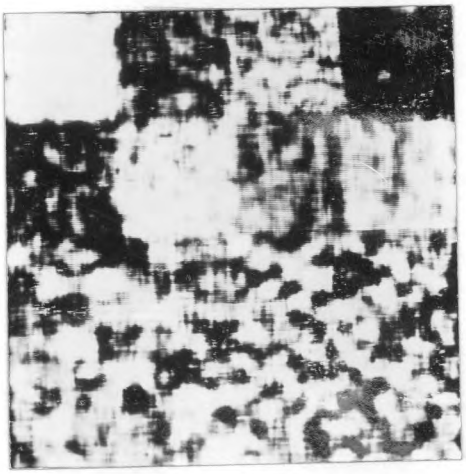
Figure 8-7. Texture Energy Planes, House.



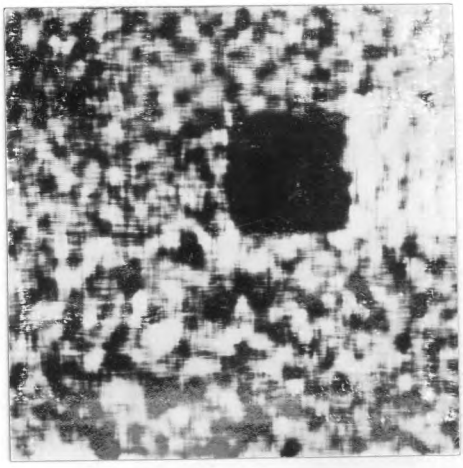
(a) First Component



(b) Second Component

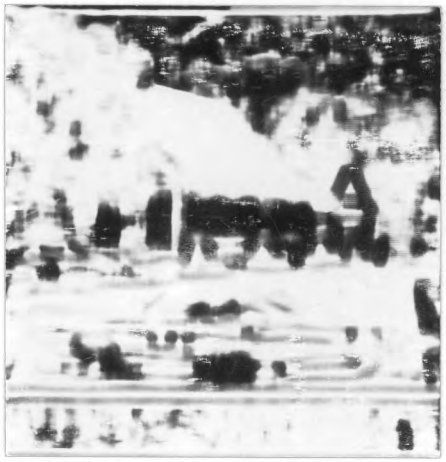


(c) Third Component

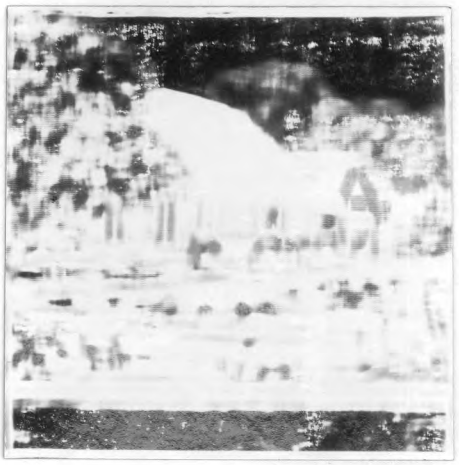


(d) Fourth Component

Figure 8-8. Principal Components, Composite



(a) First Component



(b) Second Component

(c) Third Component

(d) Fourth Component

Figure 8-9. Principal Components, House

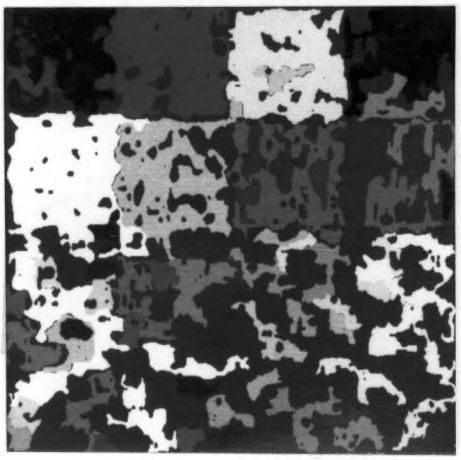
energy planes, perhaps because of contrast reversals. The discriminant planes are not necessarily principal component planes for the House textures, but their discriminating power is obvious.

8.3 Segmentation and Classification

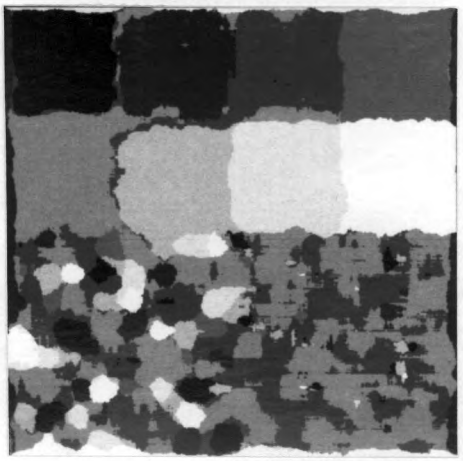
This section will illustrate the quality of image segmentation which can be obtained with texture energy measures. Two approaches will be shown, blind segmentation and classification with a priori knowledge of the texture class statistics. We will use a nearest-centroid or maximum-likelihood linear classifier as described in Appendix C.

Blind segmentation requires clustering of the image data to determine the number and types of regions present. There are many multivariate clustering algorithms, but few designed to segment images. One of the best is the "Ohlander segmenter" now maintained by Dr. Keith Price [46]. We have used this computer program without modification, despite the compromises required. The first three principal component planes were used as red, green, and blue color planes. The fourth principal component plane was not used. The segmenter thus had no way to distinguish between Water and Wood. Further, the principal component planes are unimodal and quite unlike natural color planes for which the segmenter was designed. Color transformations (Y-I-Q and Saturation-Hue-Intensity) had to be used to aid the segmenter.

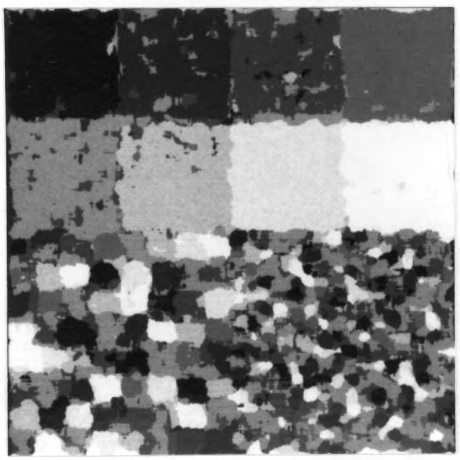
The first image in Figure 8-10 shows the result of



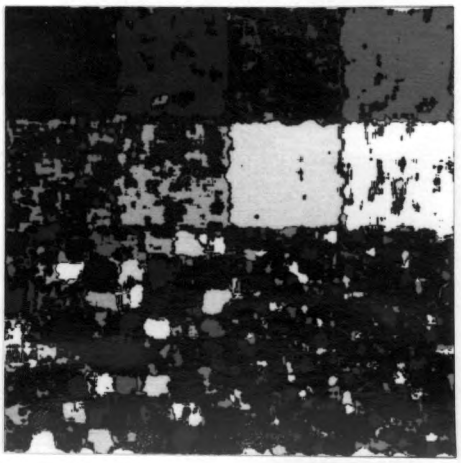
(a) 15x15 Segmentation



(b) 31x31 Classification



(c) 15x15 Classification



(d) Partial Classification

Figure 8-10. Segmentation, Composite

segmenting the Composite picture. The 128x128 blocks are reasonably well separated into seven texture classes. The 32x32 and 16x16 blocks are not resolved.

The second image shows classification results using 31x31 macro-window statistics for the eight texture classes. Large regions are almost perfectly classified, but 32x32 regions are only partially separated. The 16x16 regions are not resolved.

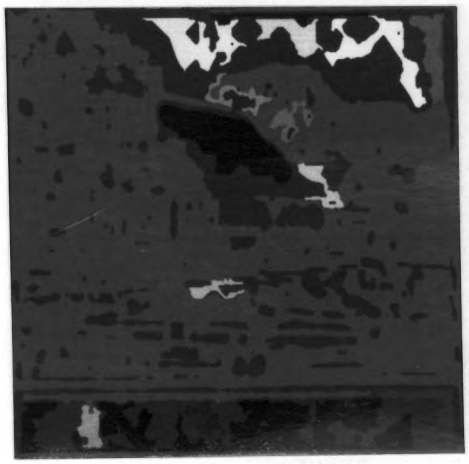
The third image, classified with 15x15 features, is a better segmentation of the scene. The Wool, Water, and Wood textures are almost perfectly identified; other textures have at least 78% accuracy across the original 512x512 images. Errors tend to occur in patches. Neither the classification nor the principal component measures tend to "go wild" near region boundaries. Table 8-4 gives the coefficients used to compute the discriminant functions. Each pixel is assigned to the class with the highest function value.

The fourth image is identical to the third, but with doubtful classifications suppressed (shown as black). Classification was skipped unless the highest classification function exceeded the second highest by at least 20%. It can be seen that some texture types are less "certain" than others.

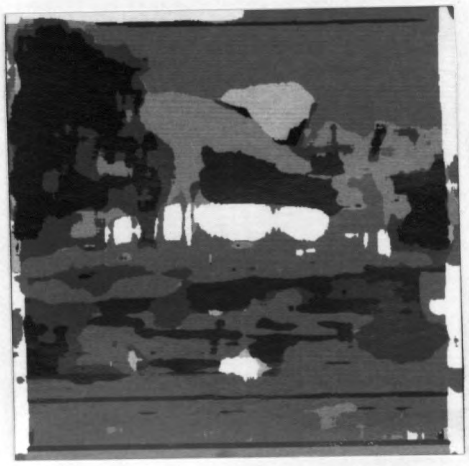
Figure 8-11 repeats the classification sequence for the House image. Blind segmentation performed very badly on this image. The results of texture classification are surprisingly good considering that pixels are being

TABLE 8-4. CLASSIFICATION COEFFICIENTS

Actual	Grass	Raffia	Sand	Wool	Pigskn	Lthr	Water	Wood
IE5L5	177	216	176	180	169	202	221	273
IL5S5	-153	-190	-162	-188	-156	-178	-57	-195
IL5R5	5	18	4	5	4	1	0	98
IE5L5	253	353	282	285	278	215	274	203
IE5E5	-411	-739	-368	-700	-403	-354	-270	-691
IE5S5	515	591	233	441	337	757	147	46
IE5R5	65	-22	12	12	-71	63	-95	-96
IS5L5	-207	-138	-227	-333	-316	-222	-391	-254
IS5E5	957	411	846	871	940	547	65	658
IS5S5	222	-895	-333	-655	-798	-539	-876	-256
IS5R5	-64	-105	-22	135	-136	103	-71	-163
IR5L5	-17	71	-13	2	78	-14	33	16
IR5E5	4	166	79	38	175	-88	22	-12
IR5S5	-240	-372	-15	140	34	-112	245	71
IR5R5	-125	-58	-19	151	-10	35	120	-8
Constant	-32	-37	-29	-30	-27	-29	-34	-33



(a) 15x15 Segmentation



(b) 31x31 Classification

(c) 15x15 Classification

(d) Partial Classification

Figure 8-11. Segmentation, House

classed as Raffia, Leather, etc. Major semantic regions are isolated in all three versions, except that the car and lawn are not separated. Note that the piece of cellophane tape in the lower-right corner is differentiated from its white background.

TABLE 8-5. CLASS CONFUSION, PERCENT

	Predicted							
Actual	Grass	Raffia	Sand	Wool	Pigskn	Lthr	Water	Wood
Grass	77.8	0.7	9.9	0.4	0.9	10.3	0.0	0.1
Raffia	0.5	91.8	3.1	0.0	4.5	0.1	0.0	0.0
Sand	4.4	0.6	80.8	0.4	9.7	4.1	0.0	0.0
Wool	0.2	0.0	6.2	86.9	4.1	2.6	0.0	0.0
Pigskin	0.4	2.0	15.2	1.1	81.2	0.2	0.0	0.0
Leather	2.3	0.0	4.0	0.9	0.1	92.5	0.3	0.0
Water	0.0	0.0	0.0	2.8	0.3	0.1	91.2	5.6
Wood	0.0	0.0	0.0	0.0	0.0	0.4	2.7	96.9

Tables 8-5 and 8-6 show the relative separation of the eight texture classes in the principal component space. Pigskin and Sand are often confused, although it is difficult to say why. Grass is often classified as Sand or Leather: the errors are nearly all in the upper third of the Grass image, which is in much sharper focus than the rest.

TABLE 8-6. PAIRWISE F-RATIOS

	Grass	Raffia	Sand	Wool	Pigskn	Lthr	Water	Wood
Grass	-	2639	623	3187	1649	1013	4581	5087
Raffia	2639	-	1746	4193	1567	3780	4814	5378
Sand	623	1746	-	1437	495	1005	3635	4796
Wool	3187	4193	1437	-	1647	1572	3570	5034
Pigskin	1649	1567	495	1647	-	1998	3500	4885
Leather	1013	3780	1005	1572	1998	-	2562	3700
Water	4581	4814	3635	3570	3500	2562	-	1934
Wood	5087	5378	4796	5034	4885	3700	1934	-

15 and 24,178 degrees of freedom

8.4 Timing Estimates

Table 8-7 shows the amount of computing time required for various operations. The total time required to segment an image depends on the options chosen. It can vary from 30 to 50 minutes with the present implementation.

Most of the run time is consumed by convolutions and matrix cumulations. The convolutions are quite fast, but could be speeded with special hardware or optimized code for each mask. The number of filtered images, and hence the number of texture energy planes, could also be cut in half with very little ill effect.

Cumulation of matrices takes only six seconds per 512x512 plane, but there are a large number of such operations. The operation itself could be reduced to half the time by using optimal techniques. The number of cumulations could also be reduced by computing

TABLE 8-7. TIMING FOR 15X15 CLASSIFICATION

Operation	Seconds	Total Minutes
Image Input	21	.35
L5L5 Convolution	57	.95
AVE, SDV Computation	41	.68
AVE, SDV Output	34	1.12
Convolutions (15)	57	14.18
Feature Plane Output (4)	34	2.23
Energy Measurement (15)	15	3.78
Energy Plane Output (4)	34	2.23
Component Initialization (4)	0	.03
Component Cumulation (15x4)	6	6.20
Component Output (4)	34	2.23
Class Initialization (8)	3	.35
Class Cumulation (15x8)	6	12.38
Classification	45	.75
Classification Output	34	.57
		48.05

classifications from the principal component planes instead of the texture energy planes. This savings grows linearly with the number of texture classes and with the number of feature planes.

Real-time implementation of texture description is quite possible. Digital hardware for 3x3 convolution is already available. The additional accuracy of 5x5 processing could be obtained with two 3x3 stages or with a 1x5 and a 5x1 stage. Only the macro-||window energy transform remains to be developed. The chief problem is

the number of image rows which must be held in memory. This could be reduced to zero by using a "fading memory" energy transform instead of an accurately updated moving window transform.

CHAPTER 9
CONCLUSIONS

We have surveyed the literature of texture analysis, developed an experimental method of comparing texture measures, evaluated co-occurrence and correlation statistics, tested hundreds of spatial-statistical operators, documented a new texture energy approach, and implemented a texture classification system. It is time to review these accomplishments and to suggest further research.

9.1 Summary

Attempts at quantitative texture measurement began at least two decades ago. Most of the tools of engineers and computer scientists have been tried, including classification, correlation, linear prediction, Fourier analysis, joint density estimation, cluster analysis, and syntactic analysis. Few methods have proven useable.

We have chosen to study high-resolution natural textures. These have been modified to have identical histograms, making texture analysis the only way to tell them apart. Any procedure which can accurately classify the image pixels must therefore be measuring texture. Relative classification accuracy for a particular dataset can be used as a quality measure.

The class of co-occurrence statistics was

investigated. Several methods of information extraction were tried, with little improvement over the Haralick measures. Classification accuracy could not be raised above 72% for our experimental dataset.

Augmented autocorrelation statistics were also evaluated. Classification accuracy was limited to 65%, and this was achievable without using autocorrelation measures. The Laplacian operator was found to extract more texture information than the Sobel gradient magnitude or Markov whitening operators.

The Laplacian method led to a more general class of spatial-statistical transforms. Hundreds of operators were tried, including statistical moments, spatial moments, rotation-invariant and contrast-invariant moments, joint spatial-statistical moments, combined 3x3 and 5x5 moments, and a large class of ad hoc convolution operators. Classification accuracies above 88% were achieved, but no one system was satisfactory.

Texture energy transforms were then developed. They are a class of spatial-statistical transforms, and incorporate all of the lessons learned in earlier work. The essence of this approach is local measurement of the energy passed by a set of symmetric and antisymetric filters. Classification accuracies as high as 94% were achieved, despite the simplicity of the algorithm.

A particular set of 5x5 masks was chosen for the final analysis system. The outputs of 15 filters, normalized by local contrast, were used to build principal

component planes and classification maps. Average classification accuracy within large areas was 87%, with sufficient resolution to identify elements in a mosaic of 16x16 texture blocks. The ability to trade resolution for higher accuracy was also demonstrated.

9.2 Iterative Improvement

Texture segmentation, as discussed so far, is a preprocessing technique for locating uniformly textured regions. The next step is to apply more specific knowledge sources to improve the segmentation or classification.

Initial segmentation of a texture image may be done with known prototypes (such as wheat, corn, forest, etc.) or with cluster centers extracted from the image data. In either case it is desirable to re-examine regions to compute more accurate texture statistics than were used in the initial segmentation.

The improved statistics may be used for reclassifying pixels along the region borders. This amounts to hypothesis testing, since the pixel is to be assigned to one texture field or the other, or to a third region such as a river or road separating the first two. The linear prediction technique of Deguchi and Morishita [18] could be adapted to this purpose, as could the relaxation methods of other researchers [83], [84].

9.3 Modeling of Natural Textures

A major application of texture perception is the interpretation of aerial photographs. If textured areas are to be identified, we must start with a training set of known textures. The parameters of these textures can be used as prototypes or design constraints in the development of classifiers.

Image textures are dependent on the imaging system with which they were created. Humans are able to compensate for changing imaging conditions, but artificial vision systems have not yet mastered this trick. It is therefore necessary to study the effect on texture features of changes in scale, illumination, rotation, geometric warp, atmospheric blur, optical aberrations, film or detector noise, and method of quantization. Texture energy features are particularly well suited for this type of modeling.

9.4 Perceptual Modeling

Texture description must ultimately be done in human terms. It would be useful to know how texture energy measures correlate with human texture perception. Texture energy processing seems similar to known functions of the visual cortex, but such claims need to be substantiated.

One area needing research is the processing of texture in color imagery. It is doubtful that natural vision systems determine texture separately in each color plane, but such methods have been suggested for digital systems. Perhaps such methods can extract more

information from multispectral imagery than is now possible. For perceptual modeling, it is more likely that texture is computed only in an adaptively processed luminance plane.

9.5 Texture Synthesis

Image synthesis is the opposite of image understanding, just as reconstruction is the opposite of compression. Both are attempts to display data in a form which humans can readily understand.

Texture synthesis is most useful for background regions. These can be transmitted or stored as sets of shape and texture parameters, then synthesized for visual display. For large background regions this permits tremendous data compression.

Some texture measures are well suited to synthesis. Haralick's co-occurrence statistics can be directly implemented as pixel-generating probabilities, and Pratt's method [8] can be used to generate texture fields from correlation statistics. The whitening method of Faugeras and Pratt [63] can also be reversed to generate textures. It has not yet been determined whether texture energy measures can be used for synthesis.

9.6 Conclusions

In retrospect, texture analysis does not seem such a difficult problem. A fast and elegant solution has been found. We have shown that texture energy measures effectively discriminate texture fields, and that they can be used for segmentation of natural images.

Texture energy measures have much in common with the Fourier statistics of Lendaris and Stanley, and with the spot density, edge density, and variance statistics of other researchers. No doubt other descriptions for this analysis method will be found, but the concept of local pattern energy is firmly established.

APPENDICES

APPENDIX A

HISTOGRAM EQUALIZATION

Each image used as input to the analysis routines was first equalized to compensate for differences in illumination and processing. Each image or feature plane printed in this document was equalized to make maximum use of the limited dynamic range of the printing process.

The following program is the core of the histogram equalization procedure used in this study. It is part of the VCTLIB segment of the SAILIB library of image processing routines written and maintained by the author. The subroutine is written in SAIL.

```
INTERNAL PROCEDURE EQLCUT
  (INTEGER ARRAY IMG!HST;
  REFERENCE INTEGER ARRAY CUT!PNT);

COMMENT
  Purpose:
    Segments a histogram vector into equal portions.
  Author:
    Kenneth I. Laws.
  Last Revision:
    March 5, 1979.
  Input:
    IMG!HST is the original histogram.  It should have
    increasing indexing and non-negative elements.

  Output:
    CUT!PNT should be indexed from 1 through the number of
    probability bins desired.  Each element of CUT!PNT
    will be set to the highest index of the original
    histogram which should be assigned to that bin.  The
    last cutpoint will always be the highest index of
    IMG!HST.
```

```
  Remarks:
    The cutpoints are similar to percentiles or quantiles.
    Each cutpoint is chosen to minimize the error in the
    cumulative probability up to and including that bin.
    Slightly different results might be obtained by
    starting at the other end, and there are a few
    histograms for which this algorithm will not yield
    good results.  For an optimal equalization algorithm
    see S.-K. Chang and Y.-W. Wong, Communications of the
    ACM, Oct. 1978.  The algorithm used here is similar to
    the EPQ method of Richard Conners (which is similar to
    that of Haralick), except that cutpoints are matched
    to percentage of total probability rather than
    percentage of remaining probability.
END COMMENT;

BEGIN "EQLCUT"

  INTEGER MIN!IMG!VAL,MAX!IMG!VAL,N!BINS;

  "Determine the old and new histogram limits."
  MIN!IMG!VAL := ARRINFO(IMG!HST,1);
  MAX!IMG!VAL := ARRINFO(IMG!HST,2);
  N!BINS := ARRINFO(CUT!PNT,2);

  "Allocate a vector for the cumulative histogram."
  BEGIN "ALLOCATE"

    INTEGER NOW!VAL,TTL!CNT,LST!CUT,NOW!CUT;
    INTEGER ARRAY HST!SUM[MIN!IMG!VAL:MAX!IMG!VAL];

    "Form the cumulative histogram."
    TTL!CNT := 0;
    FOR NOW!VAL := MIN!IMG!VAL STEP 1 UNTIL MAX!IMG!VAL DO
      HST!SUM[NOW!VAL]
        := (TTL!CNT := TTL!CNT+IMG!HST[NOW!VAL]);

    "Determine the requantization cutpoints."
    LST!CUT := MIN!IMG!VAL;
    FOR NOW!CUT := 1 STEP 1 UNTIL N!BINS DO BEGIN "CUTPNT"

      INTEGER NOW!VAL,NOW!TTL;
      REAL EQL!TTL,OLD!ERR;

      "Compute the threshold for this bin."
      EQL!TTL := TTL!CNT*NOW!CUT/N!BINS;
      OLD!ERR := TTL!CNT+1;
```

```
      "Find the highest cutpoint for which
        the error is minimum."
      FOR NOW!VAL := LST!CUT STEP 1 UNTIL MAX!IMG!VAL DO
        BEGIN "FNDCUT"

        REAL NOW!ERR;

        NOW!TTL := HST!SUM[NOW!VAL];
        NOW!ERR := ABS(EQL!TTL-NOW!TTL);
        IF OLD!ERR < NOW!ERR THEN DONE "FNDCUT";
        OLD!ERR := NOW!ERR;
        CUT!PNT[NOW!CUT] := NOW!VAL;
      END "FNDCUT";

      LST!CUT := CUT!PNT[NOW!CUT];
    END "CUTPNT";
  END "ALLOCATE";
END "EQLCUT";
```

APPENDIX B

MACRO WINDOW STATISTICAL TRANSFORM

This section documents the algorithm used to compute the ABSAVE macro feature plane from a micro feature plane. The computation of the macro window statistic is done block by block to save storage. This block size has no relation to the window size. Within each block, the transformation is done by a moving-window algorithm. The code to compute statistical moments is similar, but much more complicated.

```
INTERNAL PROCEDURE ABSAVE
  (SAFE REAL ARRAY IMG!MTX;
  INTEGER MIN!BLK!ROW,MIN!BLK!COL;
  REFERENCE SAFE REAL ARRAY AVE!MTX;
  INTEGER WDW!SZE);

COMMENT
  Purpose:
    Computes the mean absolute level around each pixel.
  Author:
    Kenneth I. Laws.
  Last Revision:
    August 26, 1979.

  Input:
    IMG!MTX must be a matrix with at least WDW!SZE%2 rows
    and columns surrounding the desired sub-block.  The
    data block will be a submatrix the same size as
    AVE!MTX.  The square window size must be an odd
    integer.  It may be larger or smaller than the block
    size.  The non-spatial moments will be computed within
    a window of this size around each pixel of the data
    block.
```

```
  Output:
    The output matrix must not be the same as the input
    matrix.  Each element of OUT!MTX will be assigned the
    average of absolute values in the corresponding data
    window.

  Remarks:
    The algorithm is linear in the block size (squared),
    and constant in the window size!
    Note that the arguments are real arrays.  This is more
    general than using integer arithmetic, but slower on
    some machines.
END COMMENT;

BEGIN "ABSAVE"

  ! Require SUBLIB procedures;
  EXTERNAL PROCEDURE ADDFLT(REAL NEW!VAL;
    REFERENCE REAL FLT!VAL);

  INTEGER MIN!OUT!ROW,MAX!OUT!ROW,BLK!ROWS,MIN!OUT!COL,
    MAX!OUT!COL,BLK!COLS,HLF!WDW!SZE,MIN!IMG!ROW,
    MAX!IMG!ROW,MIN!IMG!COL,MAX!IMG!COL;
  REAL SZE!FCTR;

  "Check validity of the input arguments."
  HLF!WDW!SZE := WDW!SZE%2;
  IF NOT (3 <= WDW!SZE < 512) OR WDW!SZE = 2*HLF!WDW!SZE
    THEN USERERR(0,1,
      "ABSAVE: WDW!SZE must be a small odd integer.");

  "Determine the data and output block dimensions."
  MIN!OUT!ROW := ARRINFO(AVE!MTX,1);
  MAX!OUT!ROW := ARRINFO(AVE!MTX,2);
  BLK!ROWS := MAX!OUT!ROW+1-MIN!OUT!ROW;
  MIN!OUT!COL := ARRINFO(AVE!MTX,3);
  MAX!OUT!COL := ARRINFO(AVE!MTX,4);
  BLK!COLS := MAX!OUT!COL+1-MIN!OUT!COL;

  "Set dimensions for the augmented image block."
  MIN!IMG!ROW := MIN!BLK!ROW-HLF!WDW!SZE;
  MAX!IMG!ROW := MIN!BLK!ROW+BLK!ROWS+HLF!WDW!SZE-1;
  MIN!IMG!COL := MIN!BLK!COL-HLF!WDW!SZE;
  MAX!IMG!COL := MIN!BLK!COL+BLK!COLS+HLF!WDW!SZE-1;

  "Precompute the window size factor."
  SZE!FCTR := 1.0/WDW!SZE^2;

  "Use block structure to allocate working vectors."
  BEGIN "ALLOCATE"
```

```
INTEGER MIN!WDW!ROW,MAX!WDW!ROW,IMG!COL,IMG!ROW,
  OUT!ROW;
REAL WDW!SUM;
SAFE REAL ARRAY COL!SUM[MIN!IMG!COL:MAX!IMG!COL];

"Set pointers to the top and bottom window rows."
MIN!WDW!ROW := MIN!IMG!ROW;
MAX!WDW!ROW := MIN!WDW!ROW+WDW!SZE-1;

"Load the accumulator vector."
ARRCLR(COL!SUM);
FOR IMG!COL := MIN!IMG!COL STEP 1 UNTIL MAX!IMG!COL DO
  FOR IMG!ROW := MIN!WDW!ROW STEP 1 UNTIL MAX!WDW!ROW
    DO ADDFLT(ABS(IMG!MTX[IMG!ROW,IMG!COL]),
      COL!SUM[IMG!COL]);

"Compute and store the local average plane."
FOR OUT!ROW := MIN!OUT!ROW STEP 1 UNTIL MAX!OUT!ROW DO
  BEGIN "ONEROW"

  INTEGER MIN!WDW!COL,MAX!WDW!COL,OUT!COL;

  "Update the column sum except on the first time."
  IF OUT!ROW > MIN!OUT!ROW THEN BEGIN "UPDATE"
    MAX!WDW!ROW := MAX!WDW!ROW+1;
    FOR IMG!COL := MIN!IMG!COL STEP 1 UNTIL
      MAX!IMG!COL DO
      ADDFLT(ABS(IMG!MTX[MAX!WDW!ROW,IMG!COL])
        -ABS(IMG!MTX[MIN!WDW!ROW,IMG!COL]),
          COL!SUM[IMG!COL]);
    MIN!WDW!ROW := MIN!WDW!ROW+1;
  END "UPDATE";

  "Set pointers to the left and right window columns."
  MIN!WDW!COL := MIN!IMG!COL;
  MAX!WDW!COL := MIN!WDW!COL+WDW!SZE-1;

  "Load the cumulative total for the 'zeroth' block."
  WDW!SUM := 0.0;
  FOR IMG!COL := MIN!WDW!COL STEP 1 UNTIL MAX!WDW!COL
    DO WDW!SUM := WDW!SUM + COL!SUM[IMG!COL];

  "Compute the sums for this row.  Use trick
    initialization of MIN!WDW!COL to start the loop."
  MIN!WDW!COL := MAX!WDW!COL;
  MAX!WDW!COL := MAX!WDW!COL-1;
  FOR OUT!COL := MIN!OUT!COL STEP 1 UNTIL MAX!OUT!COL
    DO BEGIN "WDWSUM"
```

```
        "Center the block total on the new column."
        MAX!WDW!COL := MAX!WDW!COL+1;
        WDW!SUM := WDW!SUM
          + COL!SUM[MAX!WDW!COL]-COL!SUM[MIN!WDW!COL];
        MIN!WDW!COL := MAX!WDW!COL+1-WDW!SZE;

        "Store the average of absolute values."
        AVE!MTX[OUT!ROW,OUT!COL] := WDW!SUM*SZE!FCTR;
      END "WDWSUM";
    END "ONEROW";
  END "ALLOCATE";
END "ABSAVE";
```

APPENDIX C

DISCRIMINANT ANALYSIS

All discriminant analyses used in this study were done with the SPSS statistical analysis system. This package is available from SPSS, Inc., Suite 3300, 444 N. Michigan Ave., Chicago, IL 60611.

The mathematical basis of the SPSS algorithms [85] is given below. The formulas have been simplified by the assumptions that the texture classes are equally likely and that the same number of samples have been taken from each class, conditions that were satisfied throughout this study.

C.1 Notation

f_{klm} the value of feature $l = 1,\ldots,L$ for sample $m = 1,\ldots,M$ within texture class $k = 1,\ldots,K$.

N the total number of texture samples.

Within-Group Sums of Cross-Products Matrix

$$w_{ij} = \sum_{k=1}^{K}\sum_{m=1}^{M} f_{kim}f_{kjm} - \frac{1}{M}\sum_{k=1}^{K}\left(\sum_{m=1}^{M} f_{kim}\right)\left(\sum_{m=1}^{M} f_{kjm}\right)$$

Total Sums of Cross-Products Matrix

$$t_{ij} = \sum_{k=1}^{K} \sum_{m=1}^{M} f_{kim} f_{kjm} - \frac{1}{N} \left(\sum_{k=1}^{K} \sum_{m=1}^{M} f_{kim} \right) \left(\sum_{k=1}^{K} \sum_{m=1}^{M} f_{kjm} \right)$$

Tolerance

$$TOL_1 = \begin{cases} 0 & \text{if } w_{11} = 0 \\ w^*_{11}/w_{11} & \text{variable 1 not in the model} \\ -1/w^*_{11} w_{11} & \text{otherwise} \end{cases}$$

F-to-remove

$$F_1 = \frac{(w^*_{11} - t^*_{11})/(K-1)}{t^*_{11}/(N+1-K-q)}$$

F-to-enter

$$F_1 = \frac{(t^*_{11} - w^*_{11})/(K-1)}{w^*_{11}/(N-K-q)}$$

Wilks' Lambda

$$\text{LAMBDA} = |\underline{W}_{11}|/|\underline{T}_{11}|$$

with degrees of freedom q, K-1, and N-K.

C.2 Variable Selection

SPSS permits either direct or stepwise entry of variables into the model. This study used stepwise entry with the threshold constants given below. At each step:

- Each variable in the model is considered for removal. A variable is eligible for removal if its F-to-remove is less than FOUT=40. If more than one is eligible, that variable is removed which leaves the lowest Wilks' lambda for the remaining model. Variables are then re-evaluated and removal continues until no more variables are eligible.

- The best variable not in the model is then selected. A variable is not considered if its inclusion would cause the tolerance of any included variable (or its own tolerance) to drop below TOLERANCE=0.0001. Neither is it considered if its F-to-enter is less than FIN=40. The eligible variable with the highest F-to-enter is then included in the model.

- Processing stops when no more variables are eligible for inclusion.

During variable selection, the matrix $\underline{W}$ is replaced at each step by matrix $\underline{W}^*$. If the first q variables have been included, we partition $\underline{W}$ to be

$$\underline{W} = \begin{bmatrix} \underline{W}_{11} & \underline{W}_{12} \\ \underline{W}_{21} & \underline{W}_{22} \end{bmatrix}$$

where $\underline{W}_{11}$ is qxq. Then

$$\underline{W}^* = \begin{bmatrix} -\underline{W}_{11}^{-1} & \underline{W}_{11}^{-1}\underline{W}_{12} \\ \underline{W}_{21}\underline{W}_{11}^{-1} & \underline{W}_{22} - \underline{W}_{21}\underline{W}_{11}^{-1}\underline{W}_{12} \end{bmatrix}$$

or, by definition,

$$\underline{W}^* = \begin{bmatrix} \underline{W}_{11}^* & \underline{W}_{12}^* \\ \underline{W}_{21}^* & \underline{W}_{22}^* \end{bmatrix}$$

$\underline{T}$ is similarly replaced by $\underline{T}^*$.

C.3 Fischer's Linear Discriminant Functions

$$b_{ik} = (N-K) \sum_{l=1}^{q} w^*_{il} \bar{f}_{ki} \qquad i = 1,\ldots,q$$

$$a_k = \log \frac{1}{K} - \frac{1}{2} \sum_{l=1}^{q} b_{lk} \bar{f}_{kl}$$

where

$$\bar{f}_{kl} = \frac{1}{M} \sum_{m=1}^{M} f_{klm}$$

C.4 Canonical Discriminant Functions

The canonical discriminant function coefficients are determined by solving the general eigenvalue problem

$$(\underline{T}-\underline{W})\underline{V} = \underline{DWV}$$

where $\underline{V}$ is the unscaled matrix of discriminant function coefficients and $\underline{D}$ is a diagonal matrix of eigenvalues. The eigensystem is solved as follows:

$$\underline{W} = \underline{LU}$$

is formed (Cholesky decomposition), where $\underline{L}$ is a lower triangular matrix and $\underline{U} = \underline{L}'$.

The symmetric matrix $\underline{L}^{-1}\underline{BU}^{-1}$ is formed and the system

$$(\underline{L}^{-1}(\underline{T}-\underline{W})\underline{U}^{-1} - \underline{D})\underline{UV} = \underline{0}$$

is solved using tridiagonalization and the QL method. The result is $r = \min(q,K-1)$ eigenvalues and corresponding orthonormal eigenvectors $\underline{UV}$. The eigenvectors of the original system are

$$\underline{V} = \underline{U}^{-1}(\underline{UV})$$

ordered by decreasing magnitude of eigenvalue. The standardized canonical discriminant coefficient matrix is

$$\text{diag}(w_{11}^{1/2}, w_{22}^{1/2}, \ldots, w_{qq}^{1/2})\underline{V}_1$$

where $\underline{V}_1$ is the matrix of eigenvectors such that

$$\underline{V}_1\underline{w}_{11}\underline{V}_1 = \underline{I}$$

C.5 Classification

Let $\underline{f}$ be the **1xq** vector of discriminating variables for a particular texture sample. The **1xr** vector of canonical discriminant function values is

$$\underline{d} = \underline{fB} + \underline{a}$$

A chi-square distance from each centroid is computed as

$$x_k = (\underline{d} - \underline{d}_k)(\underline{d} - \underline{d}_k)'$$

where $\underline{d}_k$ is the mean vector for class k. The distribution of x_k is chi-square with r degrees of freedom if the texture sample is a member of class k.

The classification, or posterior, probability is

$$P(k|\underline{d}) = \frac{e^{-x/2}}{\sum_{i=1}^{K} e^{x/2}}$$

This takes into account the equal prior probabilities and that the pooled within groups covariance matrix of the discriminant functions is an identity matrix. Each case is classified into the class for which $P(k|\underline{d})$ is highest. The calculation actually used by SPSS is

$$p_k = \begin{cases} 0 & g_k - g_{max} \leq -46 \\ \dfrac{\exp(g_k - g_{max})}{\sum_{i=1}^{K} \exp(g_i - g_{max})} & \text{otherwise} \end{cases}$$

where

$$g_k = \log \frac{1}{K} - \frac{1}{2} x_k$$

$$g_{max} = \max_k g_k$$

REFERENCES

[1] A.L. Zobrist and W.B. Thompson, "Building a Distance Function for Gestalt Grouping," IEEE Trans. on Computers, vol. C-24, pp. 718-728, July 1975.

[2] B. Julesz, "Texture and Visual Perception," Scientific American, vol. 212, pp. 38-54, Feb. 1965.

[3] B.J. Schachter, A. Rosenfeld, and L.S. Davis, "Random Mosaic Models for Textures," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-8, pp. 694-702, Sep. 1978.

[4] A. Rosenfeld and L.S. Davis, "Image Segmentation and Image Models," Proc. IEEE, vol. 67, pp. 764-772, May 1979.

[5] B. Julesz, "Visual Pattern Discrimination," IRE Trans. on Info. Theory, vol. IT-8, pp. 84-92, Feb. 1962.

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

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

[8] W.K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Fields," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-8, pp. 796-804, Nov. 1978.

[9] H. Tamura, S. Mori, and T. Yamawaki, "Textural Features Corresponding to Visual Perception," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-8, pp. 460-473, June 1978.

[10] J.K. Hawkins, "Textural Properties for Pattern Recognition," in B.S. Lipkin and A. Rosenfeld (Eds.), Picture Processing and Psychopictorics. New York: Academic Press, 1970, pp. 347-370.

[11] R.M. Haralick, "Statistical and Structural Approaches to Texture," Proc. IEEE, vol. 67, pp. 786-804, May 1979.

[12] W.K. Pratt, Digital Image Processing. New York: Wiley-Interscience, 1978.

[13] J.R. Kender, "Instabilities in Color Transformations," in Proc. Conf. on Pattern Recognition and Image Processing, Troy, NY, June 1977, pp. 266-274.

[14] E.H. Land, "The Retinex Theory of Color Vision," Scientific American, vol. 237, pp. 108-128, Dec. 1977.

[15] H. Kaizer, A Quantification of Textures on Aerial Photographs. Note 121, AD 69484, Boston Univ. Research Labs., Boston, 1955.

[16] B.H. McCormick and S.N. Jayaramamurthy, "Time Series

Model for Texture Synthesis," Int. J. of Computer and Info. Sciences, vol. 3, pp. 329-343, Dec. 1974.

[17] J.T. Tou, D.B. Kao, and Y.S. Chang, "Pictorial Texture Analysis and Synthesis," in Proc. 3rd Int. Jnt. Conf. on Pattern Recognition, Coronado, Calif., Nov. 1976, pp. 590-590P.

[18] K. Deguchi and I. Morishita, "Texture Characterization and Texture-Based Image Partitioning Using Two-Dimensional Linear Estimation Techniques," IEEE Trans. on Computers, vol. C-27, pp. 739-745, Aug. 1978.

[19] W. Muller and W. Herman, "Texture Analyzer System," Industrial Research, pp. 49-54, Nov. 1974.

[20] R. Bajcsy and L. Lieberman, "Texture Gradient as a Depth Cue," Computer Graphics and Image Processing, vol. 5, pp. 52-67, 1976.

[21] L. Kirvida and G. Johnson, "Automatic interpretation of ERTS data for Forest Management," in Symp. on Significant Results Obtained from Earth Res. Technol. Satellite, NASA, Mar. 1973.

[22] G.G. Lendaris and G.L. Stanley, "Diffraction-Pattern Sampling for Automatic Pattern Recognition," Proc. IEEE, vol. 58, pp. 198-216, Feb. 1970.

[23] R. Bajcsy and L.I. Lieberman, "Computer Description of Real Outdoor Scenes," in Proc. 2nd Int. Jnt. Conf. on Pattern Recognition, Copenhagen, Aug. 1974, pp. 174-179.

[24] N. Gramenopoulos, "Terrain Type Recognition Using ERTS-1 MSS Images," in Symp. on Significant Results Obtained from the Earth Resources Technology Satellite, NASA SP-327, Mar. 1973, pp. 1229-1241.

[25] R.J. Horning and J.A. Smith, "Application of Fourier Analysis to Multispectral/Spatial Recognition," in Management and Utilization of Remote Sensing Data ASP Symposium, Sioux Falls, SD, Oct. 1973.

[26] H. Maurer, "Texture Analysis with Fourier Series," in Proc. 9th Int. Symp. on Remote Sensing of the Environment, Environmental Research Inst. of Michigan, Ann Arbor, Michigan, Apr. 1974, pp. 1411-1420.

[27] S.L. Tanimoto, "An Optimal Algorithm for Computing Fourier Texture Descriptors," IEEE Trans. on Computers, vol. C-27, pp. 81-84, Jan. 1978.

[28] B.F. Green, A.K. Wolfe, and B.W. White, "The Detection of Statistically Defined Patterns in a Matrix of Dots," Am. J. Psychology, vol. 72, pp. 503-520, 1959.

[29] S.R. Purks and W. Richards, "Visual texture discrimination using random dot patterns," J. Opt. Soc. Am., vol. 67, pp. 765-771, June 1977.

[30] M. Hassner and J. Sklansky, "Markov Random Fields as Models of Digitized Image Texture," in Proc. Conf. on Pattern Recognition and Image Processing, Chicago, May 1978, pp. 346-351.

[31] E.M. Darling and R.D. Joseph, "Pattern Recognition from Satellite Altitudes," IEEE Trans. on Systems Science and Cybern., vol. SSC-4, pp. 38-47, Mar. 1968.

[32] A. Rosenfeld and E.B. Troy, "Visual Texture Analysis," in IEEE Conf. on Feature Extraction and Selection in Pattern Recognition, Argonne, IL, Oct. 1970, pp. 115-124.

[33] R.M. Haralick, "A Texture-Context Feature Extraction Algorithm for Remotely Sensed Imagery," in Proc. IEEE Decision and Control Conf., Gainsville, FL, Dec. 1971, pp. 650-657.

[34] R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-3, pp. 610-621, Nov. 1973.

[35] R.M. Haralick and K. Shanmugan, "Computer Classification of Reservoir Sandstones," IEEE Trans. on Geoscience Electronics, vol. GE-11, pp. 171-177, Oct. 1973.

[36] R.M. Haralick and K.S. Shanmugam, "Combined Spectral and Spatial Processing of ERTS Imagery Data," J. of Remote Sensing of the Environment, vol. 3, pp. 3-13, 1974.

[37] Y.P. Chien and K.S. Fu, "Recognition of X-Ray Picture Patterns," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-4, pp. 145-156, Mar. 1974.

[38] N.J. Pressman, "Optical Texture Measurement of Human Cells by Markovian Analysis," in Proc. 4th New England Bioengineering Conf., New Haven, Conn., May 1976, pp. 361-364.

[39] P. Chen and T. Pavlidis, Segmentation by Texture Using a Cooccurrence Matrix and a Split-and-Merge Algorithm. TR-237, Princeton Univ., Princeton, NJ, Jan. 1978.

[40] J.S. Weszka, C.R. Dyer and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-6, pp. 269-285, Apr. 1976.

[41] R.W. Conners, Some Theory on Statistical Models for Texture and Its Application to Radiographic Image Processing. Ph.D. Thesis, College of Engineering, Univ. of Missouri - Columbia, Dec. 1976.

[42] R.W. Conners and C.A. Harlow, "Some Theoretical Considerations Concerning Texture Analysis of Radiographic Images," in Proc. IEEE Conf. on

Decision and Control, Clearwater, Fla., Dec. 1976, pp. 162-167.

[43] A. Rosenfeld and L.S. Davis, "Iterative Histogram Modification," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-8, pp. 300-302, Apr. 1978.

[44] J.T. Tou and Y.S. Chang, "Picture Understanding by Machine via Textural Feature Extraction," in Proc. IEEE Conf. on Pattern Recognition and Image Processing, Troy, NY, June 1977, pp. 392-399.

[45] R.M. Haralick, "A Resolution Preserving Textural Transform for Images," in Proc. Conf. on Computer Graphics, Pattern Recognition, and Data Structures, Los Angeles, May 1975, pp. 51-61.

[46] R. Ohlander, K. Price and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," Computer Graphics and Image Processing, vol. 8, pp. 313-333, Dec. 1978.

[47] G. Winkler and K. Vattrodt, "Measures for Conspicuousness," Computer Graphics and Image Processing, vol. 8, pp. 355-368, Dec. 1978.

[48] S.Y. Lu and K.S. Fu, "Stochastic Tree Grammar Inference for Texture Synthesis and Discrimination," in Proc. Conf. on Pattern Recognition and Image Processing, Chicago, May 1978, pp. 340-345.

[49] S.Y. Lu and K.S. Fu, "A Syntactic Approach to Texture Analysis," Computer Graphics and Image Processing, vol. 7, pp. 303-330, June 1978.

[50] G.J. VanderBrug and A. Rosenfeld, "Two-Stage Template Matching," IEEE Trans. on Computers, vol. C-26, pp. 384-393, Apr. 1977.

[51] A.L. Gilchrist, "The Perception of Surface Blacks and Whites," Scientific American, vol. 240, pp. 112-124, Mar. 1979.

[52] S.W. Zucker, On the Foundations of Texture: A Transformational Approach. TR-331, Univ. of Maryland, College Park, MD, Sep. 1974.

[53] J.T. Maleson, C.M. Brown, and J.A. Feldman, "Understanding Natural Texture," in Proc. Image Understanding Wkshp., DARPA, Palo Alto, Calif., Oct. 1977, pp. 19-27.

[54] M.M. Galloway, "Texture Analysis using Gray Level Run Lengths," Computer Graphics and Image Processing, vol. 4, pp. 172-179, June 1975.

[55] O.R. Mitchell, C.R. Myers, and W. Boyne, "A Max-Min Measure for Image Texture Analysis," IEEE Trans. on Computers, vol. C-26, pp. 408-414, Apr. 1977.

[56] O.R. Mitchell and S.G. Carlton, "Image Segmentation Using a Local Extrema Texture Measure," Pattern Recognition, vol. 10, pp. 205-210, 1978.

[57] R.W. Ehrich and J.P. Foith, "Representation of

Random Waveforms by Relational Trees," IEEE Trans. on Computers, vol. C-25, pp. 725-736, July 1976.

[58] L.S. Davis, S. Johns, and J.K. Aggarwal, "Texture Analysis Using Generalized Co-Occurrence Matrices," IEEE Trans. on Pattern Anal. and Machine Intelligence, vol. PAMI-1, pp. 251-259, July 1979.

[59] J.S. Read and S.N. Jayaramamurthy, "Automatic Generation of Texture Feature Detectors," IEEE Trans. on Computers, vol. C-21, pp. 803-812, July 1972.

[60] B.H. McCormick and S.N. Jayaramamurthy, "A Decision Theory Method for the Analysis of Texture," Int. J. of Computer and Info. Sciences, vol. 4, pp. 1-38, Mar. 1975.

[61] N. Ahuja, L.S. Davis, R.M. Haralick, and D.P. Panda, Image Segmentation Based on Local Gray Level Patterns. TR-551, Univ. of Maryland, College Park, MD, June 1977.

[62] P. Brodatz, Textures: A Photographic Album for Artists and Designers. New York: Dover, 1966.

[63] O.D. Faugeras and W.K. Pratt, Stochastic-based Visual Texture Feature Extraction. (Unpublished.)

[64] B. Ashjari and W.K. Pratt, Supervised Classification with Singular Value Decomposition Texture Measurement. USCIPI Report 860, Image Processing Inst., Univ. of Southern Calif., Los Angeles, Mar. 1979.

[65] E.L. Hall, R.P. Kruger, S.J. Dwyer III, D.L. Hall, R.W. McLaren, and G.S. Lodwick, "A Survey of Preprocessing and Feature Extraction Techniques for Radiographic Images," IEEE Trans. on Computers, vol. C-20, pp. 1032-1044, Sep. 1971.

[66] E.L. Hall, "Almost Uniform Distributions for Computer Image Enhancement," IEEE Trans. on Computers, vol. C-23, pp. 207-208, Feb. 1974.

[67] J. Sklansky, "Image Segmentation and Feature Extraction," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-8, pp. 237-247, Apr. 1978.

[68] W. Frei, "Image Enhancement by Histogram Hyperbolization," Computer Graphics and Image Processing, vol. 6, pp. 286-294, June 1977.

[69] R.W. Conners and C.A. Harlow, "Equal Probability Quantizing and Texture Analysis of Radiographic Images," Computer Graphics and Image Processing, vol. 8, pp. 447-463, Dec. 1978.

[70] S.-K. Chang and Y.-W. Wong, "Optimal Histogram Matching by Monotone Gray Level Transformation," C. ACM, vol. 21, pp. 835-840, Oct. 1978.

[71] I.E. Abdou and W.K. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge

Detectors," Proc. IEEE, vol. 67, pp. 753-763, May 1979.

[72] E.E. Triendl, "Automatic Terrain Mapping by Texture Recognition," in Proc. 8th Int. Symp. on Remote Sensing of Environment, Environmental Research Inst. of Michigan, Ann Arbor, Michigan, Oct. 1972.

[73] S. Hsu, "A Texture-Tone Analysis for Automated Landuse Mapping with Panchromatic Images," Proc. of the Amer. Soc. for Photogrammetry, pp. 203-215, Mar. 1977.

[74] D.H. Hubel and T.N. Wiesel, "Brain Mechanisms of Vision," Scientific American, vol. 241, pp. 150-162, Sep. 1979.

[75] J. Toriwaki and T. Fukumura, "Extraction of Structural Information from Grey Pictures," Computer Graphics and Image Processing, vol. 7, pp. 30-51, Feb. 1978.

[76] T. Peucker and D. Douglas, "Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data," Computer Graphics and Image Processing, vol. 4, pp. 375-387, Dec. 1975.

[77] K.I. Laws, A Proposed Class of Picture Operators. USCIPI Report 840, Image Processing Inst., Univ. of Southern Calif., Los Angeles, Sep. 1978.

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

[79] K.C. Hayes, Jr., A.N. Shah, and A. Rosenfeld, "Texture Coarseness: Further Experiments," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-4, pp. 467-472, Sep. 1974.

[80] M.H. Hueckel, "A Local Visual Operator which Recognizes Edges and Lines," J. ACM, vol. 20, pp. 634-647, Oct. 1973.

[81] M.K. Hu, "Visual Pattern Recognition by Moment Invariants," IRE Trans. on Info. Theory, vol. IT-8, pp. 179-187, Feb. 1962.

[82] S. Maitra, "Moment Invariants," Proc. IEEE, vol. 67, pp. 697-699, Apr. 1979.

[83] A. Rosenfeld, R.A. Hummel and S.W. Zucker, "Scene Labeling by Relaxation Operations," IEEE Trans. on Syst., Man, and Cybern., vol. SMC-6, pp. 420-433, June 1976.

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

[85] M.J. Norusis, SPSS Statistical Algorithms Release 8.0. Chicago: SPSS Inc., 1979.