12-1-1987

Texture Measures ofSpatial Patterns on Thematic Mapper Imagery: An Experiment.

Li Bin
University of Nebraska at Omaha

Texture Measures of Spatial Patterns on Thematic Mapper Imagery: An Experiment.

A Thesis
Presented to the

Department of Geography/Geology
and the
Faculty of the Graduate College
University of Nebraska

In Partial Fulfillment
of the Requirements for the Degree

Master of Arts

University of Nebraska at Omaha

By

Li Bin

December, 1987

Thesis Acceptance

Accepted for the faculty of the Graduate College, University of Nebraska, in partial fulfillment of the requirements for the degree Master of Arts, University of Nebraska at Omaha.

Committee

Name	Department
Dr. Michael P. Peterson	Geography/Geology
Dr. David A. DiMartino	Geography/Geology
Peter T. Suzuki	Public Admin./Urban Studies

Dr. Michael P. Peterson
Chairman

Dec. 1, 1987
Date

Abstract

The digital format of remote sensing data facilitates the measurement of spatial patterns. The concept of measurability of spatial patterns has important geographic implications and may open up an alternative applications for satellite remote sensing of urban areas. Texture analysis, a set of techniques developed in pattern recognition, is found to be useful in measuring spatial pattern on digital imagery. Two approaches of texture analysis are selected. One is Haralick's Spatial Dependence Matrix, the other is Jernigan's, et. al., Entropy-based texture measures. They perform in spatial domain and frequency domain respectively. Ten subimage areas in Omaha suburb are selected from a Landsat TM image. The subimage areas includes the major residential spatial patterns in the area. Through analysis, it is found that residential areas with

different spatial features do present distinguishable texture measures, in both SPADEP and Entropy-based texture analysis. With the introduction of texture analysis, a new set of terminology can be used to describe a spatial pattern and may greatly enhance our concepts of certain spatial phenomena. Potential application of texture analysis in this context could be in urban land use mapping, computer-assisted land use monitoring and comparative study in urban spatial patterns.

Acknowledgements

I would like to thank the members of my thesis committee: Dr. Michael Peterson and Dr. David Dimartino of the Geography Department, and Dr. Peter Suzuki of the Urban Studies Department. Their expertise and efforts were imperative and sincerely appreciated.

Special thanks are given to Dr. Peterson, for his advise, encouragement, and for his patience in reading and correcting my English writing.

I also appreciate the helps of Mark Laustrup, the manager of the Remote Sensing Application Laboratory, who provided the imagery data and unlimited computing time.

Additionlly, I would like to thank Dr. Joseph Wood, the former chair of the graduate committee, who helped me to start the graduate program here.

Finally, I would like to thank the International Studies Program at UNO. Without their initial support, I would not be able to come to this country and have such wonderful experiences and excellent education at this university.

Contents

Chapter 1 Analysis

Chapter 2 Methodology

List of Figures

Chapter 1: Introduction

I. INTRODUCTION

Satellite imagery has been used with great success in the study of the atmosphere, lithosphere, water bodies, vegetation and soil. However, its application to urban areas is limited largely to broad-scale land use mapping, and even these results are somewhat disappointing. This is due to the low spatial resolution of satellite imagery and an orientation in remote sensing toward spectral analysis. The improvements of the sensor system and the processing techniques may eventually overcome the obstacle of spatial resolution; meanwhile, extending our concepts of remote sensing beyond spectral analysis could also open up alternative applications of satellite remote sensing to the study of urban areas.

Satellite data such as Landsat imagery is recorded in digital form by a series of electromagnetic sensors. To most people, this is simply a prerequisite for digital image processing and image enhancement. The geographic implications of digital imagery, especially to the study of urban areas, are largely overlooked.

A city may be composed of various land use types, such as residential, commercial, industrial etc. Each of these areas contains certain physical elements, i.e., street, building, open space (paved or vegetation covered) etc. A land use pattern is a certain combination of these elements, which is different from one to another both quantitatively and qualitatively. Even within the same type of land use, the spatial combination can be different. A new and an old residential area may have a similar composition of physical elements, i.e., they both have houses, streets, lawns and trees. An older residential area, however, may have grid street layout and houses more closely together while in the new residential area, streets are curved, trees are small and houses are further apart. When these patterns are represented in a digital image of proper scale, they become associated with reflectance patterns which can be described by the spatial relationship among reflectance levels (gray levels). In another words, the spatial characteristics of a land use pattern become measurable in a digital image. Although the spatial characteristics thus obtained are only the physical features of certain land use patterns at a certain scale, the notion of the measurability of spatial patterns in digital imagery is worthy of consideration.

In urban geography, it is a tradition to study the physical form in a city as a device to reveal the social and economic processes. In fact, major geographic models on urban growth and intra-urban structures are morphological with the concentric zone (Burgess), sector (Hoyt), and multi-nuclei models (Harris and Ullman) being often cited. When a city is represented in a Landsat image, the identity of most individual morphological elements, such as houses, expressed by their shapes, will be lost, but the spectral characteristics of these elements and the spatial relationships among them are generalized in the reflectance patterns. Furthermore, such relationships become measurable and may enhance some of our conceptions of urban geography.

Although measuring characteristics of spatial patterns in digital imagery may be a novel idea, a similar notion, texture analysis, has long been recognized in the field of pattern recognition. Texture analysis, though definitions and methods vary, is fundamentally a way to measure the spatial relationships among gray levels. Numerous mathematical models have been developed to implement texture analysis. The primary objective of such studies is to find distinguishable texture measures for the sake of automated pattern recognition and classification. Many of these models are designed to simulate pattern recognition by human beings, and are not necessarily relevant to the spatial characteristics with which we are concerned. Nevertheless, it is found that some texture measures, such as Haralick's (1976) Spatial Dependence Matrix approach and Jernigan's, et. al. (1983) Entropy-based texture measure in the frequency domain, tend to reveal the overall characteristics of a spatial pattern and contain certain geographical implications.

The theoretical assumption of this thesis is that different spatial patterns on the earth surface are measurable in digital imagery. Further, texture analysis, a set of techniques developed in pattern recognition, may be used to acquire such measurements. The measurability of spatial pattern in a digital image may greatly enhance our understanding of certain spatial organizations. It is the objective of this thesis to measure spatial patterns on digital imagery and in particular, to seek possible applications to the analysis of urban areas.

To evaluate the potential of the concept of measurability of spatial pattern in digital imagery and techniques of texture analysis, urban residential patterns are chosen for study. Residential areas constructed at different times, based on different street systems, resided in by different groups of people, situated in different locations will exhibit different spatial characteristics. Would such characteristics be revealed by certain texture measures from Landsat digital imagery? Would similar residential patterns have similar measures? What are the geographic implications of these measurements?

Landsat Thematic Mapper (TM) imagery, is used in this project. The higher spatial resolution of TM imagery (30 meters per pixel) may better represent the spectral reflectance of basic physical elements in an urban scene. Texture analysis has been largely applied on MSS data (79 meters per pixel). Selecting TM imagery in this study can be viewed as another investigation of its potential application to urban areas.

The following section provides a comprehensive review of texture analysis and spatial pattern recognition.

II. TEXTURE ANALYSIS AND SPATIAL PATTERN RECOGNITION

A. The Definition of Texture

In general, texture in an image refers to the spatial relationships of reflectance levels, often expressed as gray tones. Texture has generally been defined through an enumeration of characteristics such as fine, coarse, regular, irregular, etc., in order to facilitate the implementation of corresponding quantitative measures. However, it is found that a precise definition of texture does not exist. In his review article on texture analysis, Haralick (1979) proposed the *tone-texture concept* which is the further development of his early concept of *discrete tonal feature* (1973). According to the tone-texture concept, gray tone and texture are not independent. The relationship between tone and texture is inextricable. Tonal primitive has been defined as a maxima connected set of pixels having a given tonal property (Haralick, 1979). In order to characterize texture, one must characterize the tonal primitive properties as well as the spatial interrelationships among them due to the inextricable relationships between tone and texture; thus, texture analysis, indeed, is two dimensional. However, Haralick pointed out, the existing approaches tend to emphasize one or the other aspect and do not treat each equally.

B. Major methods of textural analysis

Computer-aided texture analysis has been studied since 1960 (Weszka, et. al. 1976). Major methods have been developed in both spatial and frequency domains. To give a general view of texture analysis, Haralick published an article in 1979 reviewing typical statistical and structural approaches to texture analysis. Since then, new approaches have been emerging but it seems that they are still confined to the classic methods. Therefore, Haralick's classification of texture approaches is still useful for purposes of review.

According to Haralick, approaches to texture analysis of imagery can be divided into two categories, statistical and structural. There had been eight statistical approaches to the measurement and characterization of image texture: autocorrelation functions, optical transform, digital transforms, textural edgeness, structural elements, spatial gray tone coocurrence probabilities, gray tone run lengths, and autoregressive models. As indicated before, each of these techniques tends to emphasize one or other aspects of the texture feature.

The first three of these approaches are performed in the frequency domain. It is well known that specific components in the spatial frequencies domain representation of an image contain explicit information about the spatial distribution. Autocorrelation functions, i.e., the Fourier transform of the power spectrum, is a measure of the linear dependence between gray levels. It tends to reveal the properties of tonal primitives (especially their sizes). The faster the autocorrelation function drops off with distance, the smaller size of tonal primitives is indicated, the finer the texture. Pioneer work was done by Kaizer with seven aerial photographs of an Arctic region (1955). Since then, the autocorrelation approach has been seldom used. More recent texture analyses in frequency domain has focused on the Fourier power spectrum. Early experiments were optical processing, measuring the light distribution of the Fraunhofer diffraction pattern (the optical equivalence of the Fourier power spectrum). The experiments done by Lendaris and Stanley (1969) present an example of the most popular texture measures of this kind. The pattern vectors they used are the average energy in annular rings and in 9 degree wedges of the diffraction pattern respectively. Aerial photographs were used to test the power of such methods in distinguishing man-made from non-man-made features and the subclasses of man-made features. Ninety percent of identification accuracy was reported. In general, summed energy measures in the Fourier power spectrum is the major method used in texture analysis.

Although successful texture extraction (usually over 90 percent accuracy of classification) has been found in many projects with various remotely sensed data (Egbert, et. al., Gramenopoulos, Horning and Smith, Kirvida and Jonhson, Maurer, Bajcsy and Lieberman, etc.), texture analysis in the frequency domain has met criticism in that this approach only reveals the global information from across the complete image and neglects important local discrimination information about the texture. It is also found that texture measures in the frequency domain are not invariant with size, orientation and even with monotonic gray level transformation (Haralick, 1979). In a comparative study of texture measures for terrain classification, Weszka, et. al., concluded that measures in the

the spatial domain (Weszka, et. al., 1976). In practice, Rosenfield, et. al., consistently found that the frequency approach is less successful than the other approaches (Rosenfield, et. al., from 1981 to 1982).

Nevertheless, research on texture measures in the frequency domain did not end. Recently, M. E. Jernigan and F. D'astous developed an approach of entropy-based texture analysis. They used the regional entropy measures in the spatial frequency domain which would provide texture discriminating information independent of information contained in the usual summed energy within frequency domain features. The measure is size invariant and comparable to that of gray level coocurrence contrast feature.

Besides viewing texture as spatial frequency distribution, Rosenfield, et. al., found that texture can be also measured in terms of edgeness per unit area (Rosenfield and Troy, 1970, Rosenfield and Thurston, 1971). Coarse textures have a small number of edges per unit area while fine textures have a high number of edges per unit area. Further experiments have been carried out by Sutton and Hall (1972), Hsu (1977).

The structural element approach is proposed by J. Serra(1974), and G. Matheron (1967). They use a matching procedure to detect the spatial regularity of shapes called structured elements in a binary image. This measure emphasizes the shape aspects of the tonal primitives but can only do so for binary images.

Another major second order statistical measure of texture is called the Gray Level Spatial Dependence approach. It emphasizes the spatial distribution and spatial dependence among the gray tones in a local area. B. Julesz (1962) first used gray tone spatial dependence coocurrence statistics in texture discrimination experiments. E. M. Darling and R. D. Joseph (1968) first used this approach in identifying cloud types in satellite imagery. Bartels et. al., (1969) used one dimensional coocurrence in a medical application. Rosenfield and Troy (1970) and Haralick (1971) suggested two dimensional spatial dependence of gray tones in a coocurrence matrix for each fixed distance and/or angular spatial relationship; Haralick et. al., used statistics of this matrix as measures of texture in satellite imagery, aerial, and microscopic imagery (1973, 1972). Chien and Fu (1974) showed the application of gray tone coocurrence to computer-assisted chest X-ray analysis. Pressman (1972) applied the similar techniques to cervical cell discrimination. Chen and Pavlidis (1978) used coocurrence in conjunction with split and merge procedure to segment an image on the basis of texture. More recently, Jensen (1979) and Jensen and Toll (1982) reported on the use of Haralick's angular second moment (ASM) as an additional feature in the supervised classification of Landsat MSS imagery at the urban fringe and in urban land use change-detection mapping. All of these studies achieved reasonable results on different textures using gray tone coocurrence. In their comparative studies of textural measures, Weszka, et. al., (1976) found that the coocurrence approach was among the best so far. The study by Conners and Harlow (1976) theoretically concluded that Haralick's gray-tone coocurrence matrices had the best innate discriminative ability. The power of the coocurrence approach is that it characterizes the spatial interrelationship of the gray tones in a textural pattern and is invariant under monotonic gray level transformation.

Further development of this idea by Sun and Wee (1983) resulted in a new texture transformation called Neighborhood Gray Level Dependence Matrix (NGLDM) approach. It is said to be essentially invariant even under spatial rotation.

The gray level run length approach represents a family of first order statistical texture measurements in the spatial domain. It characterizes coarse texture as having many pixels in a constant gray run and fine texture as having few pixels in a constant gray tone run. The study by Hsu (1978) found, among 17 proposed first-order texture measures to classify level I land cover from digital aerial photography, gray level run length statistics were superior. Further experiments were reported by Irons and Peterson (1981) and Shih and Showengerdt (1983).

The autoregressive models are a way of revealing the linear dependence one gray level has on another. It was introduced by McCormick and Jayaramamurthy (1974) and experimented by Deguchi and Morishila (1976), Tou et. al., (1976), and Tou and Chang (1976). Theoretically, the autoregressive approach is sufficient to capture everything about a texture, however, the textures it can characterize are likely to consist of microtextures.

Besides the eight statistical approaches reviewed above, there is another set of texture measures of structural approaches. Pure structural models of texture are based on the view that texture are made up of primitives which appear in near regular repetitive spatial arrangements. It is a much more complex approach and is not used widely.

Recently, the idea of using fractal analysis to extract spatial features in Landsat imagery has been reported (Goodchild and Mark, 1987). The application of fractal geometry would introduce a whole set of texture measures in the fractional dimensions. Further experiments of fractal analysis are being conducted in the field of pattern recognition.

From the literature review, we see that no effort has been devoted to textural analysis with TM imagery and that most studies are purely technical experiments for pattern recognition or classification. Few applications of texture analysis have been found in urban study and geographic inquiry.

III. STUDY AREA, DATA, METHODOLOGY, AND THESIS ARRANGEMENT

The city of Omaha is selected as the study area. The major emphasis is placed on the urban fringe; the most active area in the city in terms of change. Landsat Thematic Mapper Digital data acquired on June 12, 1985 will constitute the primary data source. Other data sources used to assist in the study are: two Landsat MSS images of Omaha taken in 1976 and 1978, aerial photos, updated land use maps, and the USGS topographic sheets.

For comparative purposes, two approaches of texture analysis will be used. One is Haralick's gray level spatial dependence matrix approach in the spatial domain, the other is Jernigan's, et. al., entropy-based texture analysis in the frequency domain. Since these two approaches have not been implemented in the current image processing system at the Remote Sensing Application Laboratory, considerable amount of effort is needed to be devoted to computer programming. The Eye-com Spatial Data system and PDP-11 mini-computer will serve as the major computer facilities used in the project.

The thesis is divided into four chapters. The following two chapters will discuss the methodology and the process of analysis. The conclusions from this study will be drawn in chapter four.

Chapter 2: Methodology

In this chapter, two approaches of textural analysis, the Spatial Dependence (SPADEP) and the Entropy-Based Texture (EBT) approaches and their computer implementation will be discussed.

I. Spatial Dependence Matrix Approach

A. The Principle

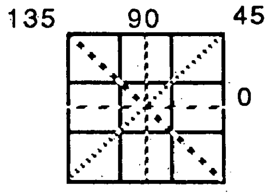
This approach is based on the assumption that textual information on an image is contained in the overall spatial relationships among gray levels and that such relationships can be expressed by the measurement of the coocurrence of one gray level to another in different directions and distance within a limited space, such as 3 X 3 subimage. The mathematical expression of the coocurrence frequency is the Spatial Dependence Matrix (SPADEP).

Consider the following example. Suppose Fig. 2-1a is the image to be measured. If the range of the gray level is from 0 to 3, the possible coocurrence relationships of the four gray levels can be expressed as Fig. 2-1b. Notice that this matrix is symmetrical.

$$\begin{array}{ccc} 0 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 3 & 3 \end{array} \qquad \begin{array}{cccc} (0,0) & (0,1) & (0,2) & (0,3) \\ (1,0) & (1,1) & (1,2) & (1,3) \\ (2,0) & (2,1) & (2,2) & (2,3) \\ (3,0) & (3,1) & (3,2) & (3,3) \end{array}$$

a b

Fig. 2-1 A 3 X 3 image and the general form of the SPADEP (N=4).



$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 2 & 0 & 1 \\ 1 & 2 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 & 0 \\ 3 & 1 & 0 & 0 & 2 \end{array}$$

a b

Fig. 2-2 Neighboring relations (a) and the SPADEP in the 0 degree direction (b).

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 4 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 \\ 2 & 0 & 0 & 2 & 1 \\ 3 & 0 & 1 & 1 & 0 \end{array} \qquad \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 \\ 3 & 0 & 0 & 1 & 0 \end{array} \qquad \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 1 & 1 & 0 & 0 \end{array}$$

Fig. 2-3 SPADEP in the 90, 45, and 135 degree directions.

Now, let's compute the SPADEP matrices in four directions with neighboring gray level and distance all equal to 1 (Fig. 2-2). Fig2-2b depicts the 0 degree coocurrences for the 3 X 3 matrix in Figure 1. The 0 entry at (0,0) indicate no 0-0 coocurrences while the 2 entry at (0,1) indicates two 0-1 coocurrences at this angle. Using the same method, we can construct the SPADEP matrices in 90 , 45 and 135 degree directions (Fig. 2-3). For the purposes of explanation, we define such matrices as P(i,j).

To obtain the normalized frequency, i.e., the relative probability of these matrices, each entry in P(i,j) is divided by the number of nearest neighboring pixel pairs. For an image of N X M, the normalizing factors R are:

$$R_1 = 2N(M-1);$$

$$R_2 = 2M(N-1);$$

$$R_3 = 2(M-1)(N-1);$$

$$R_4 = 2(N-1)(M-1);$$

where R_1, R_2, R_3, R_4 represent the R in the four directions: 0, 90, 45, 135. Fig. 2-4 shows the normalized P(i,j). Notice that the sum of rows and the sum of columns are both equal to 1.

0.00	0.17	0.00	0.08
0.17	0.00	0.17	0.00
0.00	0.17	0.00	0.00
0.08	0.00	0.00	0.17

0 degrees

0.03	0.00	0.00	0.00
0.00	0.17	0.00	0.08
0.00	0.17	0.17	0.08
0.00	0.08	0.08	0.00

90 degrees

0.00	0.25	0.00	0.00
0.25	0.00	0.12	0.00
0.00	0.12	0.00	0.12
0.00	0.00	0.12	0.00

45 degrees

0.00	0.12	0.00	0.12
0.12	0.00	0.12	0.12
0.00	0.12	0.00	0.00
0.12	0.12	0.00	0.00

135 degrees

Fig. 2-4 The normalized SPADEP

After constructing the normalized P(i,j), we can now apply statistical models to extract textural information from these matrices. Haralick, et. al., proposed fourteen statistical models for texture extraction from the SPADEP. Each of these measures tends to emphasize certain aspects of textural properties in an image, e.g., homogeneity, complexity, linear structure, contrast, number and nature of boundaries present, etc. Among the 14 statistical measures, some are relatively difficult to interpret. No detailed explanations on these models have been found. Since it is important to know what textural information each of these features expresses, a pre-study of these models is included in the following sections.

B. Texture Measures from SPADEP

A theoretical explanation of the statistical texture models is out of the scope of this thesis. Only a brief discussion will be presented with examples. We give the following notations that are used for the texture models:

N: number of gray levels in the image;

P(i,j): (i,j)th entry in a normalized SPADEP;

Px(i), Py(j): ith/or jth entry in the marginal probability matrix obtained by summing the rows/or columns of P(i,j);

$Px+y(K) = \Sigma P(i,j)$, for $i + j = K = 2, 3, 4, \ldots\ldots, 2N$; (Fig. 2-5a);

$$P_{x-y}(K) = \Sigma\ P(i,j), \text{ for } |i - j| = K = 0,\ 1,\ 2,\ \ldots\ldots,\ N\text{-}1. \quad \text{(Fig. 2-5b).}$$

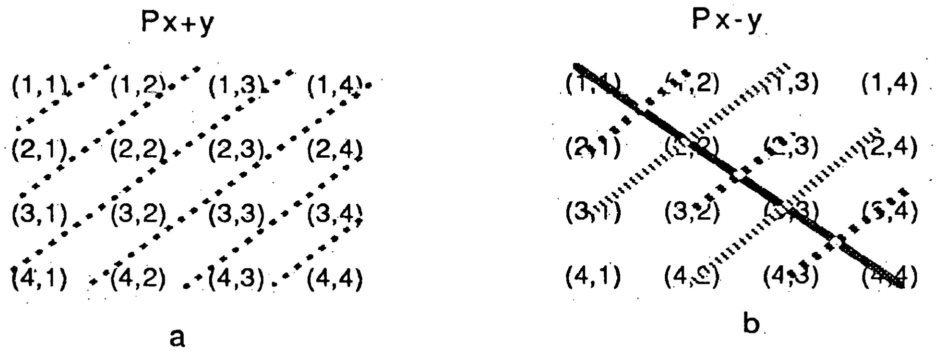


Fig. 2-5 Illustration of Px+y(i) and Px-y(i).

Px+y(K) is a matrix representing the sums along the right diagonal of P(i,j) (Fig. 2-5). Px-y(K) is a matrix obtained by summing each group of elements with subscripts i and j and |i-j| = 0, 1, 2,......N-1. For example, elements along the dark line in the matrix of Figure 2-5b., |i-j| = 0.

1) **Angular Second Moment**

$$ASM = \Sigma\ P(i,j)^2.$$

ASM is one of the most frequently used SPADEP measures. In general it measures the homogeniety of the image. Since P(i,j) ranges from 0 to 1, the more widely distributed of P(i,j) the smaller the ASM, indicating less homogeneity in the image (Fig. 2-6a). ASM itself ranges from 0 to 1. In an image presenting only one gray level, all but one element in P(i,j) equal to 0, ASM reaches the maximum value 1 (Fig. 2-6b). ASM is invariant under monotonic gray level transformation.

$\begin{matrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{matrix}$	$\begin{matrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{matrix}$		ASM1 = 1.0000	(a)
N=4	The SPADEP in horizontal direction			

$\begin{matrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{matrix}$	ASM1 = 0.0833
	ASM2 = 0.0833
	ASM3 = 0.1250
	ASM4 = 0.1250
		$\begin{matrix} 8 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{matrix}$	ASM1 = 1.0000	(b)

N=9	Measures in four directions	N=9	Horizontal measure	

Fig. 2-6 Maximum minimum case of ASM.

2) **Contrast**

$$CON = \Sigma\ [n^2 P(i,j)] \quad \text{for } |i - j| = n = 0, 1, \ldots\ldots, N-1.$$

This is essentially the moment of inertial of the P(i,j) around its main diagonal. It is a natural measure of the degree of spread of the matrix value, i.e., the contrast or the amount of local variation present in an image. The higher the value of CON, the higher the contrast; $0 \leq CON \leq (N-1)^2$. Fig. 2-7 is an example of contrast measurement. Notice that CON is not independent of gray level and the measure of CON will be changed under monotonic gray level transformation.

N=4

$$\begin{matrix}3 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 3\end{matrix}$$

The SPADEP at 135 degree.

$$\begin{matrix}0.50 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.50\end{matrix}$$

Measures in four directions.

CON1 = 6.0000
CON2 = 6.0000
CON3 = 4.5000
CON4 = 0.0000

N=5

$$\begin{matrix}4 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 4\end{matrix}$$

$$\begin{matrix}0.50 & 0.00 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.00 & 0.50\end{matrix}$$

CON1 = 10.6667
CON2 = 10.6667
CON3 = 8.0000
CON4 = 0.0000

Fig. 2-7 Examples of Contrast measurement.

3) **Correlation**

$$COR = \{\Sigma\ [ijP(i,j)] - UxUy\} / SxSy;$$

where Ux, Uy, Sx, Sy are the means and standard deviations of the marginal probability matrices Px, Py.

$$\begin{matrix}0 & 0 & 0\\ 3 & 3 & 3\\ 0 & 0 & 0\end{matrix}$$

(a)

N=4

COR1 = 1.0000
COR2 = -1.0000
COR3 = -1.0000
COR4 = -1.0000

(b)

Correlation measures in four directions

Fig. 2-8 Correlation measurement.

COR measures the gray level linear dependencies in an image. $-1 \leq COR \leq 1$. The COR measures in Fig. 2-8 indicate perfect linear correlation in the horizontal direction and the inverse correlated property along vertical and the two diagonal directions. COR is also not invariant under monotonic gray level transformation.

4) **Sum of Squares**

$$SOS = \Sigma\, [(i-U)^2 P(i,j)].$$

SOS is likely the moment about the mean of P(i,j), U. This feature is difficult to interpret. It is not invariant under gray level transformation.

5) **Inverse Difference Moment**

$$IDM = \Sigma\, [P(i,j)/(1+(i-j)^2))];$$

IDM measures the difference among gray levels; $0 \leq IDM \leq 1$. When all non zero P(i,j) are located along the diagonal ((i,j) = (j,i)), then Sum [P(i,j)] = 1, i-j=0, IDM reaches its maximum value 1. Fig. 2-9 gives an example of the IDM measure and the normalized P(i,j) in the right diagonal coocurrence measure. IDM varies under gray level transformation.

N=4	SPADEP along left diagonal	Measures in four directions

$\begin{matrix}2&1&1\\3&2&1\\0&3&2\end{matrix}$	$\begin{matrix}0.00&0.00&0.00&0.00\\0.00&0.25&0.00&0.00\\0.00&0.00&0.50&0.00\\0.00&0.00&0.00&0.25\end{matrix}$	IDM1 = 0.5167
		IDM2 = 0.5167
		IDM3 = 0.2750
		IDM4 = 1.0000

Fig. 2-9 Inverse Difference Moment measurement.

6) **Sum Average**

$$\mathrm{SUMAVG} = \sum [KP_{x+y}(K)];\ i+j = K = 2, 3, 4, \ldots\ldots, 2N;$$

$$2P(1,1)\leq\mathrm{SUMAVG}\leq 2P(N, N).$$

This measure is also difficult to interpret. It seems that the high SUMAVG indicates the higher coocurrence among high gray levels (Fig. 2-10). SUMAVG is not an invariant under gray level transformation.

N=4		N=4	

$\begin{matrix}0&0&0\\3&3&3\\0&0&0\end{matrix}$	SUMAVG1 = 4.0000
	SUMAVG2 = 5.0000
	SUMAVG3 = 5.0000
	SUMAVG4 = 5.0000
		$\begin{matrix}0&0&0\\2&2&2\\0&0&0\end{matrix}$	SUMAVG1 = 3.0000
			SUMAVG2 = 4.0000
			SUMAVG3 = 4.0000
			SUMAVG4 = 4.0000

Fig.2-10 Sum average measurement.

7) **Entropy**

$$\mathrm{ETP} = -\sum \{P(i,j)\mathrm{Log}_2 [P(i,j)].$$

This is the measurement of average joint information or joint entropy. From the properties of entropy, $\mathrm{ETP}\leq \mathrm{ETPi} + \mathrm{ETPj}$, with the equality if and only if, the two gray levels i and j are statistically independent. ETPmax = $\text{Log}_2 R$ (R is the number of resolution pairs). For a 3X3 image, the R in horizontal and vertical directions is 12 (with neighboring distance and gray level interval equal to 1), ETPmax = $\text{Log}_2 12$ = 3.58496. Along the left and right diagonals, R = 8, ETPmax = $\text{Log}_2 8$ = 3.0. Fig. 2-11 shows the maximum case of ETP. Obviously, ETP measures the complexity of an image which can be defined as the number of gray levels in a subimage. The higher the ETP, the more complex it is. ETP is invariant under monotonic gray level transformation. Therefore, it is very useful for the comparative study of texture.

0 1 2
3 4 5
6 7 8

N=9

ETP1 = 3.5895
ETP2 = 3.5895
ETP3 = 3.0000
ETP4 = 3.0000

Fig. 2-11 Entropy measure, the maximum case.

8) **Sum Entropy**

$$\text{SUMETP} = -\sum \{P_{x+y}(i)\text{Log}_2[P_{x+y}(i)]\}.$$

Since P(i,j) is a symmetrical matrix, P(i,j) = P(j,i), $P_{x+y}(i)$

contains the sums for all (i+j) = (j+i) (Fig. 2-5a). SUMETP is simply another way to measure the complexity of an image. SUMETPmax = Log_2HR, where HR = R/2. For the same image of Figure 2-11, HR in horizontal and vertical direction equals 12/2 = 6, along left and right diagonal, HR = 8/2 = 4. In this maximum case, SUMETP is equal to 2.5849 and 2.0 respectively (Fig. 2-12). SUMETP is also an invariant under monotonic gray level transformation.

0	1	2
3	4	5
6	7	8

N=9

SUMETP1 = 2.5850
SUMETP2 = 2.5850
SUMETP3 = 2.0000
SUMETP4 = 2.0000

Fig. 2-12 Sum entropy measure, the maximum case.

9) **Difference Entropy**

$$\text{DIFETP} = -\Sigma\,\{P_{x-y}(i)\text{Log}_2\,[P_{x-y}(i)]\}.$$

$P_{x-y}(i)$ is the sum for each |i-j| = 0,1,2,......,N-1; in P(i,j) (Fig. 2-5b). The minimum value of DIFETP occurs when there is only one non-zero value in $P_{x-y}(i)$, i.e., non-zero values in P(i,j) cluster in one group of |i-j| = 0, 1, 2, ..., N-1. DIFETP measures the similarity of spatial relationships among gray levels in an image. The high DIFETP indicates more different spatial relationships among gray levels. In the image of Figure 2-11, the neighboring relations are the same in each direction, i.e., one gray level difference along the horizontal direction, three levels along the vertical, two levels and four levels difference along 45 degree and 135 degree angles respectively. As a result, the DIFETP in all directions equals to 0. Fig. 2-13 are some other examples of the DIFETP measures. It seems that DIFETP would be useful to measure the regularity of an image. Compared with the IDM measure, DIFETP reveals spatial relationships at another level. Moreover, DIFETP is invariant under monotonic gray level transformation.

$$\begin{matrix}0&0&0\\2&2&2\\0&0&0\end{matrix}$$

DIFETP1 = 0.0000
DIFETP2 = 0.0000
DIFETP3 = 0.0000
DIFETP4 = 0.0000

$$\begin{matrix}1&2&3\\1&2&3\\1&2&3\end{matrix}$$

DIFETP1 = 0.0000
DIFETP2 = 0.0000
DIFETP3 = 0.0000
DIFETP4 = 0.0000

$$\begin{matrix}1&0&0\\2&2&2\\0&0&0\end{matrix}$$

DIFETP1 = 0.6500
DIFETP2 = 0.6500
DIFETP3 = 0.0000
DIFETP4 = 0.8113

N=4

Fig. 2-13 Examples of DIFETP measurement.

10) Sum Variance

$$\text{SUMVAR} = \Sigma\, [(\text{i-SUMETP})^2 P_{x+y}(\text{i})].$$

Let's consider the maximum/minimum case. When there is only one gray level in an image, the sum entropy equals to 0; values in $P_{x+y}(i)$ are focused on $P_{x+y}(i) = 1$, SUMVARmax = i^2. The higher occurrence of neighboring high gray levels, the larger the SUMVAR; SUMVAR becomes smaller when SUMETP increases and $P_{x+y}(i)$ is widely distributed (Fig. 2-14).

N=9

$$\begin{matrix}8&8&8\\8&8&8\\8&8&8\end{matrix}$$

SUMVAR1 = 324.0
SUMVAR2 = 324.0
SUMVAR3 = 324.0
SUMVAR4 = 324.0

$$\begin{matrix}2&2&2\\2&2&2\\2&2&2\end{matrix}$$

SUMVAR1 = 36.0000
SUMVAR2 = 36.0000
SUMVAR3 = 36.0000
SUMVAR4 = 36.0000

N=9

$$\begin{matrix}0&1&2\\3&4&5\\6&7&8\end{matrix}$$

SUMVAR1 = 79.9873
SUMVAR2 = 66.6495
SUMVAR3 = 74.0000
SUMVAR4 = 74.0000

*SUMVAR is not independent of gray levels in an image.

Fig. 2-14 Sum variance measurement.

The textural property expressed by SUMVAR is not clear. Moreover, since the measure likely varies with gray level, the same spatial relationship but different primitives would have different SUMVAR values (Fig. 2-14). Therefore, SUMVAR is not a very desirable measure for texture analysis.

11) Difference Variance

$$\text{DIFVAR} = \Sigma\,[(i\text{-DIFETP})^2 P_{x\text{-}y}(i)].$$

DIFVAR measures the variation in the P(i,j) matrix. It is similar to the contrast measure. In fact, when the DIFETP equals to 0, DIFVAR and CON have the same value (Fig. 2-15). DIFVAR is invariant under monotonic gray level transformation, therefore, it is more useful in a comparative study.

N=9			
	DIFVAR1 = 1.0000	CON1 = 1.0000	
0 1 2	DIFVAR2 = 9.0000	CON2 = 9.0000	(a)
3 4 5	DIFVAR3 = 4.0000	CON3 = 4.0000	
6 7 8	DIFVAR4 =16.0000	CON4 =16.0000	
N=4			
0 0 1 1	DIFETP1 = 0.9183	DIFVAR1 = 0.5644	
0 0 1 1	DIFETP2 = 0.9183	DIFVAR2 = 0.2855	(b)
0 0 2 2	DIFETP3 = 0.9911	DIFVAR3 = 0.4366	
2 2 3 3	DIFETP4 = 1.4335	DIFVAR4 = 0.6341	

Fig. 2-15 Comparison of DIFVAR with CON and DIFETP.

12) Information Measures of Correlation

Haralick proposed two models of IMC:

$$\text{IMCI} = (\text{HXY} - \text{HXY1})/\text{Max}(\text{HX}, \text{HY});$$

$$\text{IMCII} = \text{SQRT}\,\{1 - \exp\,[-2.0(\text{HXY2} - \text{HXY})]\};$$

where

$\text{HXY} = -\,\Sigma\ P(i,j)\text{Log}_2[P(i,j)];$ the joint entropy of P(i,j);

$HX = -\Sigma\ Px(i)Log_2[Px(i)]$;   entropy of the marginal matrix Px(i);

$HY = -\Sigma\ Py(j)Log_2[Py(j)]$;   entropy of the marginal matrix Py(i);

$HXY1 = -\Sigma\ P(i,j)Log_2[Px(i)Py(j)]$;   conditional entropy;

$HXY2 = -\Sigma\ Px(i)Py(j)Log_2[Px(i)Py(j)]$.

The theoretical connotation of these measures are complicated. It is noticed by experiment that $-1 \le IMCI \le 0$, $-1 \le IMCII \le 1$. For IMCII, let A=HXY2 - HXY, we see exp(-2.0*A)≤1. The higher the A, the smaller the exp(-2.0*A), the larger the IMCII. Since P(i,j) is symmetrical, HXY1 = HXY2. IMCII becomes large when P(i,j) is equally distributed. The maximum value occurs when any two pixels do not have the same value in a defined distance within the image. It is smaller when the image is dominated by only a few gray levels. There is more information contained in these correlation measures. They have some desirable properties which are not brought out in the rectangular correlation (COR) (Fig. 2-16).

1 2 3	IMCI1 = -0.8379	IMCII1 = 0.9972	COR1 = 0.9231	
4 5 6	IMCI2 = -0.8379	IMCII2 = 0.9972	COR2 = 0.1290	(a)
7 8 9	IMCI3 = -0.9091	IMCII3 = 0.9966	COR3 = 0.4286	
	IMCI4 = -0.9091	IMCII4 = 0.9966	COR4 =-2.3077	
N=10				
1 1 1	IMCI1 = -1.0000	IMCII1 = 0.9168	COR1 = 1.0000	
8 8 8	IMCI2 = -0.1500	IMCII2 = 0.4662	COR2 =-0.3000	(b)
8 8 8	IMCI3 = -0.1500	IMCII3 = 0.4662	COR3 =-0.3000	
	IMCI3 = -0.1500	IMCII4 = 0.4662	COR4 =-0.3000	
N=9				

Fig. 2-16 Compare IMC with COR.

13) **Maximal Correlation Coefficient**

$$\text{MAXCOR} = (\text{Second Largest Eigenvalue of } Q)^{1/2};$$

$$Q(i,j) = \Sigma\,[P(i,k)P(j,k) \,/\, Px(i)Py(k)].$$

Fig. 2-17a is the Q matrix for the image in Fig. 2-15a (horizontal direction). Since Q(i,j) is not symmetrical, the computation of the eigenvalue is somewhat complicated. It is known that most matrices can be transformed to a Hessenberg matrix and it is easier to compute the eigenvalue from the Hessenberg matrix. Fig. 2-17b is the Hessenberg transform of the Q(i,j) in Fig. 2-17a. For detailed procedures in computing eigenvalue of an asymmetrical matrix, refer to Fortran subroutine UPPERH and EIGEN in Appendix 4.

The mathematical connotation of the maximal correlation coefficient is discussed by C. B. Bell (1962). MAXCOR is different from COR. For the same image of Fig. 2-15a, maximal correlations in all four directions is equal to the maximum value 1.0. For the 3x3 image in Fig.2-16b, MAXCOR has the following measures: 1.0, 0.3, 0.3, 0.3.

The above statistical texture model provides a set of measures for the spatial relationships among gray levels in an image. However, texture properties are independent of gray level and orientation. In order to perform texture analysis, we wish the texture measures to be invariant in different orientations and under monotonic gray level transformation. Yet, the 14 texture features discussed above are all angular dependent and, only seven of them are invariant under monotonic gray level transformation: ASM, ETP, SUMETP, DIFETP, IMCI, IMCII, and MAXCOR. Therefore, it is more desirable to use the mean and the range of the measures in all four directions. Moreover, to obtain generalized results, equal probability gray level transformation, i.e., histogram equalization, must be performed. Nevertheless, it is still more preferable to use the invariants for textural analysis.

The Q matrix

```
0.5000 0.0000 0.5000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.5000 0.0000 0.5000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.5000 0.0000 0.5000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000     (a)
0.0000 0.0000 0.0000 0.5000 0.0000 0.5000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.5000 0.0000 0.5000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.5000 0.0000 0.5000
```

The Hessenberg matrix

```
0.5000 0.5000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.5000 0.5000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.5000 0.5000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.5000 0.5000 0.0000 0.0000 0.0000 0.0000     (b)
0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.5000 0.5000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.5000 0.5000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000
```

Fig. 2-17 The Q matrix and its Hessenberg transform.

C. The Computer Implementation of SPADEP Approach

Three steps are identified to perform SPADEP textural analysis, using PDP-11 and the Spatial Data Processing System.

1. Gray Level Transformation.

Since SPADEP is a coocurrence measurement, in order to obtain generalized textural information, it is necessary to change the gray level distribution so that each gray level in an image can obtain approximately equal probability. Furthermore, although the computation of SPADEP matrices are only related to the size of the image, the size of the SPADEP matrices is proportional to the number of gray levels in the image. For gray levels equal to N, the required internal storage will be N X N X 4. The range of gray levels in a TM image is 0-255. An array of 256X256X4 cannot be stored in the internal memory of the PDP-11 computer. For this reason, the number of gray levels must be reduced. An equal probability quantizing algorithm can solve the above problem. For details on the procedure, see program EGAL in Appendix I.

2. Compute SPADEP matrices.

The SPADEP matrix proposed by Haralick contains adjacencies of gray levels in both orientations and four directions. Therefore, SPADEP is symmetrical, i.e., in SPADEP P(i,j)=P(j,i). In actual computation, only one orientation needs to be measured, the final matrix is constructed by adding the matrix obtained previously to its transpose (Hord, 1986). Figure 2-18 illustrates the construction of SPADEP in the horizontal direction.

$$\begin{array}{ccc} \begin{matrix}1&1&0\\0&0&2\\2&2&1\end{matrix} & \begin{matrix}1&0&1\\1&1&0\\0&1&1\end{matrix} & \begin{matrix}2&1&1\\1&2&1\\1&1&2\end{matrix} \\ \text{original image} & \text{right adjacencies} & \text{add to the transpose} \\ (N=4) & (\text{distance} = 1) & \end{array}$$

Fig. 2-18 Illustration of SPADEP matrix construction (horizontal).

The Fortran subroutine to compute the coocurrence matrix, SPADEP, is included in Appendix 4. For the convenience of programming, SPADEP is divided into four, two dimensional arrays instead of using one three dimensional array. Each array stores the SPADEP in one direction. When using the Extended Memory Monitor (virtual memory) on PDP-11, the input image can have a size up to 128 by 128 and maximum number of gray levels of 128.

3. Computing the Statistical Texture Models

Most of the statistical models proposed by Haralick are easy to compute. The most difficult one is the maximal correlation coefficient measure since its computation involves finding the second largest eigenvalue from an asymmetrical matrix. An example of the approach is given in the former section (Fig. 2-17). All basic computations of the statistical textural measures are included in the subroutine library TXLIB (see Appendix 4).

II. The Entropy-Based Texture Analysis

An image can be defined in the spatial domain, i.e., the x, y coordinate space, or in the spatial frequency domain in which an image is viewed as a periodic function and represented by an infinite, weighted sum of trigonometric sine and cosine functions with different amplitudes, frequencies and phases. This representation is termed the Fourier series of an image. The SPADEP approach discussed previously represents textural analysis in the spatial domain. The second method to be used, the Entropy-based Textural analysis, is performed in the spatial frequency domain. It extracts texture information from another dimension. In this section, the two dimensional discrete Fourier transform, the computer implementation of the fast Fourier transform, and the display of the Fourier spectrum will be discussed briefly; then the detailed procedures to perform the entropy-based textural analysis is presented.

A. The Fourier Transform of an Image

Let f(m,n) be an N X N image and F(u,v) its two dimensional discrete Fourier transform, then

$F(u,v) = \Sigma\Sigma\ f(m,n)\ e^{-j(2\pi/N)(mu+nv)}$ (inverse transform);

$f(m,n) = \Sigma\Sigma\ F(u,v)\ e^{j(2\pi/N)(mu+nv)}$ (forward transform);

are the discrete Fourier transform pair. The Fourier spectrum, phase, and energy spectrum are given by the following relations:

$$|F(u,v)| = SQRT(R(u,v)^2 + I(u,v)^2);$$

$$Phase(u,v) = \tan^{-1}(I(u,v)/R(u,v));$$

$$E(u,v) = |F(u,v)|^2.$$

R and I denote the real and the imaginery part of F(u,v). The two dimensional discrete Fourier transform F(u,v) is obtained by performing a one dimensional transform along each row of f(m,n), then along each column of the intermediate matrix. Fig. 2-20a is an example of the Fourier transform of the image in Fig. 2-19. Since the discrete Fourier is periodic with period N and is conjugate symmetrical, it is more desirable to shift the frequency origin to the center (N/2+1, N/2+1) in order to observe and measure the function. Fig. 2-20b is the origin centered Fourier transform. It is obtained by multiplying each entry of the input image of Figure 2-19 by $(-1)^{(i+j)}$.

$$\begin{matrix} 7 & 7 & 7 & 8 & 8 & 8 & 6 & 6 \\ 7 & 7 & 7 & 8 & 8 & 8 & 6 & 6 \\ 7 & 7 & 7 & 8 & 8 & 8 & 6 & 6 \\ 6 & 6 & 6 & 9 & 9 & 9 & 9 & 9 \\ 6 & 6 & 6 & 9 & 9 & 9 & 9 & 9 \\ 6 & 6 & 6 & 9 & 9 & 9 & 9 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Fig. 2-19 An 8x8 matrix to be transformed to the Fourier series.

The Fourier spectrum can be shown in a three-dimensional plot or as an intensity function in which brightness is proportional to the amplitude of $|F(u,v)|$. The graphic display of the Fourier spectrum is very helpful to visualize certain textural properties of an image. Usually, radial spikes in the spectrum image indicate presence of linear features and the breadth of the bright area around the center indicates the coarseness of the image.

B. Regional Entropy Measures in the Fourier Energy Spectrum

Proposed by Jernigan and D'astous (1983) the regional entropy measure is primarily designed to measure local and global texture properties of an image. This approach can be described as follows (see Fig. 2-21):

1). Compute the origin centered Fourier transform of the analyzed image; compute the energy spectrum $E(u,v)=|F(u,v)|^2$;

2) Specify the number and size of concentric regions to be measured in $E(u,v)$. For each region, perform the following computations:

3) Obtain the regional energy by summing the $E(u,v)$ within the region, $SUME = \Sigma\, [E(u,v)]$;

4) Normalize $E(u,v)$ in the region by SUME, obtain the probability function $P(u,v) = E(u,v)/SUME$; $\Sigma\, [P(u,v)] = 1$;

5) Compute the entropy of the spectral components within the region,

$$h' = -\Sigma\, P(u,v)\mathrm{Log}_2[P(u,v)],$$

$$0 \leq h' \leq \mathrm{Log}_2 K,$$

where k is the number of elements in the E(u,v) within the region;

6) Normalize h' by LogK, obtain the relative entropy

$$h = h'/\mathrm{Log}_2 K, \qquad 0 \leq h \leq 1.$$

Suppose n regions are defined, as a result, there is a n-dimensional vector representing the textural properties of the image,

$$H = [h1, h2, h3, h4, \ldots\ldots, hn].$$

To compare texture features of different images, H is computed for each of the images. Obviously, hn is the measure of the spread of the spatial frequency component within the region. The higher the hn, the wider distribution of the frequency components indicated. Images with different texture features will have different characteristics of the frequency component distribution. For example, Fig. 2-22c has high frequency components concentrated along the vertical axis while Fig. 2-22a is more spread out.

0,0 v

u

45.000 0.000	-3.091 1.280	0.750 0.750	0.091 -0.220	-1.500 0.000	0.091 0.220	0.750 -0.750	-3.091 -1.280
-6.849 -12.163	0.515 -1.244	0.213 -1.545	0.567 0.088	0.692 0.213	0.442 -0.567	0.640 0.692	1.244 2.576
7.875 -7.125	-0.854 1.030	-0.125 0.125	-0.073 -0.250	-0.375 0.125	-0.146 -0.030	0.125 -0.625	-0.427 -0.250
5.349 2.087	-0.317 0.088	0.110 0.192	-0.015 0.006	-0.192 -0.037	0.006 0.076	0.037 -0.045	-0.442 -0.317
-0.750 0.000	0.250 -0.854	0.250 -0.500	0.250 0.146	0.250 0.000	0.250 -0.146	0.250 0.500	0.250 0.854
5.349 -2.087	-0.442 0.317	0.037 0.045	0.006 -0.076	-0.192 0.037	-0.015 -0.006	0.110 -0.192	-0.317 -0.088
7.875 7.125	-0.427 0.250	0.125 0.625	-0.146 0.030	-0.375 -0.125	-0.073 0.250	-0.125 -0.125	-0.854 -1.030
-6.849 12.163	1.244 -2.576	0.640 -0.692	0.442 0.567	0.692 -0.213	0.567 -0.088	0.213 1.545	0.515 1.244

(a)

0.250 0.000	0.250 -0.146	0.250 0.500	0.250 0.854	-0.750 0.000	0.250 -0.854	0.250 -0.500	0.250 0.146
-0.192 0.037	-0.015 -0.006	0.110 -0.192	-0.317 -0.088	5.349 -2.087	-0.442 0.317	0.037 0.045	0.006 -0.076
-0.375 -0.125	-0.073 0.250	-0.125 -0.125	-0.854 -1.030	7.875 7.125	-0.427 0.250	0.125 0.625	-0.146 0.030
0.692 -0.213	0.567 -0.088	0.213 1.545	0.515 1.244	-6.849 12.163	1.244 -2.576	0.640 -0.692	0.442 0.567
-1.500 0.000	0.091 0.220	0.750 -0.750	-3.091 -1.280	45.000 0.000	-3.091 1.280	0.750 0.750	0.091 -0.220
0.692 0.213	0.442 -0.567	0.640 0.692	1.244 2.576	-6.849 -12.163	0.515 -1.244	0.213 -1.545	0.567 0.088
-0.375 0.125	-0.146 -0.030	0.125 -0.625	-0.427 -0.250	7.875 -7.125	-0.854 1.030	-0.125 0.125	-0.073 -0.250
-0.192 -0.037	0.006 0.076	0.037 -0.045	-0.442 -0.317	5.349 2.087	-0.317 0.088	0.110 0.192	-0.015 0.006

(b)

Fig. 2-20 The Fourier transform of fig.19; (b) original centered.

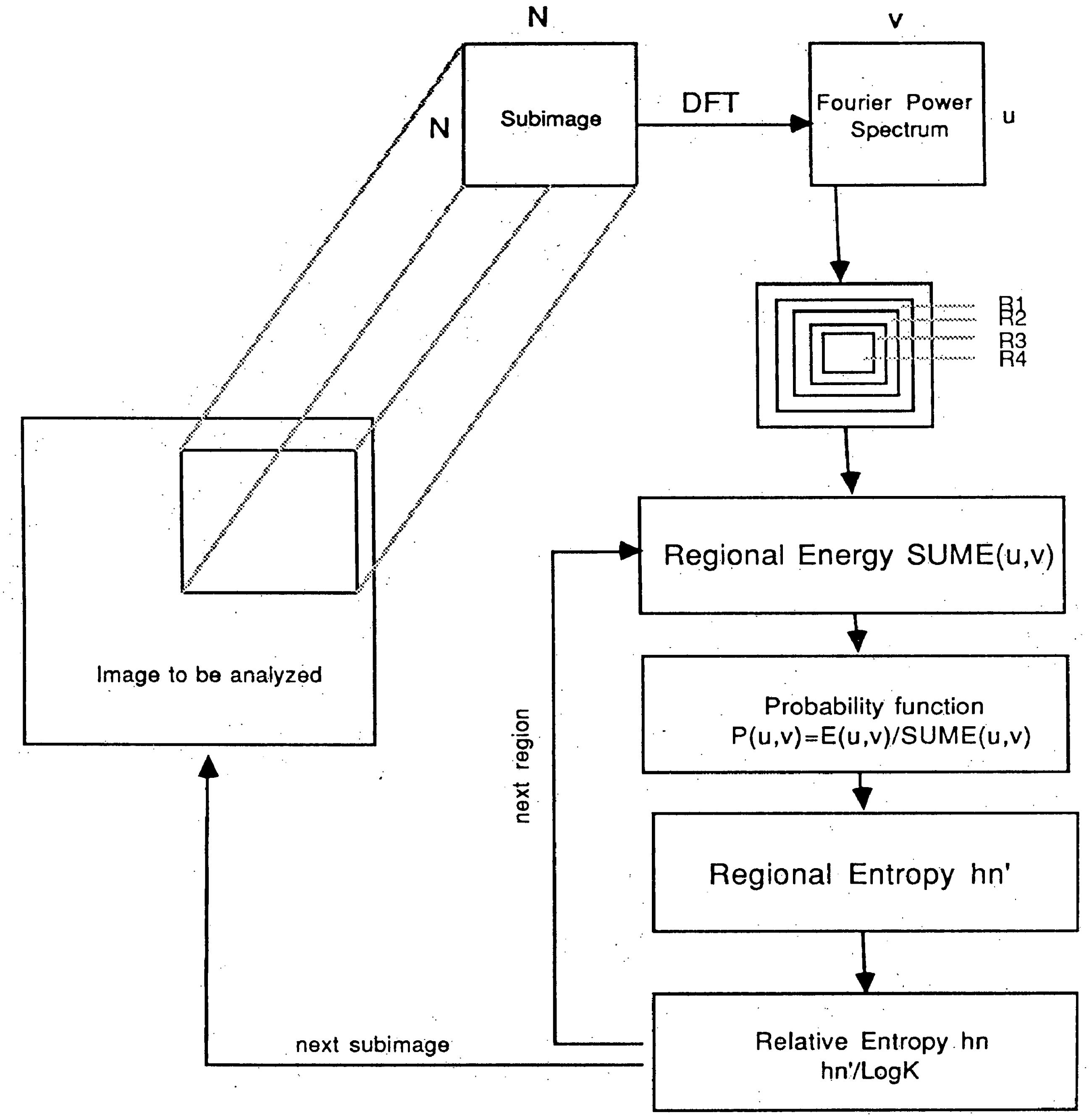


Fig. 2-21 EBT analysis.

C. The Computer Implementation of the Fourier Transform: the FFT

The discrete Fourier transform is implemented with the Fast Fourier Transform algorithm proposed by Cooley-Tukey. The FFT approach dramatically reduces the computation times from N^2 to NLogN (N is the number of input elements). Detailed discussion of the FFT algorithm can be found in variety of text books in digital remote sensing (Rosenfield, et. al., 1982, Gonalez and Wintz, 1977). A standard FFT Fortran subroutine FOUREA is included in Appendix 8.

0.250	0.290	0.559	0.889	0.750	0.889	0.559	0.290
0.195	0.016	0.221	0.329	5.741	0.544	0.058	0.076
0.395	0.261	0.177	1.338	10.620	0.495	0.637	0.150
0.724	0.574	1.560	1.346	13.959	2.860	0.943	0.719
1.500	0.238	1.061	3.346	45.000	3.346	1.061	0.238
0.724	0.719	0.943	2.860	13.959	1.346	1.560	0.574
0.395	0.150	0.637	0.495	10.620	1.338	0.177	0.261
0.195	0.076	0.058	0.544	5.741	0.329	0.221	0.016

(a) The Fourier spectrum of fig. 2-19.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b) An 8X8 image.

0.000	0.000	0.000	0.000	2.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.828	0.000	0.000	0.000
0.000	0.000	0.000	0.000	2.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	4.828	0.000	0.000	0.000
0.000	0.000	0.000	0.000	6.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	4.828	0.000	0.000	0.000
0.000	0.000	0.000	0.000	2.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.828	0.000	0.000	0.000

(c) The Fourier spectrum of (b).

Fig. 2-22 Texture pattern and tho Fourior cpoctrum.

III. Methods to Analyze the Texture Measures

One of the major objectives of this thesis is to see if similar spatial patterns have similar texture measures. Methods are needed therefore, to compare the texture measures. In this project, the distance measure and cross-correlation analysis are used.

A. The Distance Measure

Suppose $X_1(n)$, $X_2(n)$, ..., $X_m(n)$ are the texture vectors of m images to be analyzed, the distance between two texture vectors can be computed as:

$$D(X_m, X_{m-1}) = SQRT\,\{\Sigma\,[X_m(n) - X_{m-1}(n)]^2]\}.$$

We say that X_{m-1} is more similar to X_j than to X_m if

$$D(X_{m-1}, X_j) < D(X_{m-1}, X_m).$$

B. Cross Correlation

The texture information of an image can also be represented by a texture matrix constructed by a local operation (Fig. 2-23). Since the texture property is contained in a series of local measurements, cross correlation can be used to measure the similarity between two texture matrices. Let $T_1(x,y)$ and $T_2(x,y)$ be the two texture functions, the cross correlation can be defined as

$$R(m,n) = \Sigma\ [T1(x,y)T2(x,y)]\ /\ SQRT(ST1ST2),$$

where,

$$ST1 = \Sigma\ [T1(x,y)^2];$$
$$ST2 = \Sigma\ [T2(x,y)^2];$$
$$-1 \leq R(m,n) \leq 1.$$

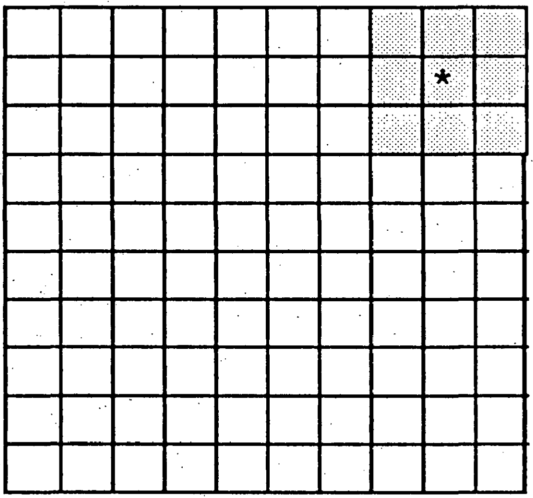


Fig. 2-23 Local measurement of texture properties.

When used to measure similarity between two functions, only the largest correlation function is usually of interest. Thus, a searching process in R(m,n) is needed. However, if the two functions are the same size, i.e., the images to be analyzed have the same number of rows and columns, cross-correlation can be more conveniently performed in the frequency domain:

$$R(m,n) <\text{--}\ T_1(u,v)T_2(u,v);$$

T1(u,v) and T2(u,v) are the forward Fourier transform of T1(x,y) and T2(x,y). The cross correlation of two functions is equal to the inverse Fourier transform of the product between one Fourier series and the complex conjugate of the other. A search for the largest correlation function is not needed in R(m,n) obtained in this way since it is always located around R(0,0) (when the two functions are the same size). For more details of the computation of cross correlation, refer to the Fortran program FFTCOR submitted in Appendix 14.

In this chapter, major technical aspects of the thesis have been discussed. Since it is a relatively exploratory study, a great deal of effort has been devoted to the investigation of the mathematical models to be used.

CHAPTER 3: ANALYSIS

I. INTRODUCTION

In this chapter, methods of texture analysis presented in chapter 2 are to be applied in measuring and differentiating urban residential spatial patterns. The SPADEP, both regional and local, and the Fourier measures are applied to a Landsat Thematic Mapper image of western Omaha. The general methodology for analysis is presented in Figure 3-1.

A 512 by 512 pixel TM subimage centered at about 132nd and Dodge St., Omaha, Nebraska, represents the study area (Fig. 3-2). Band 3 (0.63μm-0.69μm, red reflectance) is the basic spectral band used for textural analysis since it provides the better penetration of the atmosphere among the visible wavelengths and provides a higher contrast image. Ten subimage areas are selected, sized 32 (columns) by 32 (rows), numbered 1 to 10 (Fig. 3-3). The selection of subimage areas was purposive. The study areas represent the major residential patterns in the image area. Among these ten areas, area 1 is a low density residential area, 7 is a partially developed urban area; 9 includes cleared subdivisions; 10 is agricultural land; the other areas

(2, 3, 4, 5, 6) represent other major residential patterns in this region, such as the traditional grid street layout and the low density, irregular new residential area.

There are basically three parts to the analysis: the SPADEP regional measurement, the SPADEP local measurement, and Fourier analysis. In each of these parts, individual texture measures will be observed with regard to the spatial relationships represented. Then, a similarity measurement (distance measure and cross-correlation measure) is applied to the texture measures of all spatial patterns.

Fig. 3-2 Study Areas.

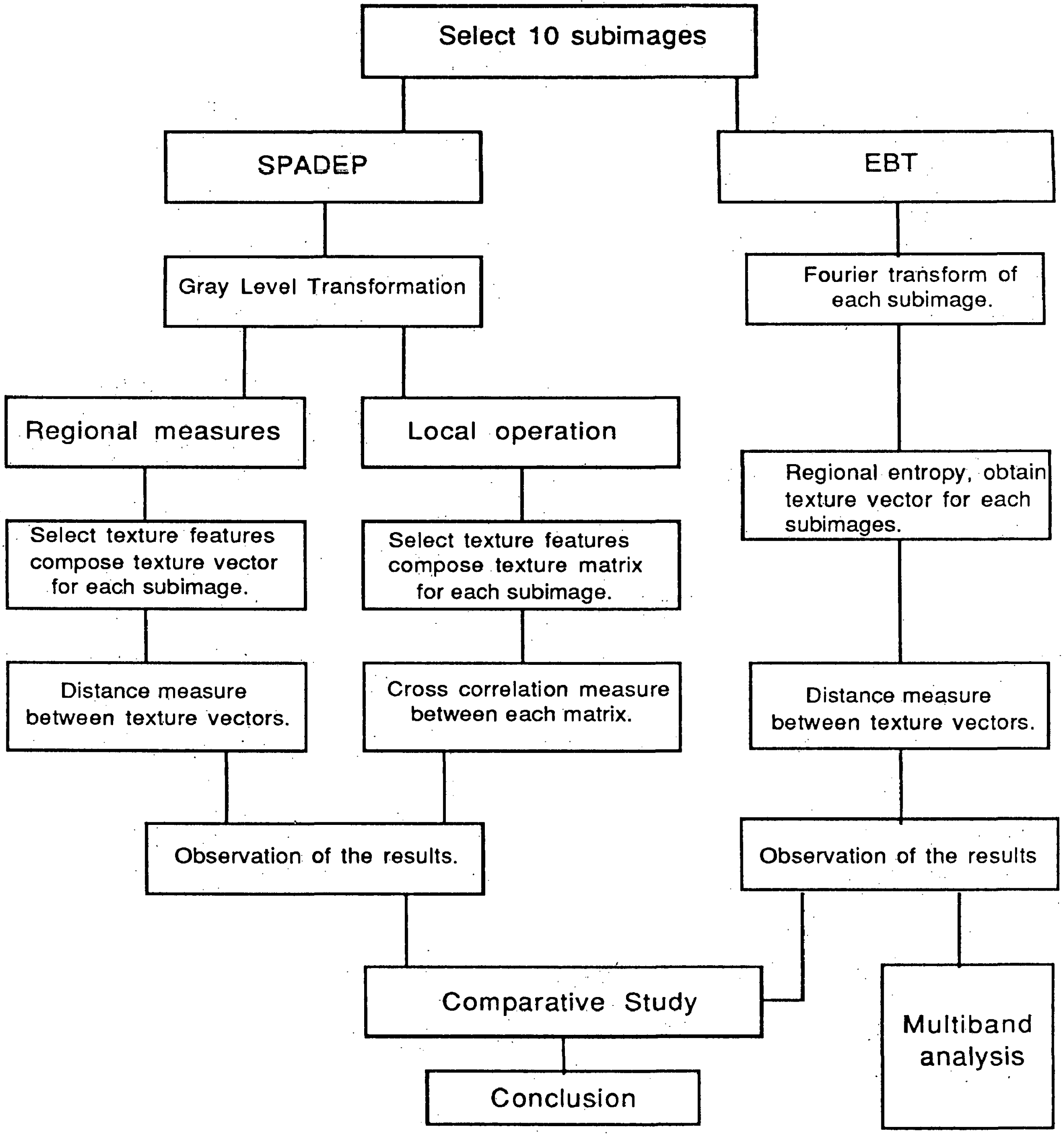
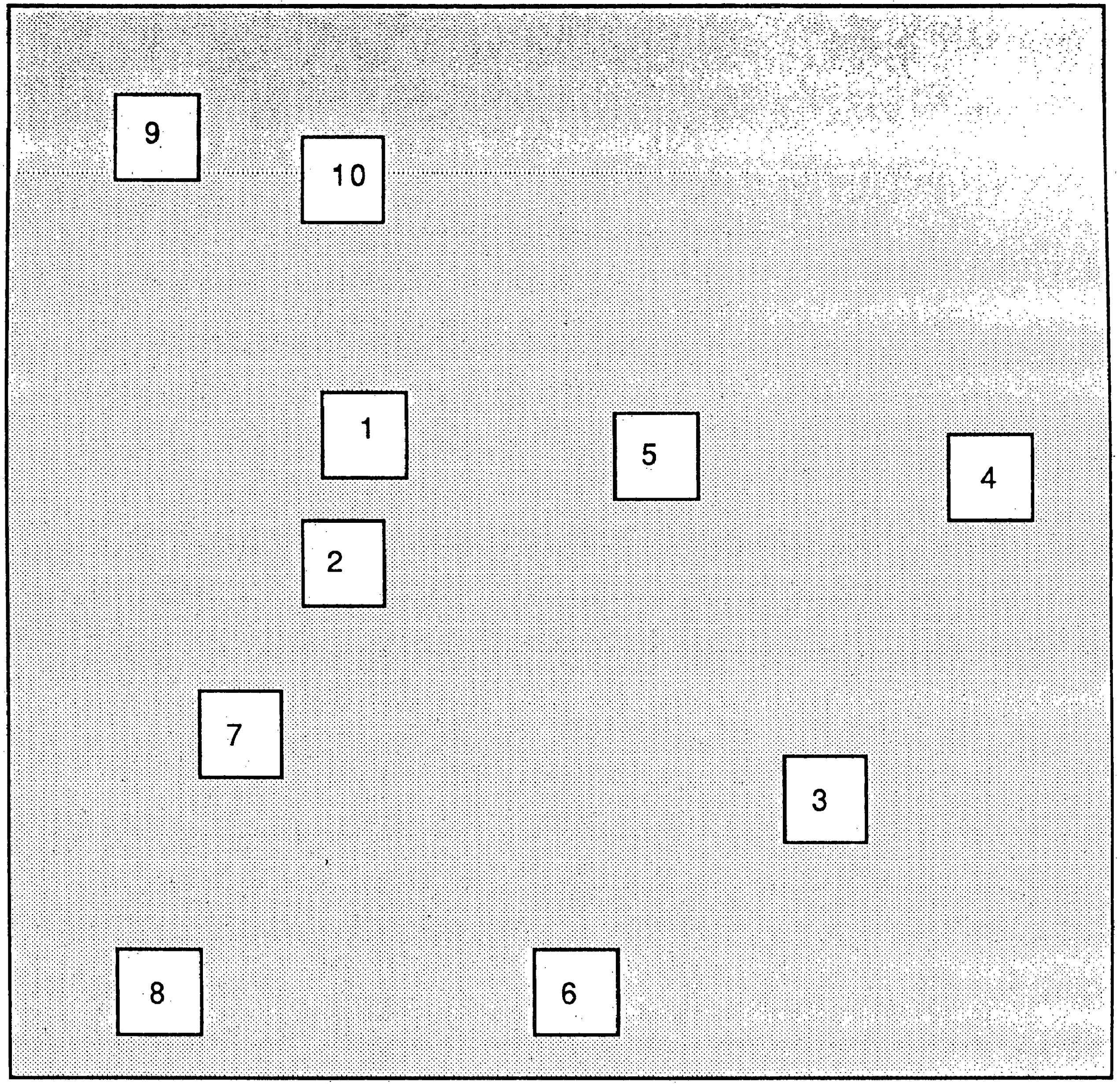

Select 10 subimages
SPADEP
EBT
Gray Level Transformation
Fourier transform of each subimage.
Regional measures
Local operation
Regional entropy, obtain texture vector for each subimages.
Select texture features compose texture vector for each subimage.
Select texture features compose texture matrix for each subimage.
Distance measure between texture vectors.
Cross correlation measure between each matrix.
Distance measure between texture vectors.
Observation of the results.
Observation of the results
Comparative Study
Multiband analysis
Conclusion

Fig. 3-1 Guideline of texture analysis



COORDINATES:	(199,168),	(191,208),	(341,405),	(457,201),	(284,189)
	(185,408),	(130,307),	(51, 416),	(97, 83),	(192,96)

Fig. 3-3 Study areas.

II. Texture Analysis in a Suburban Area

A. SPADEP Regional Measurement

As required by the SPADEP approach, the image to be analyzed is reduced to 32 gray levels with the historgram equalization program EGAL.

Among the 14 statistical texture measures discussed in chapter 2, seven of them are selected for comparative study due to the reasons discussed in chapter 2. They are: Angular Second Moment (ASM), Correlation (COR), Sum Entropy (SUMETP), Entropy (ETP), Difference Entropy (DIFETP), Information Measure of Correlation II (IMCII), and Maximal Correlation (MAXCOR). All but one of these measures are invariant under monotonic gray level transformation.

Since texture measures from SPADEP are angular dependent, measures in specific directions would not represent the overall texture features of the area. Therefore, the range, mean and variant instead of measures in the four directions are used as the textural descriptors. Thus, there are 7 X 3 or 21 total vectors for each subimage area. To experiment with the textural similarities of the ten areas, a distance measure is performed.

Fig. 3-4 graphs the average ASM measures among the ten areas. The non-built and partially built area 7, 9, 10 have distinguishable values from the residential areas while the variations for the areas that are essentially residential are very small. As indicated in chapter two, ASM measures the homogeneity of the area. From Fig. 3-4, we see that the non-built areas have high ASM, indicating that the cleared subdivision and the partially built areas (7, 9, 10) have higher homogeneity than the residential areas. In this experiment, ASM does not reflect the difference of homogeneity among residential patterns.

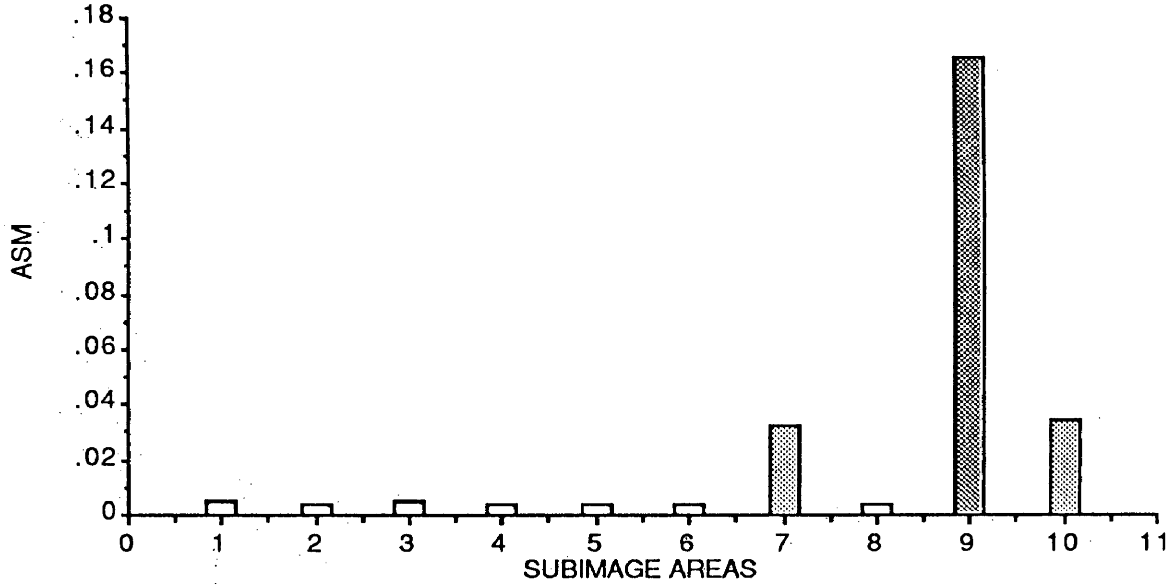


Fig. 3-4 ASM measures.

Of the average correlation measures (COR) among the ten areas, area 7, 9, 10 have the higher values (Fig. 3-5).

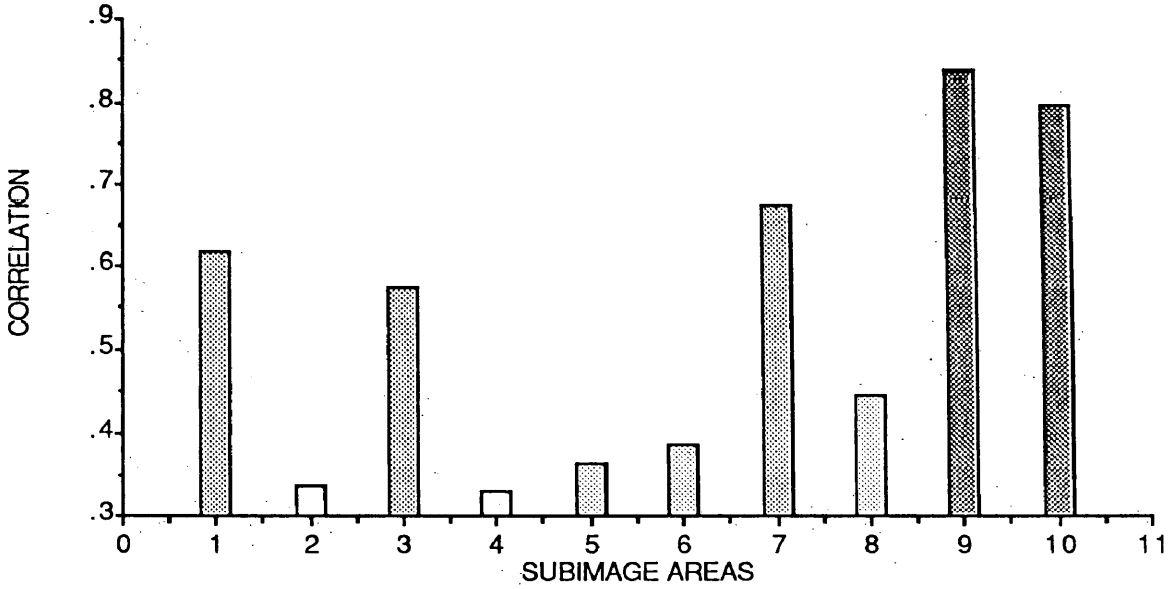


Fig. 3-5 Correlation measures.

For these areas, the high linear correlation is probably produced by the long edges of the cleared subdivision and agricultural land. Fig. 3-5 also shows a certain amount of variation of correlation measures among the residential areas. Although residential areas would have typical linear patterns, the high linear correlation would be only in one or two directions. In the other directions, the correlation value would be very low, thus, reducing the average measure. Another reason for such a distribution of the correlation measures is related to the distance used to compute the neighboring coocurrence. A distance of 1 (neighboring distance and gray level interval) may be too small to reveal the linear features presented in the residential areas. Therefore, the high and low correlations do mean something here, e.g., the coarseness of the image, but not necessarily the visualized linear feature.

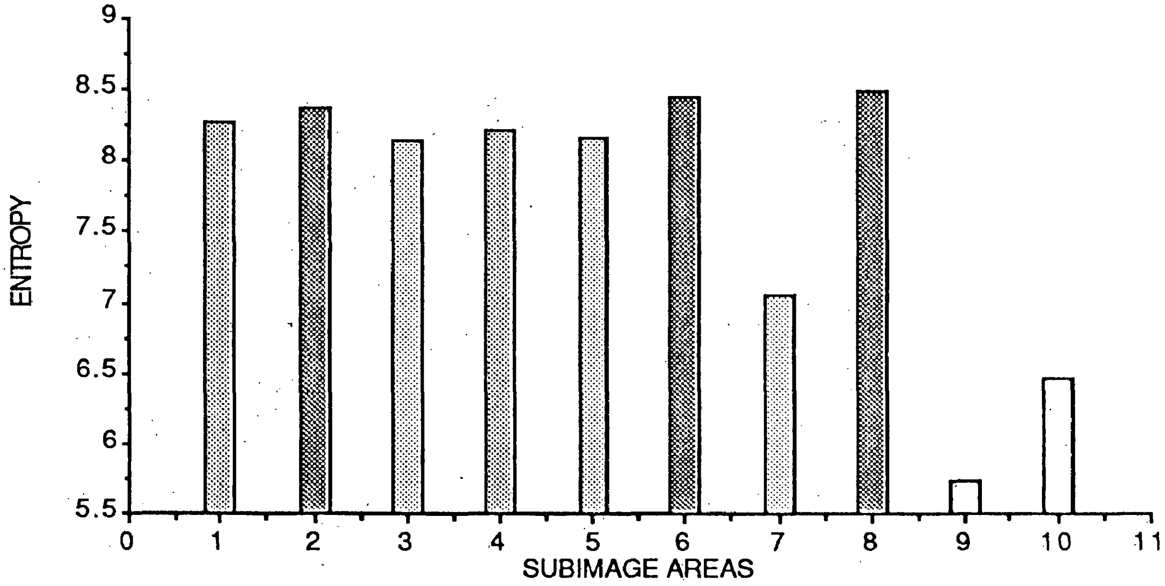


Fig. 3-6 Entropy measures.

The next three measures are entropy based. The Sum Entropy (SUMETP) and Entropy (ETP) measure the relative complexity of the area. From Fig. 3-6 and Fig. 3-7, we see that all residential areas have higher values of SUMETP and ETP, indicating a higher complexity than that of the non-built areas. However, how can the differences among the residential areas be explained? From Figure 3-8, we find that area 3 has a very small measure in range, the similar phenomena happened to area 4 and 6 in Figure 3-9. Area 3, 4, 6 are all older neighborhoods. It seems that less variation in complexity with directions indicates the higher degree of development among residential areas. As discussed before, DIFETP may measure the irregularity of an area. In Fig. 3-10 areas 2, 4, 6 have higher average DIFETP since they are the most irregular patterns among the ten areas.

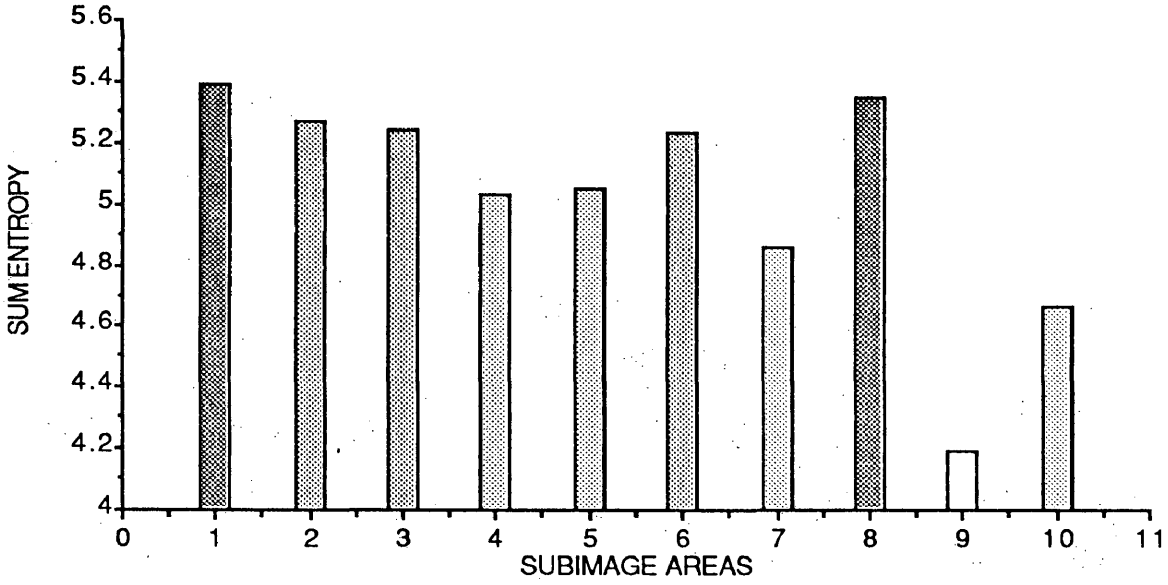


Fig. 3-7 Sum Entropy measures.

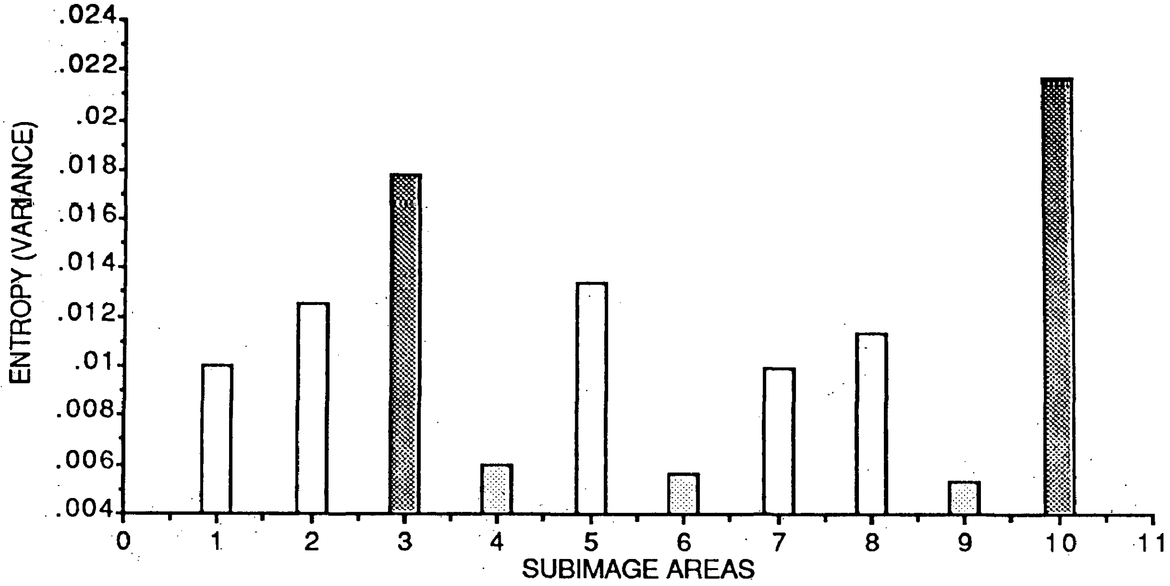


Fig. 3-8 Variance of the Entropy measures.

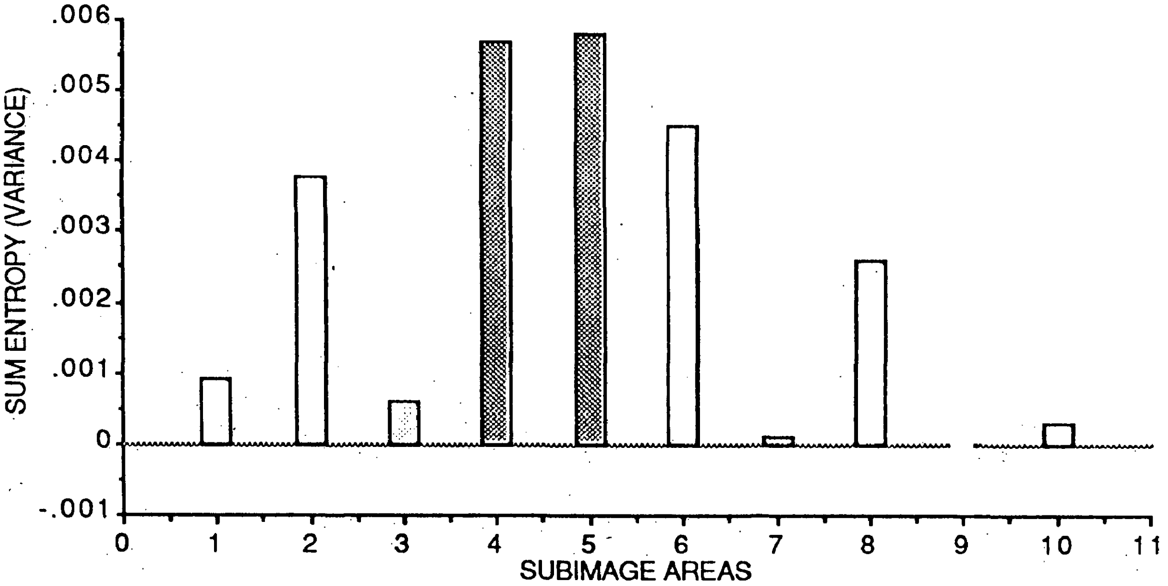


Fig. 3-9 Variance of the Sum Entropy measures.

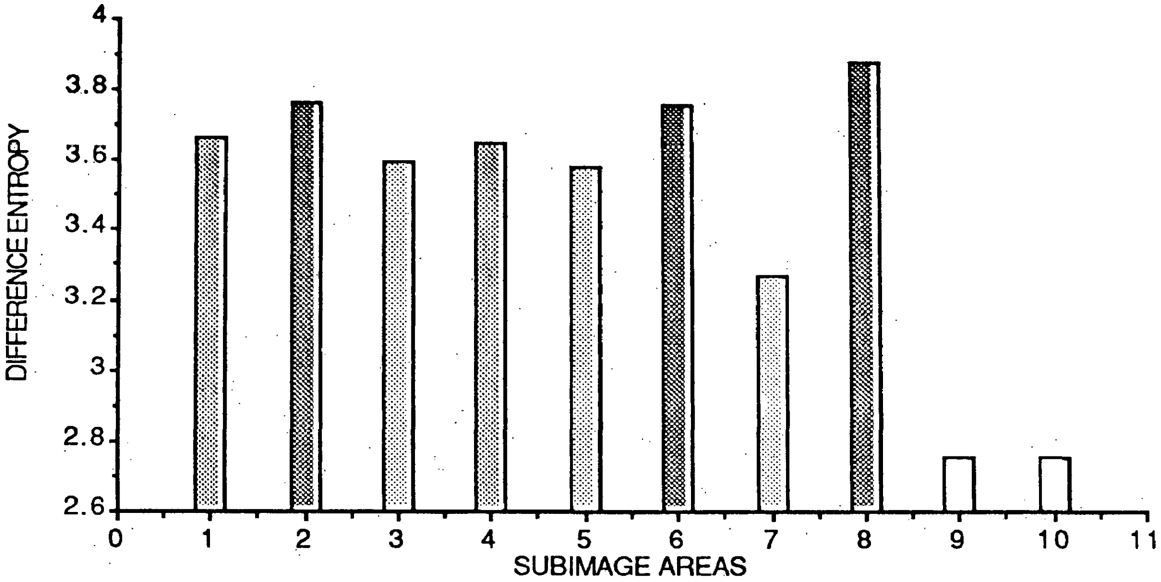


Fig. 3-10 Difference entropy measures.

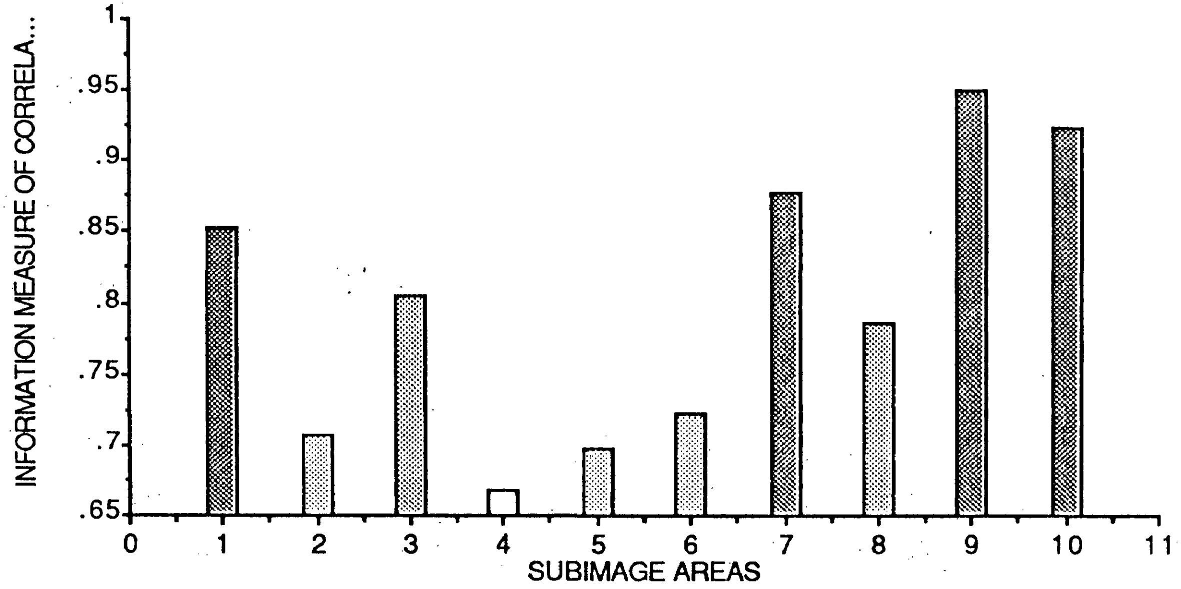


Fig. 3-11 Information measures of correlation.

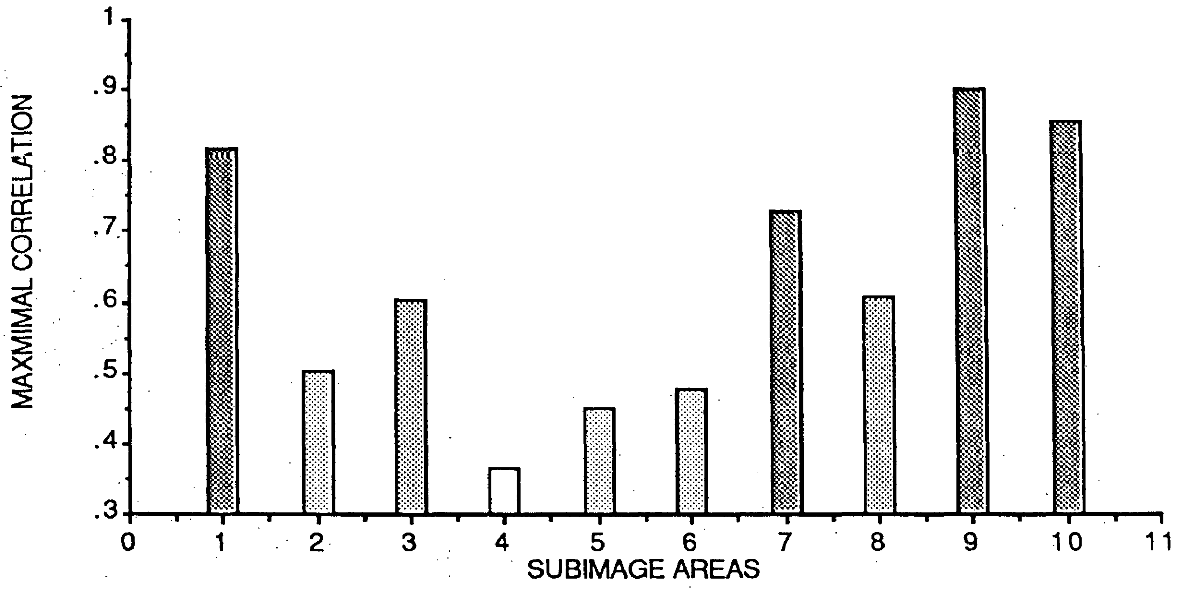


Fig. 3-12 Maximal correlation measures.

The last two texture measures are the Information Measure of Correlation (IMCII) and the Maximal Correlation (MAXCOR). The connotation of such measures is difficult to interpret, however, they seem to distinguish different spatial patterns quite well. This is illustrated by the wide variation of these measures graphed in Figure 3-11 and 3-12. The non-built areas have higher measures of IMCII and MAXCOR; among the built-areas, the lower density residential pattern and partially built areas have higher values. Furthermore, residential areas having higher density housing or more vegetation coverage, have higher values than those with less housing or vegetation coverage.

It seems that each measure tends to emphasize certain aspects of the spatial characteristics in a subimage. However, they are related with each other. When we group these measures together, they should represent the over-all texture features of the area. There are many ways to identify similar texture patterns. The simplest way is by computing the distance among all the study areas and those having the least distance can then be grouped together. Fig. 3-13 is the matrix of distance measures for the ten subimage areas. From the distance matrix, we easily find the following most similar groups: (1,3), (2, 6, 8), (4, 5), (9, 10, 7). This clearly indicates that similar spatial patterns present similar texture measures.

	1	2	3	4	5	6	7	8	9	10
1	0.000	0.645	$\boxed{0.389}$	0.703	0.719	0.541	1.410	0.481	2.987	2.195
2	0.645	0.000	0.484	0.419	0.394	$\boxed{0.314}$	1.543	0.457	3.107	2.339
3	$\boxed{0.389}$	0.484	0.000	0.508	0.422	0.491	1.223	0.518	2.817	2.012
4	0.703	0.419	0.508	0.000	$\boxed{0.326}$	0.350	1.373	0.599	2.907	2.171
5	0.719	0.394	0.422	$\boxed{0.326}$	0.000	0.456	1.323	0.598	2.865	2.094
6	0.541	0.314	0.491	0.350	0.456	0.000	1.596	$\boxed{0.300}$	3.168	2.406
7	1.410	1.543	1.223	1.373	1.323	1.596	0.000	1.687	1.601	$\boxed{0.855}$
8	0.481	0.457	0.518	0.599	0.598	$\boxed{0.300}$	1.687	0.000	3.273	2.496
9	2.987	3.107	2.817	2.907	2.865	3.168	1.601	3.273	0.000	$\boxed{0.926}$
10	2.195	2.339	2.012	2.171	2.094	2.406	$\boxed{0.855}$	2.496	0.926	0.000

Fig. 3-13 Distance measure using 7 texture features.

B. SPADEP Local Measurement

Figure 3-13 demonstrated that the SPADEP texture measures selected best distinguish spatial characteristics of the built and non-built areas. However, the measures represent the texture characteristics of the whole sub-area. Thus, the selection of areas to be analyzed is very important; they should have a consistent pattern.

In this experiment, area 7 is half subdivision and half developed area but the distance measures show that it is similar to the non-built area. The measure is reasonable but may not be desirable. However, texture consistency is not easy to be selected; therefore, in such case, it may be useful to apply local operation for texture analysis. In the following section, an experiment on local operation of SPADEP is presented. The size of the local operator is 3 by 3 and the texture matrices produced from each 32 by 32 image is 30 by 30. ASM is used as an example to compare the regional and local measures. To analyze the similarities, cross correlation is performed among the texture matrices. The largest correlation function is used as the entry in the correlation matrix. The higher the correlation coefficient, the more similar the two areas.

Subimage Areas

Subimage Areas	1	2	3	4	5	6	7	8	9	10
1	0.0000	0.8200	0.8192	0.8204	0.8193	0.8209	0.5984	0.7953	0.5832	0.6279
2		0.0000	0.9694	0.9773	0.9806	0.9803	0.7310	0.9550	0.7580	0.7959
3			0.0000	0.9665	0.9737	0.9702	0.7245	0.9448	0.7674	0.7945
4				0.0000	0.9816	0.9801	0.7428	0.9572	0.7545	0.7914
5					0.0000	0.9830	0.7367	0.9568	0.7743	0.8122
6						0.0000	0.7401	0.9573	0.7733	0.8012
7							0.0000	0.7180	0.6750	0.6429
8								0.0000	0.7257	0.7706
9									0.0000	0.8050
10										0.0000

Fig. 3-14 Correlation matrix of ASM measure.

In the correlation matrix presented in Figure 3-14, built areas have high correlation values, clearly dividing the two most obvious spatial patterns. The most similar group is not easy to put together from this matrix. In fact, only area 6 and area 5 show great similarity with each other. However, the significance of the local operation is to represent local texture properties. As we see in the regional analysis and single feature analysis of ASM (Fig. 3-15), area 7 is grouped with area 9 and 10. In the local analysis, however, area 7 shows a low correlation with area 10 but relatively high correlation with other residential areas. Individual texture measures can be performed in this way, extracting texture properties of the subimage on a local base. Although the local operation has this advantages over regional analysis, it is not very applicable for quick texture analysis because it is computationally intensive.

	1	2	3	4	5	6	7	8	9	10
1	0.000	0.002	0.000	0.001	0.001	0.002	0.028	0.002	0.160	0.029
2	0.002	0.000	0.002	0.000	0.001	0.000	0.029	0.000	0.161	0.031
3	0.000	0.002	0.000	0.001	0.001	0.002	0.028	0.002	0.160	0.029
4	0.001	0.000	0.001	0.000	0.000	0.001	0.029	0.000	0.161	0.030
5	0.001	0.001	0.001	0.000	0.000	0.001	0.029	0.001	0.161	0.030
6	0.002	0.000	0.002	0.001	0.001	0.000	0.029	0.000	0.161	0.031
7	0.028	0.029	0.028	0.029	0.029	0.029	0.000	0.029	0.132	0.001
8	0.002	0.000	0.002	0.000	0.001	0.000	0.029	0.000	0.161	0.031
9	0.160	0.161	0.160	0.161	0.161	0.161	0.132	0.161	0.000	0.131
10	0.029	0.031	0.029	0.030	0.030	0.031	0.001	0.031	0.131	0.000

Fig. 3-15 Distance measure of ASM (mean).

C. Fourier Spectrum Pattern of Texture and the Entropy-based Textural Analysis

The texture information extracted by the SPADEP approach is difficult to visualize. Sometimes it is useful to combine the computer analysis with human interpretation. One approach to visualize the texture properties of an image is the display of the Fourier spectrum. Using the amplitude of the frequencies as an intensity function, we can produce either a gray-scale image or a 3D plot of the Fourier spectrum. Figure 3-18 to 3-27 are the 3D plots of Fourier spectrum for the ten study areas. Figure 3-28 are the images of the Fourier spectrum for the corresponding areas. To analyze these patterns, the entropy-based analysis is accompanied with these patterns.

By visual interpretation, we can divide the Fourier spectrum displays into two groups. One group, including area 1, 7, 9, 10, has high frequencies concentrated at the center. The other group, including areas 2, 3, 4, 5, 6, 8, has frequencies around the outer edge. In fact, both areas 9 and 10 are non-built, area 7 is partially developed while area 1 is low density residential area. Thus, at first glance, we can easily distinguish regular residential areas from the low density housing and the non-built areas.

However, what do these patterns tell us about the texture properties of the areas? One way to understand these patterns is to relate these patterns with the frequency distributions (regional entropy measure). Figure 3-29 is the line chart of the regional entropy for all the ten areas. The 32 X 32 Fourier spectrum is divided into 25X25, 17X17, 11X11, 5X5 (Fig. 3-30) subregions. The

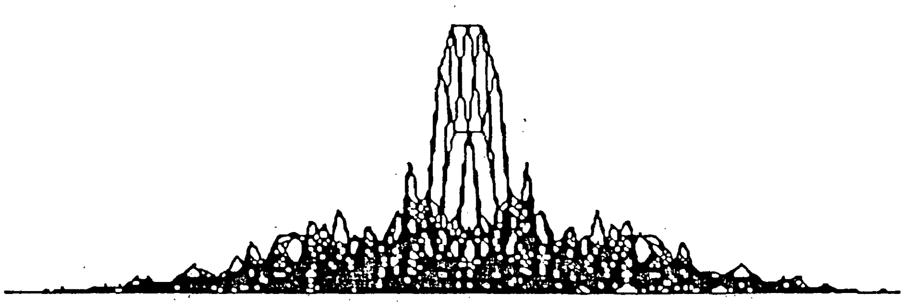


Fig. 3-18 Fourier Spectrum of area 1.

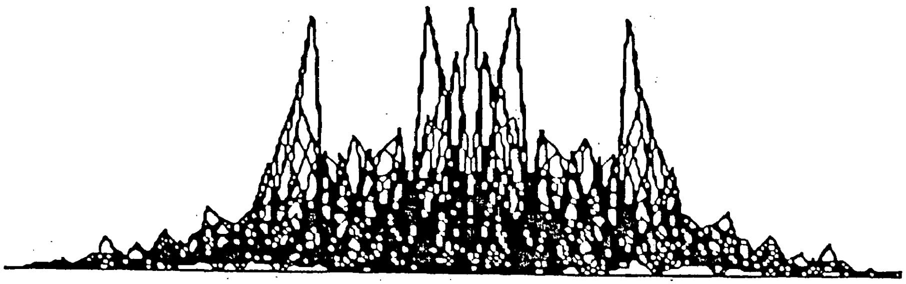


Fig. 3-19 Fourier Spectrum of area 2.

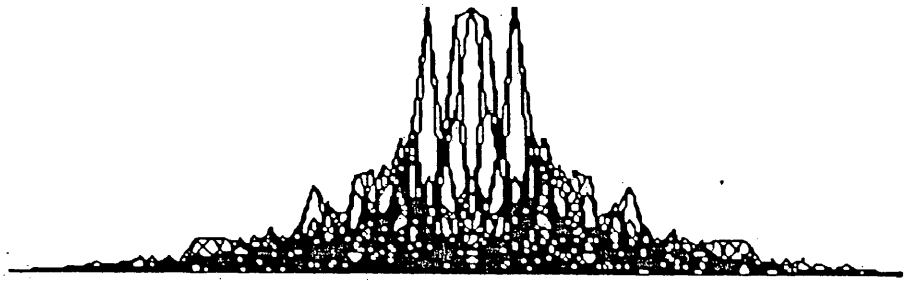


Fig. 3-20 Fourier Spectrum of area 3.

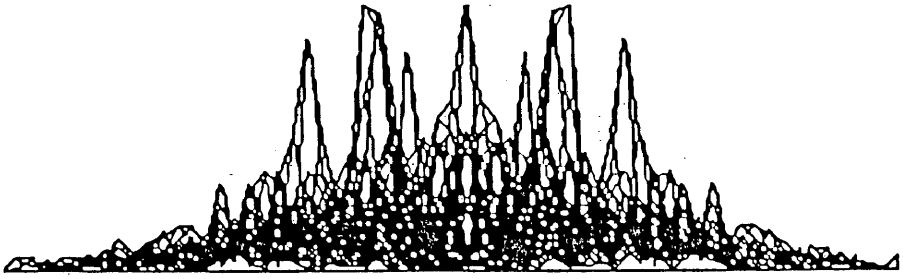


Fig. 3-21 Fourier Spectrum of area 4.

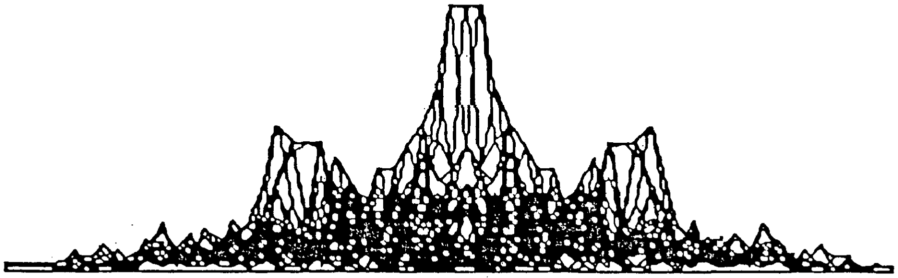


Fig. 3-22 Fourier Spectrum of area 5.

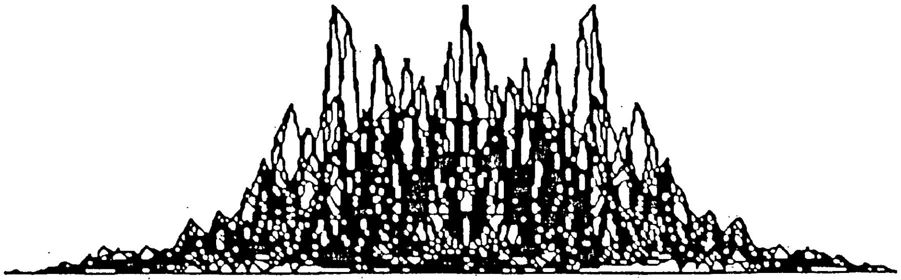


Fig. 3-23 Fourier Spectrum of area 6.

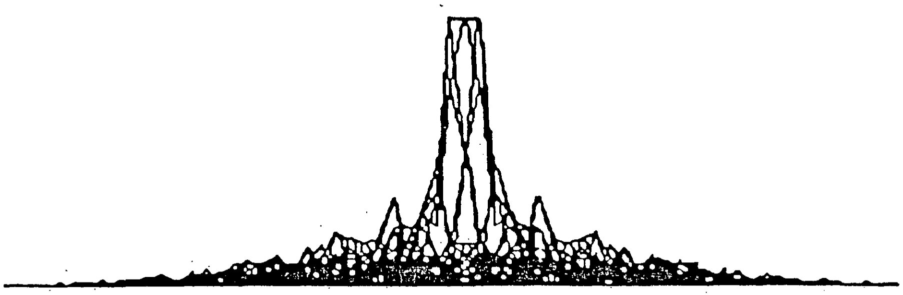


Fig. 3-24 Fourier Spectrum of area 7.

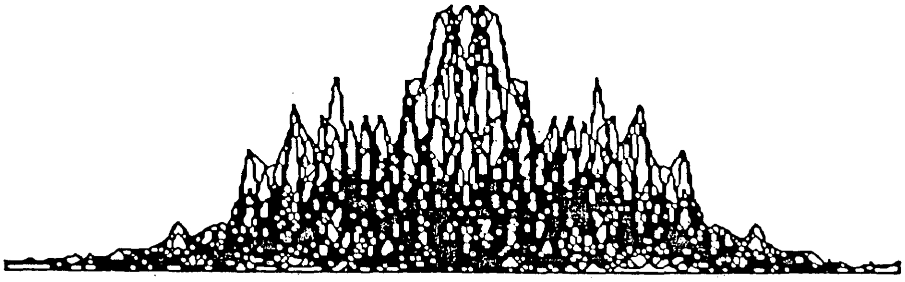


Fig. 3-25 Fourier Spectrum of area 8.

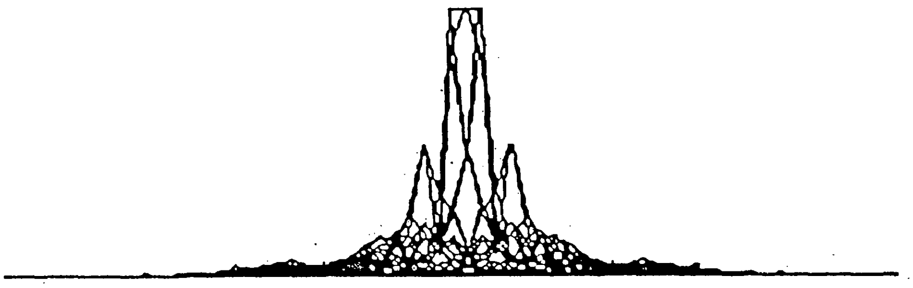


Fig. 3-26 Fourier Spectrum of area 9.

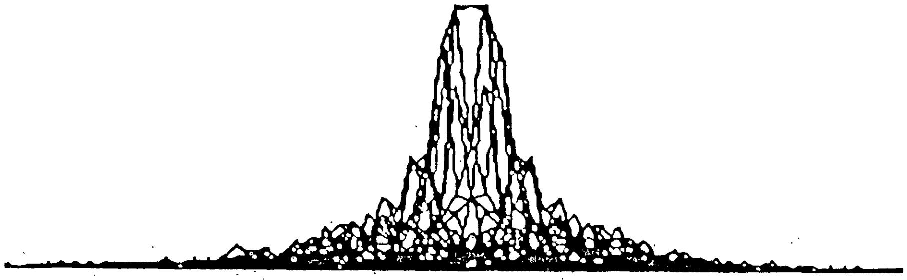


Fig. 3-27 Fourier Spectrum of area 10.

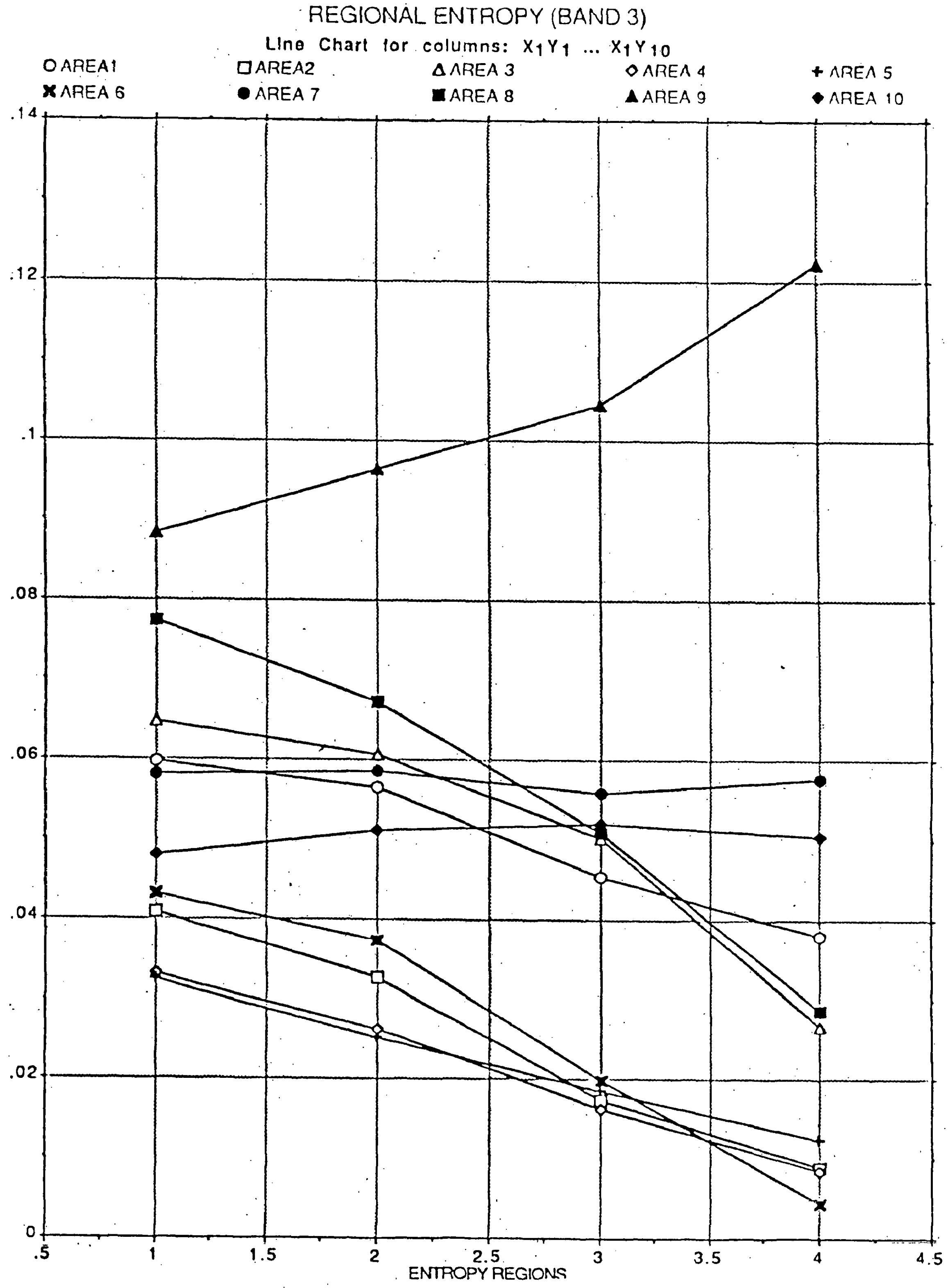


Fig. 3-29 Regional Entropy measures in band 3.

characteristics of regional entropy measures are highly related to the spectrum patterns we plot. Such measures can be evaluated on the absolute or relative base. On one hand, textures with more highly structural spatial distributions yield a low entropy value, while textures with random distributions yield a high value (D'Astous, et. al., 1983). On the other hand, the variations of entropy from one region to another reflect the characteristics of the spectral frequency distribution. Since it is the characteristics of the frequency distribution that influence the spectral pattern, the absolute values of these measures are less significant; thus, our focus is on the variations in the entropy in different regions of the power spectrum.

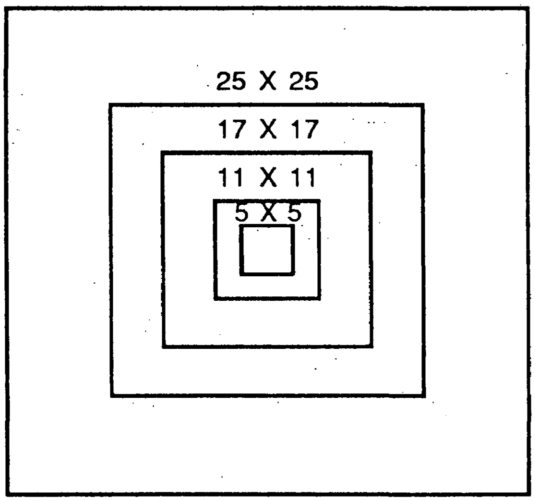


Fig. 30 The spectrum regions used to measure the entropy.

Let's observe the regional measures for areas in group one (Fig. 3-29). It is found that, for area 7 and 10, entropy in the four regions varies slightly. Area 9 stands out with its overall high entropy and the inverse distribution (low entropy in the larger region, high entropy in the center). The inverse and even distribution indicate that frequency components are evenly distributed in the small region around the origin. When spectral frequencies concentrate in a few components away from origin, the entropy in the wider areas become larger. This is a transition from 10, 9, 7, to 1 then to the other group. It is correspondent to a transition from farm land to clear subdivision, partially built area, residential area with very low density and the fully developed area.

Examining the residential group, comparing the slope of the entropy changes from one region to another, we may have better understanding of the spectrum patterns. First, we compare area 1 and area 3. The two patterns are similar except there are two peaks around the origin of the pattern for area 3. Accordingly, the two entropy curves are parallel through the 1, 2 and 3 spectrum regions. They then split away from each other through region 3 and 4. For area 3, entropy drops dramatically, indicating concentration of high frequency components -- that is the two peaks around the origin. High frequency peaks away from the origin seem to be related to the coarseness and linearity of the image. The more frequency peaks that occur within a spectrum region, the lower the entropy value, the more edges in different directions indicated. Areas 6, 2, 4, 8, 5 all present such characteristics.

It becomes obvious that similar texture patterns will have similar characteristics of frequency distribution and similar spectrum patterns; i.e., images with same texture patterns but different gray levels and orientations should have paralleled regional entropy curves. For example, area 6 and 8 have similar regional entropy patterns except that the slope is greater for area 6 from region 2 to region 4. This causes the peripheral high frequency peaks in 3D plot of area 6. Comparing the 3D plots of area 6, 2, 4, 8 and 5, we find that new housing area 6, 8 and 2, have similar patterns while older areas 4 and 5 are alike. When an area is gradually covered by vegetation, the coarseness will be reduced and high frequency components will be grouped together.

If we use the results of the distance measures from the SPADEP regional analysis, we find the Fourier spectrum patterns are nicely matched with those groups: (1, 3), (4, 5), (2, 6, 8), (7, 9 10). Compared with the SPADEP approach, Fourier analysis is fast and can be visualized. The amount of computation time is a function only of the size of the subimage area, whereas with SPADEP computation time is a function of both size and number of gray levels in the subimage. Moreover, no specific preprocessing of the image is required in Fourier analysis. We can extract texture information described by the regional entropy measures and display this with a 3D plot and the diffraction pattern for each of the interested subimage areas.

Since ground objects will have different reflectance characteristics in different spectral bands, features of a spatial pattern may have different textures in different bands. It takes only 10 seconds to calculate the Fourier spectrum for one 32 by 32 subimage area. This makes it an easy task to perform multi-spectral band texture analysis. The following is a brief presentation of this approach. Analyses are taken for the same subimage areas in TM band 1, 4, and 5. The regional entropy measures are illustrated in Figure 3-31 to 3-33. The distributions of regional entropies in different spectral bands seem similar between band 1 and band 3, but significant differences exist between band 4 and band 3, and band 5 and band 3. Further research is needed in order to find the relationships of texture properties in different spectral bands.

III. Summary

In this chapter, two approaches of texture analysis have been applied on a Landsat TM image of suburban Omaha. An emphasis is placed upon establishing the similarities of texture properties among different residential patterns. It is found that the differences between the developed areas and partially developed / non-developed areas are easily identified. Subtle similarities among different residential patterns are also identified to a certain extent either by the texture features from SPADEP approach or the regional entropy measures from the Fourier analysis. Considering the effectiveness of each measure, the regional entropy analysis in the spatial frequency domain is simpler, faster and produces a meaningful graphic display.

Chapter 4: Conclusion

The fundamental assumption of this thesis is that different spatial patterns on the earth surface are measurable in digital imagery. Texture analysis is used to acquire such measurements. The measurability of spatial pattern in a digital image would greatly enhance our understanding of certain spatial organizations.

A great deal of effort has been devoted in this thesis to selecting and developing approaches to texture analysis and the computer implementation of such techniques. Texture measures are treated as being representative of the characteristics of a spatial pattern. It has been shown that the properties of a spatial pattern in an image can be described in many ways, either with statistical measures or with graphic displays. Texture analysis can be performed in the spatial domain as well as in the frequency domain; each has its own advantages and shortcomings. It can also be implemented locally and regionally, with the former more desirable where homogeneous image areas are difficult to define.

It is found that different residential patterns do have different texture measures, thus different spatial characteristics. More specifically, the difference can be found among built-up areas vs. cleared subdivision, built vs. partially built areas, high density vs. low density residential areas, regular grid pattern vs. irregular cul-de-sac type patterns, partially developed areas vs. fully developed areas. The seven texture measures selected from the SPADEP approach clearly indicates that:

1) Fully developed residential areas have much less homogeneity than the partially developed or under-developed areas (cleared subdivision) while the differences of homogeneity among fully developed residential areas are very small;

2) Fully developed residential areas have much higher complexity than the partially developed vs. under-developed areas; low density residential areas present lower complexity; for the same area, increase of vegetation coverage will reduce the complexity measures on the image;

3) Spatial relationships of new residential areas are more irregular than that of the older areas due to the different street layout.

With texture analysis, spatial characteristics of a spatial pattern thus can be described in a set of new features, such as homogeneity, complexity, linearity, regularity, etc.

In the frequency domain, texture patterns are represented by their characteristics of frequency distributions in the Fourier series. The Fourier spectrum pattern provides a generalized representation of the texture properties of an image. Generally, uniform spatial patterns, such as cleared subdivisions, will have pyramidal Fourier spectrum patterns while the spectrum of newly built irregular residential areas have frequency peaks away from the origin. From a piece of farm land to a fully developed residential area, frequency distribution changed accordingly as represented with the Fourier spectrum patterns presented.

It is found that texture analysis with the Fourier spectrum, although criticized by people in the field of pattern recognition, is an attractive approach for the analysis of urban residential spatial patterns. The Fourier spectrum is one of the few ways to graphically display texture information. In the previous chapter, different spatial patterns were represented clearly by a series of 3 dimensional plots of the Fourier spectrum, along with the descriptive regional entropy measures.

Visualizing the texture properties of an image is very important since humans are capable of synthesizing a great deal of graphic information. Methods for texture analysis should make use of both human ability and the advantages of the computer. In fact, no thorough investigations on the relationships between texture properties and its Fourier spectrum have been done. However, it is felt that the Fourier analysis will be more promising for a man-machine texture analysis system.

It is believed that this thesis has provided an insight into the possibilities of applying texture analysis to the study of spatial patterns in an urban area, presenting a possible direction for urban remote sensing. Measuring residential patterns with texture analysis not only enhances our previous concepts of this spatial phenomena but also indicates some possible applications of digital image processing to urban planning. The fourteen statistical texture measures from SPADEP not only extract different aspects of the spatial relationships among ground objects but also give a set of criteria for land use classification. The Fourier analysis presents graphic displays of different residential patterns as well as the descriptive entropy measures. These may allow us to describe characteristics of residential patterns with a new set of terminology for further inquiries of the underlying social, political, and cultural processes.

However, further experiments on the techniques and applications of texture analysis to urban residential patterns and other spatial patterns in urban areas need to be carried out. Firstly, enhancement of the concept of measurability of spatial pattern in digital imagery is needed. This concept could have an important implication for geographic study. Secondly, texture analysis is only one way to perform the measurement of spatial patterns in a digital image. Further investigation and development of related analytical techniques are necessary. With better understanding of the existing texture analysis techniques, such as fractal analysis we should explore more possible approaches.

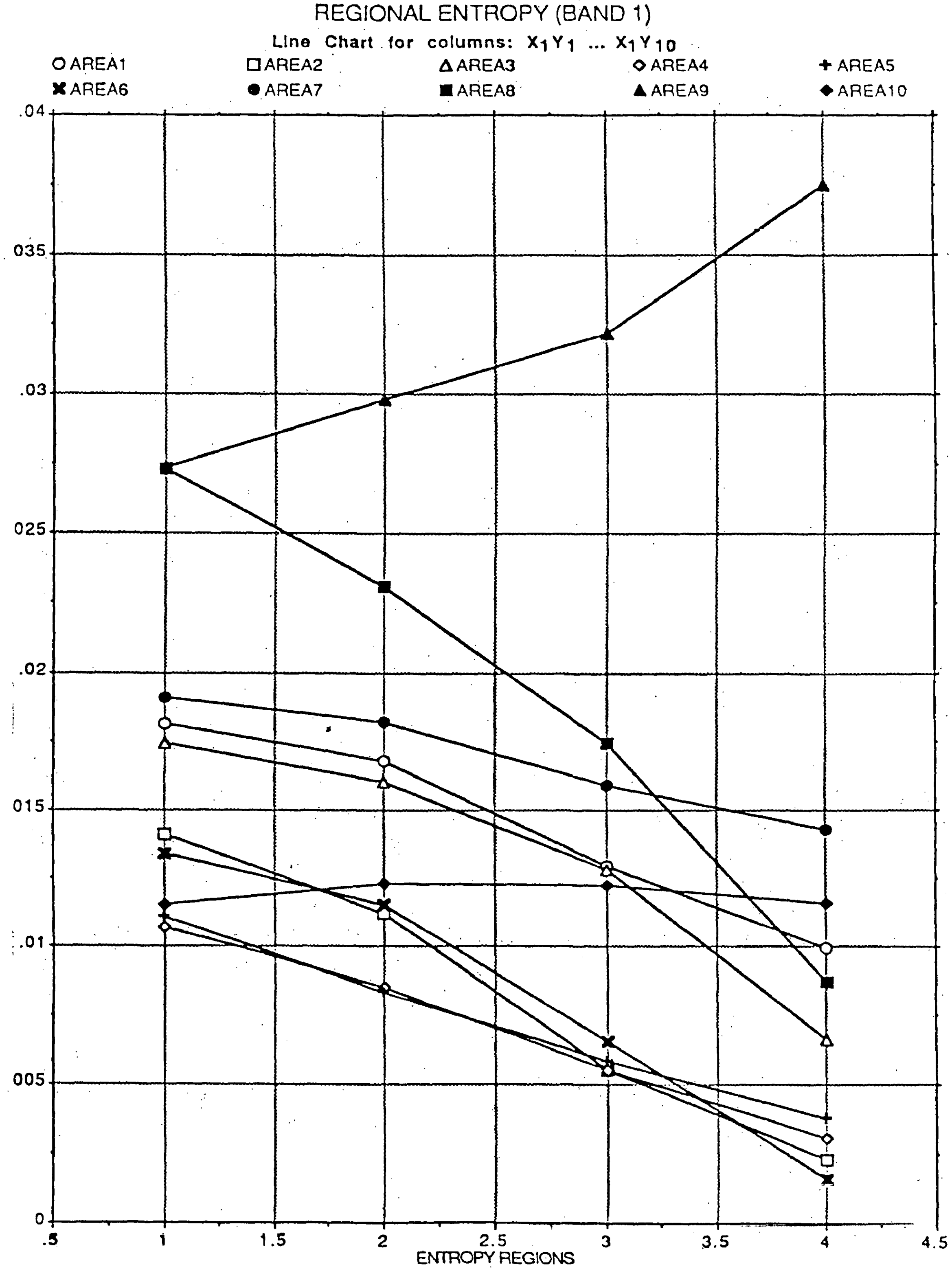


Fig. 3-3] Regional Entropy measures in band].

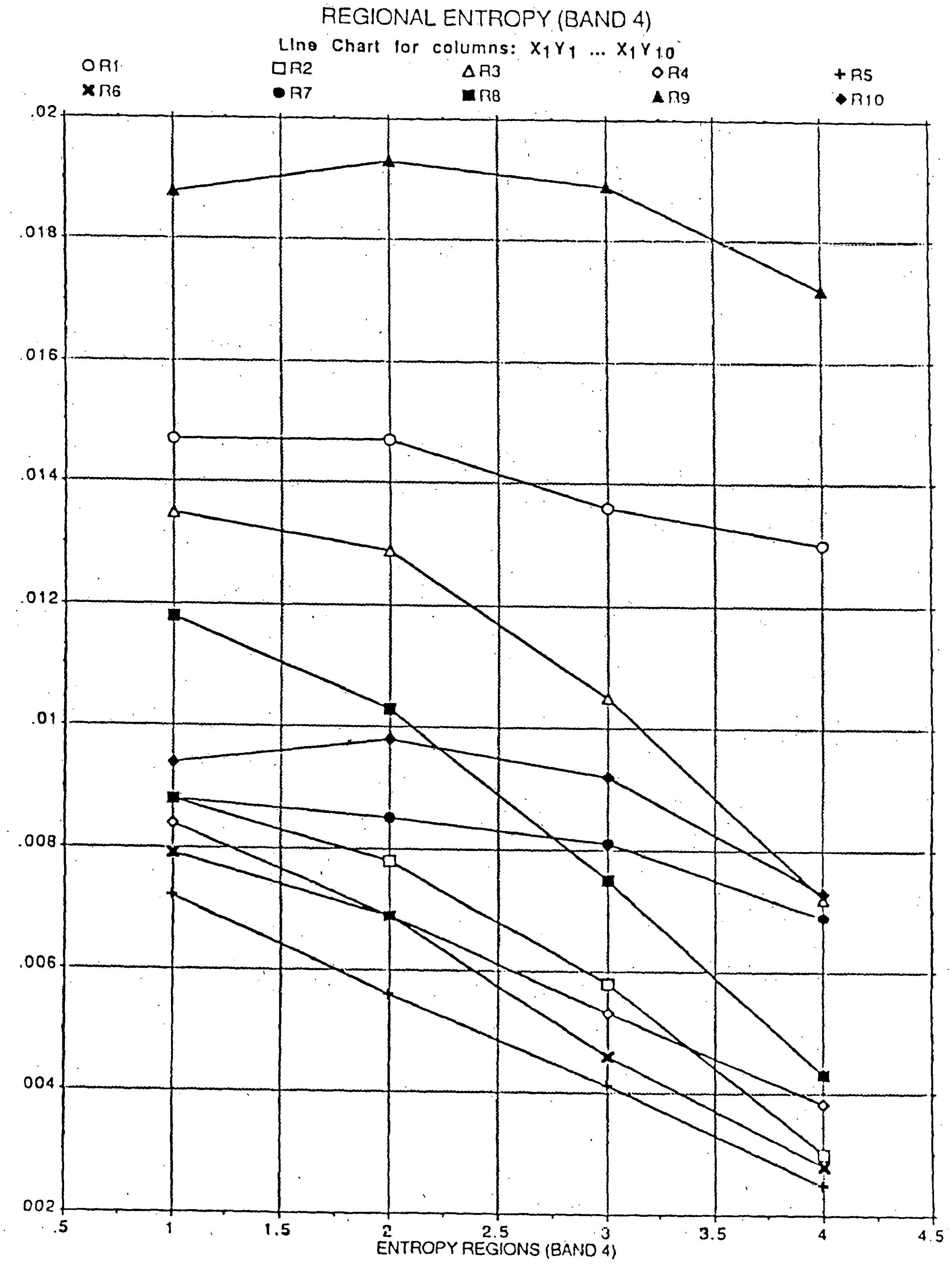


Fig. 3-32 Regional Entropy measures in band 4.

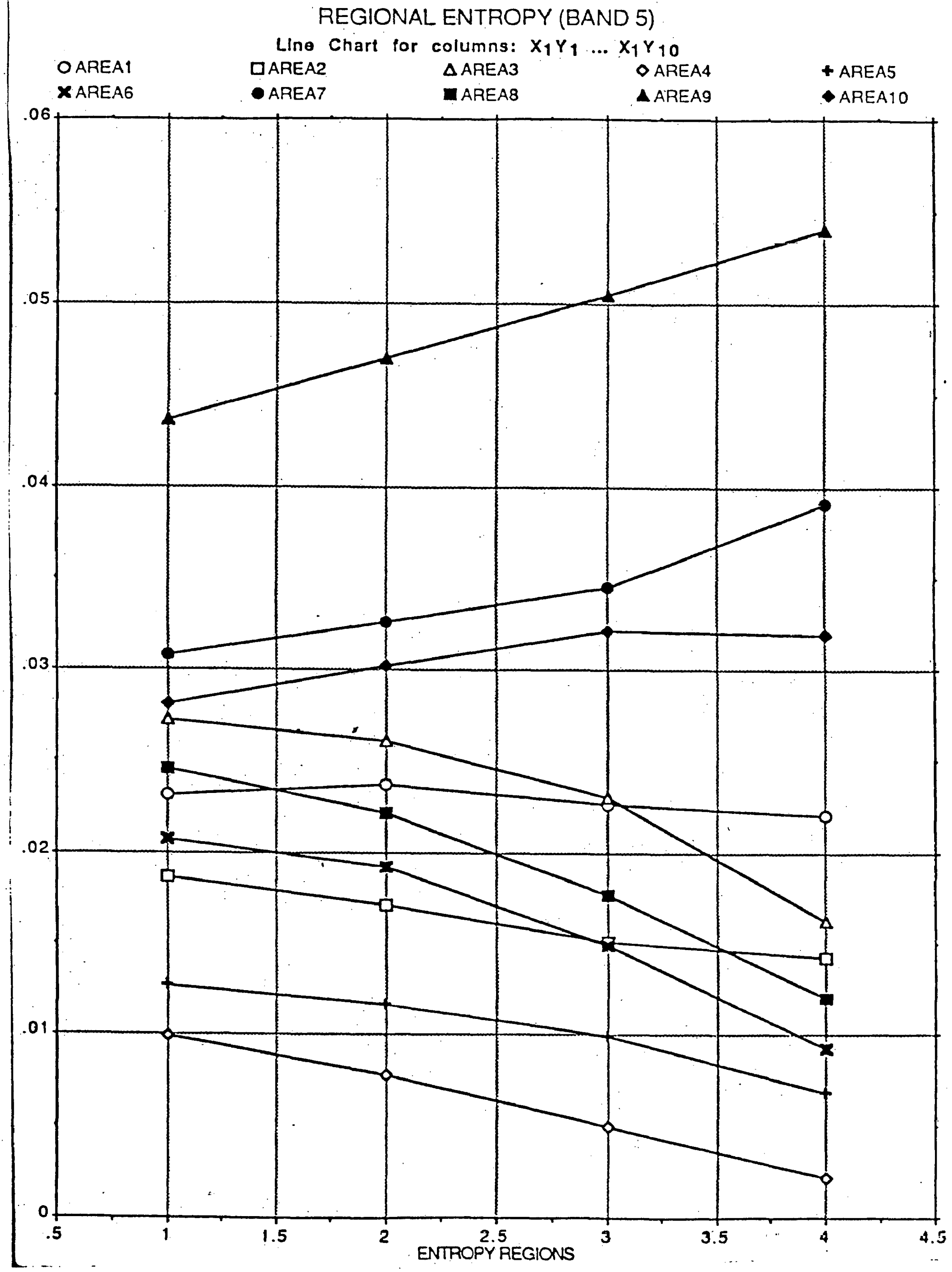


Fig. 3-33 Regional Entropy measures in band 5.

Appendices

```
      PROGRAM EGAL     !79710.5
C  HISTOGRAM EQUALIZATION ROUTINE.  EP78213.
C   F77 EGAL
C   LINK EGAL,SY:TSXLIB,SY:TVLIB,SY:F77LIB

      DIMENSION HIST(256),FX(257)
      INTEGER*4 IH(256)
      BYTE IMAGE(512,16)
      INTEGER*2 ION(39),IEXT(4),S(4,2),Q(256),KH(512)
      EQUIVALENCE (IH(1),KH(1))
      DATA S(1,1)/'G'/ S(1,2)/'W'/ NOCOL/16/ NG/16/
      DATA KH/512*0/ HIST/256*0.0/     !HISTOGRAM ARRAYS
C
      call mpiops      !tsx addition
      CALL SCREEN (IXMAX,IYMAX,IYVIS) !HARDWARE CONSTANTS.
      NOW = 256 * NOCOL        !BUFFER SIZE (WORDS)

C     NG IS NUMBER OF RESULTANT GRAY LEVELS (DEFAULT = 64).
C
C    REQUEST I/O FILE NAMES AT KEYBOARD, WITH SWITCHES.
  5   IF (ICSI(ION,IEXT,,S,2).NE.0) GO TO 5
      IF (S(2,1).EQ.2) NG = S(4,1)
      ICHANI = IGETC(I)               !ALLOCATE I/O CHANNELS.
      ICHANO = IGETC(I)
      IF ((ICHANI.LT.0).OR.(ICHANO.LT.0)) STOP 'NO CHANNEL'

C    LOAD DEVICE HANDLER IF REQUIRED.

      IF ((IFETCH(ION(1)).NE.0).OR.IFETCH(ION(16)).NE.0) STOP 'NO FETCH'
      IF (LOOKUP(ICHANO,ION(1)).LT.0) STOP 'OUTPUT FILE NOT FOUND.'
      IF (LOOKUP(ICHANI,ION(16)).LT.0) STOP 'INPUT FILE NOT FOUND'

      DO 25 I = 1,IXMAX,NOCOL !ACQUIRE GRAY-SCALE FREQUENCIES.STEP=NOCOL
         KBLK = I-1
      IF (IREADW(NOW,IMAGE,KBLK,ICHANI).LT.0) STOP 'READ FAULT'
         DO 20 K = 1,NOCOL
         CALL ALEPH (IMAGE(1,K),IYVIS,IH)
 20   CONTINUE
      DO 25 K = 1,256
         HIST(K) = HIST(K) + FLOAT(IH(K))
         IH(K) = 0
 25   CONTINUE

      FX(1) = 0.0
      DO 30 I = 1,256          !FIND CUMULATIVE DISTRIBUTION.
              FX(I+1) = FX(I) + HIST(I)
 30   CONTINUE

      WRITE(7,*) 'ENTER  GRAY LEVEL: '
      READ(5,*) NG
      N=NG
      IGRAY = N/NG
      PART = FLOAT(IYVIS)*FLOAT(IXMAX)/FLOAT(NG)
      LEVEL = 0
      K = 0
      SPART = PART     !SUM PARTS.
      DO 60 I = 1,256 !COUNT THRU TRANSFORM TABLE.
 40   IF ((ABS(FX(I)-SPART)).GE.(ABS(FX(I+1)-SPART))) GO TO 50
         LEVEL = LEVEL + IGRAY
         SPART = SPART + PART
CD    WRITE (7,1001) SPART
         GO TO 40
 50   Q(I) = K
 60   K = LEVEL

C  REPLACE INPUT PICTURE VIA LOOKUP TABLE.

      DO 90 I = 1,IXMAX,NOCOL
         KBLK = I-1
      IF (IREADW(NOW,IMAGE,KBLK,ICHANI).LT.0) STOP 'READ FAULT 2'
         DO 80 K = 1,NOCOL
         DO 80 J = 1,IYVIS
            M = IMAGE(J,K)
            M = (M.AND.255) + 1
 80      IMAGE(J,K) = Q(M)
      IF (IWRITW(NOW,IMAGE,KBLK,ICHANO).LT.0) STOP 'WRITE FAULT'
 90   CONTINUE
      CALL EXIT
1000  FORMAT (20I4)
1001  FORMAT (8(1XF9.1))
      STOP
      END
```

```
C
C------------------------------------------------------------
      PROGRAM TXALL
C
C     CALCULATE ALL TEXTURE MEASURES FOR SPECIFIED SUBIMAGE.
C
C     JULY, 24, 1987.
C
C------------------------------------------------------------
C
      PARAMETER L=32,L2=64,IWSIZE=32
      INTEGER DATA(IWSIZE,IWSIZE)
      DIMENSION P1(L,L),P2(L,L),P3(L,L),P4(L,L)
      DIMENSION P1x(L),P2x(L),P3x(L),P4x(L),
     +          P1y(L),P2y(L),P3y(L),P4y(L)
      DIMENSION P1xy(L2),P2xy(L2),P3xy(L2),P4xy(L2),
     +          P1yx(L),P2yx(L),P3yx(L),P4yx(L)
      CHARACTER*10 FORM1,FORM2,OUTFL
      DATA FORM1/'(####I4)'/FORM2/'(''$'',A)'/

      WRITE(7,FORM2) 'ENTER OUTPUT FILE NAME : '
      READ (5,'(A)') OUTFL

      OPEN(UNIT=1,FILE=OUTFL,STATUS='NEW')

      IXY=IWSIZE*IWSIZE
      WRITE(FORM1(2:5),'(I4)') IWSIZE

      WRITE(7,FORM2) 'STARTING POSITION (XO,YO) = '
      READ(5,*) JO,IO
      WRITE(7,FORM2) 'GRAY LEVEL (=<32) =  '
      READ(5,*) N
      WRITE(7,FORM2) 'OUTPUT UNIT = '
      READ(5,*) U
C
      CALL OPEN(1,'P0',2)
C
      T1=SECNDS(0.0)
C
              CALL INPUT (1,DATA,JO,IWSIZE,IO,IWSIZE)
              CALL UNPACK (DATA,DATA,IXY)
C             WRITE (U,102) JO,IO
C             WRITE (U,FORM1) ((DATA(I,J),J=1,IWSIZE),I=1,IWSIZE)
C
      CALL SPADEP (DATA,N,IWSIZE,IWSIZE,P1,P2,P3,P4)
      CALL RCOL1(P1,P2,P3,P4,N,P1X,P2X,P3X,P4X,P1Y,P2Y,P3Y,P4Y)
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
C
              CALL F1(P1,N,ASM1)
              CALL F1(P2,N,ASM2)
              CALL F1(P3,N,ASM3)
              CALL F1(P4,N,ASM4)
      CALL MAXMIN(ASM1,ASM2,ASM3,ASM4,RANGE,AVERAG,DEVI)
      WRITE(U,200) ASM1,ASM2,ASM3,ASM4,RANGE,AVERAG,DEVI

              CALL F2(P1YX,N,CTR1)
              CALL F2(P2YX,N,CTR2)
              CALL F2(P3YX,N,CTR3)
              CALL F2(P4YX,N,CTR4)
      CALL MAXMIN(CTR1,CTR2,CTR3,CTR4,RANGE,AVERAG,DEVI)
      WRITE(U,200) CTR1,CTR2,CTR3,CTR4, RANGE, AVERAG,DEVI

              CALL F3(P1,N,P1X,P1Y,CORRE1)
              CALL F3(P2,N,P2X,P2Y,CORRE2)
              CALL F3(P3,N,P3X,P3Y,CORRE3)
              CALL F3(P4,N,P4X,P4Y,CORRE4)
      CALL MAXMIN(CORRE1,CORRE2,CORRE3,CORRE4,RANGE,AVERAG,DEVI)
      WRITE(U,200) CORRE1,CORRE2,CORRE3,CORRE4,RANGE,AVERAG,DEVI

              CALL F4(P1,N,SUMSQ1)
              CALL F4(P2,N,SUMSQ2)
              CALL F4(P3,N,SUMSQ3)
              CALL F4(P4,N,SUMSQ4)
              CALL MAXMIN(SUMSQ1,SUMSQ2,SUMSQ3,SUMSQ4,RANGE,AVERAG,DEVI)
      WRITE(U,200) SUMSQ1/1000.,SUMSQ2/1000.,SUMSQ3/1000.,
     +           SUMSQ4/1000.,RANGE/1000.,AVERAG/1000.,DEVI/1000.

              CALL F5(P1,N,FIDM1)
              CALL F5(P2,N,FIDM2)
              CALL F5(P3,N,FIDM3)
              CALL F5(P4,N,FIDM4)
              CALL MAXMIN(FIDM1,FIDM2,FIDM3,FIDM4,RANGE,AVERAG,DEVI)
      WRITE(U,200) FIDM1,FIDM2,FIDM3,FIDM4,RANGE,AVERAG,DEVI
```

```
                CALL F6(P1XY,N,SUMAV1)
                CALL F6(P2XY,N,SUMAV2)
                CALL F6(P3XY,N,SUMAV3)
                CALL F6(P4XY,N,SUMAV4)
                CALL MAXMIN(SUMAV1,SUMAV2,SUMAV3,SUMAV4,RANGE,AVERAG,DEVI)
        WRITE(U,200) SUMAV1,SUMAV2,SUMAV3,SUMAV4,RANGE,AVERAG,DEVI

                CALL F7(P1XY,N,SUMET1)
                CALL F7(P2XY,N,SUMET2)
                CALL F7(P3XY,N,SUMET3)
                CALL F7(P4XY,N,SUMET4)
        CALL MAXMIN(SUMET1,SUMET2,SUMET3,SUMET4,RANGE,AVERAG,DEVI)
        WRITE(U,200) SUMET1,SUMET2,SUMET3,SUMET4,RANGE,AVERAG,DEVI

                CALL F8(P1XY,N,SUMET1,SUMV1)
                CALL F8(P2XY,N,SUMET2,SUMV2)
                CALL F8(P3XY,N,SUMET3,SUMV3)
                CALL F8(P4XY,N,SUMET4,SUMV4)
        CALL MAXMIN(SUMV1,SUMV2,SUMV3,SUMV4,RANGE,AVERAG,DEVI)
        WRITE(U,200) SUMV1,SUMV2,SUMV3,SUMV4,RANGE,AVERAG,DEVI

                CALL F9(P1,N,ENTRP1)
                CALL F9(P2,N,ENTRP2)
                CALL F9(P3,N,ENTRP3)
                CALL F9(P4,N,ENTRP4)
        CALL MAXMIN(ENTRP1,ENTRP2,ENTRP3,ENTRP4,RANGE,AVERAG,DEVI)
        WRITE(U,200) ENTRP1,ENTRP2,ENTRP3,ENTRP4,RANGE,AVERAG,DEVI
C
                CALL F10(P1YX,N,DIFET1)
                CALL F10(P2YX,N,DIFET2)
                CALL F10(P3YX,N,DIFET3)
                CALL F10(P4YX,N,DIFET4)
        CALL MAXMIN(DIFET1,DIFET2,DIFET3,DIFET4,RANGE,AVERAG,DEVI)
        WRITE(U,200) DIFET1,DIFET2,DIFET3,DIFET4,RANGE,AVERAG,DEVI
C
                CALL F11(P1YX,N,DIFET1,DIFV1)
                CALL F11(P2YX,N,DIFET2,DIFV2)
                CALL F11(P3YX,N,DIFET3,DIFV3)
                CALL F11(P4YX,N,DIFET4,DIFV4)
        CALL MAXMIN(DIFV1,DIFV2,DIFV3,DIFV4,RANGE,AVERAG,DEVI)
        WRITE(U,200) DIFV1,DIFV2,DIFV3,DIFV4,RANGE,AVERAG,DEVI
C
                CALL F12(P1,P1X,P1Y,N,ENTRP1,FIMC11,FIMC21)
                CALL F12(P2,P2X,P2Y,N,ENTRP2,FIMC12,FIMC22)
                CALL F12(P3,P3X,P3Y,N,ENTRP3,FIMC13,FIMC23)
                CALL F12(P4,P4X,P4Y,N,ENTRP4,FIMC14,FIMC24)
C
        CALL MAXMIN(FIMC11,FIMC12,FIMC13,FIMC14,RANGE,AVERAG,DEVI)
        WRITE(U,200) FIMC11,FIMC12,FIMC13,FIMC14,RANGE,AVERAG,DEVI
C
        CALL MAXMIN(FIMC21,FIMC22,FIMC23,FIMC24,RANGE,AVERAG,DEVI)
        WRITE(U,200) FIMC21,FIMC22,FIMC23,FIMC24,RANGE,AVERAG,DEVI
C
                CALL F13(P1,P1X,P1Y,N,FMCC1)
                CALL F13(P2,P2X,P2Y,N,FMCC2)
                CALL F13(P3,P3X,P3Y,N,FMCC3)
                CALL F13(P4,P4X,P4Y,N,FMCC4)
        CALL MAXMIN(FMCC1,FMCC2,FMCC3,FMCC4,RANGE,AVERAG,DEVI)
        WRITE(U,200) FMCC1,FMCC2,FMCC3,FMCC4,RANGE,AVERAG,DEVI
C
C
200     FORMAT(1X,7F10.4)
C
        TYPE*,'TIME USED = ',SECNDS(T1)

        STOP
        END
C
C-------------------------------------------------
        SUBROUTINE MAXMIN(A,B,C,D,MIMA,AVG,DEVI)
C
C       MIMA: RANGE OF (A,B,C,D);
C       AVG: MEAN OF (A,B,C,D);
C       DEVI: VARIANCE OF (A,B,C,D).
C
C-------------------------------------------------
C
        REAL MIMA
C
        MIMA=0.0
        AVG=0.0
        DEVI=0.0
        SUM=0.0
```

```
C
      AVG=(A+B+C+D)/4
      MIMA=AMAX1(A,B,C,D)-AMIN1(A,B,C,D)
      SUM=(A-AVG)**2+(B-AVG)**2+(C-AVG)**2+(D-AVG)**2
      DEVI=SUM/4

      RETURN
      END
```

```
C
C----------------------------------------------------------------
      PROGRAM TXLOCL
C
C     LOCAL OPERATION OF SPADEP
C     CURRENT PROGRAM WORK WITH 32 BY 32 SUBIMAGE AND 3X3 WINDOW.
C-----------------------------------------------------------------
C
      INTEGER DUMMY(32,32), XO,YO,WINSIZ,OPT
      CHARACTER*10 FORM
      DATA FORM/'(''$'',A)'/

      WRITE(7,FORM) 'DEFINE THE IMAGE AREA (X,Y) = '
      READ(5,*) ICOL,IROW
      WRITE(7,FORM) 'STARTING POSITION (XO,YO) = '
      READ(5,*) XO,YO
      WRITE(7,FORM) 'WINDOW SIZE = '
      READ(5,*) WINSIZ
      WRITE(7,FORM) 'GRAY LEVEL (=<32) =  '
      READ(5,*) NGRAY
      WRITE(7,*)
      WRITE(7,*) 'TEXTURE FUNCTIONS CAN BE EXTRACTED: '
      WRITE(7,*)'  1. ANGULAR SECOND MOMENT;'
      WRITE(7,*)'  2. CONTRAST;'
      WRITE(7,*)'  3. CORRELATION;'
      WRITE(7,*)'  4. SUM OF SQUARES;'
      WRITE(7,*)'  5. INVERSE DIFFERENCE MOMENT;'
      WRITE(7,*)'  6. SUM AVERAGE;'
      WRITE(7,*)'  7. SUM ENTROPY;'
      WRITE(7,*)'  8. SUM VARIANCE;'
      WRITE(7,*)'  9. ENTROPY;'
      WRITE(7,*)' 10. DIFFERENCE ENTROPY;'
      WRITE(7,*)' 11. DIFFERENCE VARIANCE;'
      WRITE(7,*)' 12. INFORMATION MEASURES OF CORRELATION;'
      WRITE(7,*)' 13. MAXIMAL CORRELATION COEFFICIENT;'
      WRITE(7,*)
      WRITE(7,FORM) 'ENTER YOUR CHOICE (0 TO QUIT): '
      READ(5,*) OPT
      IF(OPT.EQ.0) STOP
C
      CALL DOING (ICOL,IROW,XO,YO,WINSIZ,DUMMY,NGRAY,OPT)
C
      STOP
      END
C
C-------------------------------------------------------
      SUBROUTINE DOING(IC,IR,JO,IO,IWSIZE,DATA,N,IOPT)
C-------------------------------------------------------
C
      PARAMETER L=32,L2=64
      INTEGER DATA(IWSIZE,IWSIZE)
      DIMENSION P1(L,L),P2(L,L),P3(L,L),P4(L,L)
      DIMENSION P1x(L),P2x(L),P3x(L),P4x(L),
     +          P1y(L),P2y(L),P3y(L),P4y(L)
      DIMENSION P1xy(L2),P2xy(L2),P3xy(L2),P4xy(L2),
     +          P1yx(L),P2yx(L),P3yx(L),P4yx(L)
      DIMENSION TX1(30),TX2(30)
      CHARACTER*10 FORM,OUTFL
      DATA FORM/'(####I4)'/
C
      IXY=IWSIZE*IWSIZE
      WRITE(FORM(2:5),'(I4)') IWSIZE
C
      WRITE(7,*)'ENTER OUTPUT FILE NAME : '
      READ(5,'(A)') OUTFL
      OPEN (UNIT=2,FILE=OUTFL,STATUS='NEW',FORM='UNFORMATTED')
C
      CALL OPEN(1,'PO',2)
C
      T1=SECNDS(0.0)
      DO 10 IY=IO,IR+IO-3
        ICOUNT=0
        DO 30 IX=JO,IC+JO-3
              CALL INPUT (1,DATA,IX,IWSIZE,IY,IWSIZE)
              CALL UNPACK (DATA,DATA,IXY)
C             WRITE (7,102) IX,IY
C             WRITE (7,FORM) ((DATA(I,J),J=1,IWSIZE),I=1,IWSIZE)
C
      CALL SPADEP (DATA,N,IWSIZE,IWSIZE,P1,P2,P3,P4)
      IF(IOPT.EQ.1) THEN
              CALL F1(P1,N,ASM1)
              CALL F1(P2,N,ASM2)
              CALL F1(P3,N,ASM3)
              CALL F1(P4,N,ASM4)
              CALL MAXMIN(ASM1,ASM2,ASM3,ASM4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.2) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F2(P1YX,N,CTR1)
              CALL F2(P2YX,N,CTR2)
              CALL F2(P3YX,N,CTR3)
              CALL F2(P4YX,N,CTR4)
              CALL MAXMIN(CTR1,CTR2,CTR3,CTR4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.3) THEN
      CALL RCOL1(P1,P2,P3,P4,N,P1X,P2X,P3X,P4X,P1Y,P2Y,P3Y,P4Y)
              CALL F3(P1,N,P1X,P1Y,CORRE1)
              CALL F3(P2,N,P2X,P2Y,CORRE2)
              CALL F3(P3,N,P3X,P3Y,CORRE3)
              CALL F3(P4,N,P4X,P4Y,CORRE4)
              CALL MAXMIN(CORRE1,CORRE2,CORRE3,CORRE4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.4) THEN
              CALL F4(P1,N,SUMSQ1)
              CALL F4(P2,N,SUMSQ2)
              CALL F4(P3,N,SUMSQ3)
              CALL F4(P4,N,SUMSQ4)
              CALL MAXMIN(SUMSQ1,SUMSQ2,SUMSQ3,SUMSQ4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.5) THEN
              CALL F5(P1,N,FIDM1)
              CALL F5(P2,N,FIDM2)
              CALL F5(P3,N,FIDM3)
              CALL F5(P4,N,FIDM4)
              CALL MAXMIN(FIDM1,FIDM2,FIDM3,FIDM4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.6) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F6(P1XY,N,SUMAV1)
              CALL F6(P2XY,N,SUMAV2)
              CALL F6(P3XY,N,SUMAV3)
              CALL F6(P4XY,N,SUMAV4)
              CALL MAXMIN(SUMAV1,SUMAV2,SUMAV3,SUMAV4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.7) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F7(P1XY,N,SUMET1)
              CALL F7(P2XY,N,SUMET2)
              CALL F7(P3XY,N,SUMET3)
              CALL F7(P4XY,N,SUMET4)
              CALL MAXMIN(SUMET1,SUMET2,SUMET3,SUMET4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.8) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F7(P1XY,N,SUMET1)
              CALL F7(P2XY,N,SUMET2)
              CALL F7(P3XY,N,SUMET3)
              CALL F7(P4XY,N,SUMET4)
              CALL F8(P1XY,N,SUMET1,SUMV1)
              CALL F8(P2XY,N,SUMET2,SUMV2)
              CALL F8(P3XY,N,SUMET3,SUMV3)
              CALL F8(P4XY,N,SUMET4,SUMV4)
              CALL MAXMIN(SUMV1,SUMV2,SUMV3,SUMV4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.9) THEN
              CALL F9(P1,N,ENTRP1)
              CALL F9(P2,N,ENTRP2)
              CALL F9(P3,N,ENTRP3)
              CALL F9(P4,N,ENTRP4)
              CALL MAXMIN(ENTRP1,ENTRP2,ENTRP3,ENTRP4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.10) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F10(P1YX,N,DIFET1)
              CALL MAXMIN(ASM1,ASM2,ASM3,ASM4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.2) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
   +           P3YX,P4YX)
              CALL F2(P1YX,N,CTR1)
              CALL F2(P2YX,N,CTR2)
              CALL F2(P3YX,N,CTR3)
              CALL F2(P4YX,N,CTR4)
              CALL MAXMIN(CTR1,CTR2,CTR3,CTR4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.3) THEN
      CALL RCOL1(P1,P2,P3,P4,N,P1X,P2X,P3X,P4X,P1Y,P2Y,P3Y,P4Y)
              CALL F3(P1,N,P1X,P1Y,CORRE1)
              CALL F3(P2,N,P2X,P2Y,CORRE2)
              CALL F3(P3,N,P3X,P3Y,CORRE3)
              CALL F3(P4,N,P4X,P4Y,CORRE4)
              CALL MAXMIN(CORRE1,CORRE2,CORRE3,CORRE4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.4) THEN
              CALL F4(P1,N,SUMSQ1)
              CALL F4(P2,N,SUMSQ2)
              CALL F4(P3,N,SUMSQ3)
              CALL F4(P4,N,SUMSQ4)
              CALL MAXMIN(SUMSQ1,SUMSQ2,SUMSQ3,SUMSQ4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.5) THEN
              CALL F5(P1,N,FIDM1)
              CALL F5(P2,N,FIDM2)
              CALL F5(P3,N,FIDM3)
              CALL F5(P4,N,FIDM4)
              CALL MAXMIN(FIDM1,FIDM2,FIDM3,FIDM4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.6) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
   +           P3YX,P4YX)
              CALL F6(P1XY,N,SUMAV1)
              CALL F6(P2XY,N,SUMAV2)
              CALL F6(P3XY,N,SUMAV3)
              CALL F6(P4XY,N,SUMAV4)
              CALL MAXMIN(SUMAV1,SUMAV2,SUMAV3,SUMAV4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.7) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
   +           P3YX,P4YX)
              CALL F7(P1XY,N,SUMET1)
              CALL F7(P2XY,N,SUMET2)
              CALL F7(P3XY,N,SUMET3)
              CALL F7(P4XY,N,SUMET4)
              CALL MAXMIN(SUMET1,SUMET2,SUMET3,SUMET4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.8) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
   +           P3YX,P4YX)
              CALL F7(P1XY,N,SUMET1)
              CALL F7(P2XY,N,SUMET2)
              CALL F7(P3XY,N,SUMET3)
              CALL F7(P4XY,N,SUMET4)
              CALL F8(P1XY,N,SUMET1,SUMV1)
              CALL F8(P2XY,N,SUMET2,SUMV2)
              CALL F8(P3XY,N,SUMET3,SUMV3)
              CALL F8(P4XY,N,SUMET4,SUMV4)
              CALL MAXMIN(SUMV1,SUMV2,SUMV3,SUMV4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.9) THEN
              CALL F9(P1,N,ENTRP1)
              CALL F9(P2,N,ENTRP2)
              CALL F9(P3,N,ENTRP3)
              CALL F9(P4,N,ENTRP4)
              CALL MAXMIN(ENTRP1,ENTRP2,ENTRP3,ENTRP4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.10) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
   +           P3YX,P4YX)
              CALL F10(P1YX,N,DIFET1)
              CALL MAXMIN(ASM1,ASM2,ASM3,ASM4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.2) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F2(P1YX,N,CTR1)
              CALL F2(P2YX,N,CTR2)
              CALL F2(P3YX,N,CTR3)
              CALL F2(P4YX,N,CTR4)
              CALL MAXMIN(CTR1,CTR2,CTR3,CTR4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.3) THEN
      CALL RCOL1(P1,P2,P3,P4,N,P1X,P2X,P3X,P4X,P1Y,P2Y,P3Y,P4Y)
              CALL F3(P1,N,P1X,P1Y,CORRE1)
              CALL F3(P2,N,P2X,P2Y,CORRE2)
              CALL F3(P3,N,P3X,P3Y,CORRE3)
              CALL F3(P4,N,P4X,P4Y,CORRE4)
              CALL MAXMIN(CORRE1,CORRE2,CORRE3,CORRE4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.4) THEN
              CALL F4(P1,N,SUMSQ1)
              CALL F4(P2,N,SUMSQ2)
              CALL F4(P3,N,SUMSQ3)
              CALL F4(P4,N,SUMSQ4)
              CALL MAXMIN(SUMSQ1,SUMSQ2,SUMSQ3,SUMSQ4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.5) THEN
              CALL F5(P1,N,FIDM1)
              CALL F5(P2,N,FIDM2)
              CALL F5(P3,N,FIDM3)
              CALL F5(P4,N,FIDM4)
              CALL MAXMIN(FIDM1,FIDM2,FIDM3,FIDM4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.6) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F6(P1XY,N,SUMAV1)
              CALL F6(P2XY,N,SUMAV2)
              CALL F6(P3XY,N,SUMAV3)
              CALL F6(P4XY,N,SUMAV4)
              CALL MAXMIN(SUMAV1,SUMAV2,SUMAV3,SUMAV4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.7) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F7(P1XY,N,SUMET1)
              CALL F7(P2XY,N,SUMET2)
              CALL F7(P3XY,N,SUMET3)
              CALL F7(P4XY,N,SUMET4)
              CALL MAXMIN(SUMET1,SUMET2,SUMET3,SUMET4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.8) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F7(P1XY,N,SUMET1)
              CALL F7(P2XY,N,SUMET2)
              CALL F7(P3XY,N,SUMET3)
              CALL F7(P4XY,N,SUMET4)
              CALL F8(P1XY,N,SUMET1,SUMV1)
              CALL F8(P2XY,N,SUMET2,SUMV2)
              CALL F8(P3XY,N,SUMET3,SUMV3)
              CALL F8(P4XY,N,SUMET4,SUMV4)
              CALL MAXMIN(SUMV1,SUMV2,SUMV3,SUMV4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.9) THEN
              CALL F9(P1,N,ENTRP1)
              CALL F9(P2,N,ENTRP2)
              CALL F9(P3,N,ENTRP3)
              CALL F9(P4,N,ENTRP4)
              CALL MAXMIN(ENTRP1,ENTRP2,ENTRP3,ENTRP4,RANGE,AVERAG)
      GOTO 20
      ENDIF
      IF(IOPT.EQ.10) THEN
      CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +           P3YX,P4YX)
              CALL F10(P1YX,N,DIFET1)
```

```
                CALL F10(P2YX,N,DIFET2)
                CALL F10(P3YX,N,DIFET3)
                CALL F10(P4YX,N,DIFET4)
                CALL MAXMIN(DIFET1,DIFET2,DIFET3,DIFET4,RANGE,AVERAG)
        GOTO 20
        ENDIF
        IF(IOPT.EQ.11) THEN
        CALL RCOL2(P1,P2,P3,P4,N,P1XY,P2XY,P3XY,P4XY,P1YX,P2YX,
     +             P3YX,P4YX)
                CALL F10(P1YX,N,DIFET1)
                CALL F10(P2YX,N,DIFET2)
                CALL F10(P3YX,N,DIFET3)
                CALL F10(P4YX,N,DIFET4)
                CALL F11(P1YX,N,DIFET1,DIFV1)
                CALL F11(P2YX,N,DIFET2,DIFV2)
                CALL F11(P3YX,N,DIFET3,DIFV3)
                CALL F11(P4YX,N,DIFET4,DIFV4)
                CALL MAXMIN(DIFV1,DIFV2,DIFV3,DIFV4,RANGE,AVERAG)
        GOTO 20
        ENDIF
        IF(IOPT.EQ.12) THEN
        CALL RCOL1(P1,P2,P3,P4,N,P1X,P2X,P3X,P4X,P1Y,P2Y,P3Y,P4Y)
                CALL F9(P1,N,ENTRP1)
                CALL F9(P2,N,ENTRP2)
                CALL F9(P3,N,ENTRP3)
                CALL F9(P4,N,ENTRP4)
                CALL F12(P1,P1X,P1Y,N,ENTRP1,FIMC11,FIMC21)
                CALL F12(P2,P2X,P2Y,N,ENTRP2,FIMC12,FIMC22)
                CALL F12(P3,P3X,P3Y,N,ENTRP3,FIMC13,FIMC23)
                CALL F12(P4,P4X,P4Y,N,ENTRP4,FIMC14,FIMC24)
                CALL MAXMIN(FIMC11,FIMC12,FIMC13,FIMC14,RANGE1,AVERA1)
                CALL MAXMIN(FIMC21,FIMC22,FIMC23,FIMC24,RANGE2,AVERA2)
C
        GOTO 20
        ENDIF
        IF(IOPT.EQ.13) THEN
        CALL RCOL1(P1,P2,P3,P4,N,P1X,P2X,P3X,P4X,P1Y,P2Y,P3Y,P4Y)
                CALL F13(P1,P1X,P1Y,N,FMCC1)
                CALL F13(P2,P2X,P2Y,N,FMCC2)
                CALL F13(P3,P3X,P3Y,N,FMCC3)
                CALL F13(P4,P4X,P4Y,N,FMCC4)
                CALL MAXMIN(FMCC1,FMCC2,FMCC3,FMCC4,RANGE,AVERAG)
        ENDIF
C
20              ICOUNT=ICOUNT+1
                TX1(ICOUNT)=RANGE
                TX2(ICOUNT)=AVERAG
30      CONTINUE
                WRITE(2)  (TX1(I),I=1,30)
                WRITE(2)  (TX2(I),I=1,30)
10      CONTINUE
C
200     FORMAT(F20.5)
102     FORMAT (/'    COLUMN',I4,',  ROW',I4)
C
        TYPE*,SECNDS(T1)
        RETURN
        END
C
C------------------------------------------------
        SUBROUTINE MAXMIN(A,B,C,D,MIMA,AVG)
C------------------------------------------------
C
        REAL MIMA

        AVG=(A+B+C+D)/4
        MIMA=AMAX1(A,B,C,D)-AMIN1(A,B,C,D)

        RETURN
        END
```

```
      SUBROUTINE SPADEP(SDATA,NG,IR,IC,SP1,SP2,SP3,SP4)
C
C-------------------------------------------------------------------
C
C     CREATE SPATIAL GRAY TONE DEPENDENCE MATRICES FOR
C     EACH OF THE FOUR DIRECTIONS: 0, 45, 90, 135.
C
C     INPUT: SDATA --- IMAGE DATA
C            NG --- GRAY LEVEL
C            IR --- NUMBER OF ROW
C            IC --- NUMBER OF COLUMN
C            SP* --- NORMALIZED SGTDM (SP1-0,SP2-90,SP3-45,SP4-135)
C
C--------------------------------------------------------------------
C
      PARAMETER LG=32
      DIMENSION SDATA(LG,LG),SP1(LG,LG),SP2(LG,LG),SP3(LG,LG),
     +          SP4(LG,LG),DI(4),DJ(4)
      INTEGER SDATA, CENT, NABOR, DI, DJ
      DATA DI/0, -1, -1, -1/ DJ/1, 0, 1, -1/
C
C....INITIALIZE SP*(I,J)
C
         DO 4 I=1,NG
            DO 4 J=1,NG
               SP1(I,J)=0.0
               SP2(I,J)=0.0
               SP3(I,J)=0.0
               SP4(I,J)=0.0
4     CONTINUE
C
C...CALCULATE SPADEP
C
      DO 20 I=1, IR
        DO 20 J=1, IC
              CENT=SDATA(I,J)
              DO 30 K=1,4
              II=I+DI(K)
              JJ=J+DJ(K)
      IF((II.GE.1.AND.II.LE.IR).AND.(JJ.GE.1.AND.JJ.LE.IC))THEN
              NABOR=SDATA(II,JJ)
              IF(K.EQ.1)  SP1(CENT+1,NABOR+1)=SP1(CENT+1,NABOR+1)+1
              IF(K.EQ.2)  SP2(CENT+1,NABOR+1)=SP2(CENT+1,NABOR+1)+1
              IF(K.EQ.3)  SP3(CENT+1,NABOR+1)=SP3(CENT+1,NABOR+1)+1
              IF(K.EQ.4)  SP4(CENT+1,NABOR+1)=SP4(CENT+1,NABOR+1)+1
      ENDIF
30    CONTINUE
20    CONTINUE
C
C...TRANSPOSE OF THE SP
C
      DO 40 I=1,NG
        DO 40 J=1,NG
          IF(J .GE. I) THEN
              SP1(I,J)=SP1(I,J)+SP1(J,I)
              SP1(J,I)=SP1(I,J)
              SP2(I,J)=SP2(I,J)+SP2(J,I)
              SP2(J,I)=SP2(I,J)
              SP3(I,J)=SP3(I,J)+SP3(J,I)
              SP3(J,I)=SP3(I,J)
              SP4(I,J)=SP4(I,J)+SP4(J,I)
              SP4(J,I)=SP4(I,J)
          ENDIF
40    CONTINUE
C
C...NOMALIZE SPADEP. R* ARE THE NUMBER OF NEIGNBORING RESOLUTION
C...CELL PAIRS USED IN COMPUTING A PARTICULAR SPADEP.
C
      R1=2*IR*(IC-1)            ! 0 DEGREE     ! ALL FOR DISTANCE=1.
      R2=2*IC*(IR-1)            ! 90 DEGREE
      R3=2*(IR-1)*(IC-1)                 ! 45 DEGREE
      R4=2*(IC-1)*(IR-1)                 ! 135 DEGREE
C
         DO 50 I=1,NG
           DO 50 J=1,NG
              SP1(I,J)=SP1(I,J)/R1
              SP2(I,J)=SP2(I,J)/R2
              SP3(I,J)=SP3(I,J)/R3
              SP4(I,J)=SP4(I,J)/R4
50       CONTINUE
C
      RETURN
      END
C
```

```
               SUM4=0.0
       DO 50 I=1,NG
         DO 50 J=1,NG
               IF((I+J) .EQ. (K+1)) THEN
               SUM1=SUM1+SP1(I,J)
               SUM2=SUM2+SP2(I,J)
               SUM3=SUM3+SP3(I,J)
               SUM4=SUM4+SP4(I,J)
               ENDIF
50     CONTINUE
               SP1xy(K)=SUM1
               SP2xy(K)=SUM2
               SP3xy(K)=SUM3
               SP4xy(K)=SUM4
45     CONTINUE
C
C...2) SPx-y(K)=SP*yx(K)
C
       DO 55 K=1,NG
               SUM1=0.0
               SUM2=0.0
               SUM3=0.0
               SUM4=0.0
         DO 60 I=1,NG
           DO 60 J=1,NG
               IF(ABS(I-J) .EQ. K-1) THEN
               SUM1=SUM1+SP1(I,J)
               SUM2=SUM2+SP2(I,J)
               SUM3=SUM3+SP3(I,J)
               SUM4=SUM4+SP4(I,J)
               ENDIF
60       CONTINUE
              SP1yx(K)=SUM1
              SP2yx(K)=SUM2
              SP3yx(K)=SUM3
              SP4yx(K)=SUM4
55     CONTINUE
C
       RETURN
       END
```

```
      SUBROUTINE RCOL1(SP1,SP2,SP3,SP4,NG,SP1x,SP2x,SP3x,SP4x,SP1y,
     +                   SP2y,SP3y,SP4y)
C
C-----------------------------------------------------------------------
C
C
C CALCULATE SPx(i)=SUM{(j-NG)}(SP(i,j)),  SPy(j)=SUM{(i-NG)}(SP(i,j))
C NOTE: WHEN i=j=NG, SPx(i)=SPy(j)
C
C
C       SPX(I) AND SPY(J) ARE THE MARGINAL PROBABILITY MATRIX.
C
C-----------------------------------------------------------------------
C
      PARAMETER LG=32
      DIMENSION SP1(LG,LG),SP2(LG,LG),SP3(LG,LG),SP4(LG,LG),
     +          SP1x(LG),SP2x(LG),SP3x(LG),SP4x(LG),SP1y(LG),
     +          SP2y(LG),SP3y(LG),SP4y(LG)
C
C...1) SPx(i)
C
      DO 15 I=1,NG
              SUM1=0.0
              SUM2=0.0
              SUM3=0.0
              SUM4=0.0
        DO 20 J=1,NG
              SUM1=SUM1+SP1(I,J)
              SUM2=SUM2+SP2(I,J)
              SUM3=SUM3+SP3(I,J)
              SUM4=SUM4+SP4(I,J)
20    CONTINUE
              SP1x(I)=SUM1
              SP2x(I)=SUM2
              SP3x(I)=SUM3
              SP4x(I)=SUM4
15    CONTINUE
C
C...3) SPy(j)
C
      DO 25 J=1,NG
              SUM1=0.0
              SUM2=0.0
              SUM3=0.0
              SUM4=0.0
        DO 30 I=1,NG
              SUM1=SUM1+SP1(I,J)
              SUM2=SUM2+SP2(I,J)
              SUM3=SUM3+SP3(I,J)
              SUM4=SUM4+SP4(I,J)
30      CONTINUE
             SP1y(J)=SUM1
             SP2y(J)=SUM2
             SP3y(J)=SUM3
             SP4y(J)=SUM4
25    CONTINUE
C
      RETURN
      END
C
      SUBROUTINE RCOL2(SP1,SP2,SP3,SP4,NG,SP1XY,SP2XY,
     +                 SP3XY,SP4XY,SP1YX,SP2YX,SP3YX,SP4YX)
C
C-------------------------------------------------------------------
C
C    CALCULATE Px+y(K)=SUM{k}P(i,j)  (k=(i+j); 2,3,4...2N. )
C              Px-y(K)=SUM{k}P(i,j)  (k=abs(i-j); 0,1,2...N-1)
C
C              SP*XY -- Px+y
C              SP*YX -- Px-y
C
C---------------------------------------------------------------------
C
      PARAMETER L1=32,L2=64
      DIMENSION SP1(L1,L1),SP2(L1,L1),SP3(L1,L1),SP4(L1,L1),
     +          SP1xy(L2),SP2xy(L2),SP3xy(L2),SP4xy(L2),
     +          SP1yx(L1),SP2yx(L1),SP3yx(L1),SP4yx(L1)
C
C...1) SPx+y(K)=SP*xy(K)
C
      DO 45 K=1,2*NG-1
              SUM1=0.0
              SUM2=0.0
              SUM3=0.0
        SUBROUTINE F1(SP,NG,SASM)
C
C-------------------------------------------------------
C
C     CALCULATE THE ANGULAR SECOND MOMENT.
C
C-------------------------------------------------------
C
        DIMENSION SP(32,32)

        SASM=0.0
        DO 10 I=1,NG
          DO 10 J=1,NG
               SASM=SASM+SP(I,J)*SP(I,J)
10      CONTINUE

        RETURN
        END
C
        SUBROUTINE F2(SPYX,NG,SCTR)
C
C------------------------------------------------------------
C
C       CALCULATE THE CONTRAST.
C
C------------------------------------------------------------
C
        DIMENSION SPYX(32)
C
        SCTR=0.0
        DO 10 I=1,NG
           SCTR=SCTR+(I-1)*(I-1)*SPYX(I)
10      CONTINUE
C
        RETURN
        END
C
        SUBROUTINE F3(SP,NG,SPX,SPY,SCORRE)
C
C---------------------------------------------------------------------
C
C       CALCULATE THE CORRELATION
C
C       THE VALUE OF CORRELATION INDICATES THE RELATIONSHIP BETWEEN
C       COLUM AND ROW.
C
C       NOTE: SP(I) IS A NORMALIZED SPADEP FOR CERTAIN DIRECTION,
C             SPX(I) AND SPY(I) ARE THE MARGINAL FREQUENCY OF SP.
C             FOR GROUPED DATA SP(I), SUM(SPX(I))=1, SUM(SPY(I))=1;
C             THUS N=1.
C
C       GENERAL ALGORITHEM IS IN TE2.FOR.
C
C-----------------------------------------------------------------------
C
        DIMENSION SP(32,32),SPX(32),SPY(32)
C
C...DF1=SUM((SP(I,J)*(I-1)*(J-1))   [(I-1) AND (J-1) ARE THE CLASS MARK]
C
        SUM=0.0
        DO 10 I=1,NG
          DO 10 J=1,NG
            SUM=SUM+(I-1)*(J-1)*SP(I,J)
10      CONTINUE
C
        DF1=SUM
C
C...DF2=SUM(SPX(I)*(I-1)*SUM(SPY(I)*(I-1))
C
        SUM1=0.0
        SUM2=0.0
        DO 20 I=1,NG
          SUM1=SUM1+SPX(I)*(I-1)
          SUM2=SUM2+SPY(I)*(I-1)
20      CONTINUE
C
        DF2=SUM1*SUM2
C
C...DF3=[SUM(SPX(I)*(I-1)**2)]-[SUM(SPX(I)-(I-1)]**2
C
        SUM1=0.0
        SUM2=0.0
        DO 30 I=1,NG
          SUM1=SUM1+SPX(I)*(I-1)**2
          SUM2=SUM2+SPY(I)*(I-1)
30      CONTINUE
C
        DF3=SUM1-SUM2**2
C
C...DF4
C
        SUM1=0.0
        SUM2=0.0
        DO 40 I=1,NG
          SUM1=SUM1+SPY(I)*(I-1)**2
          SUM2=SUM2+SPY(I)*(I-1)
40      CONTINUE
C
        DF4=SUM1-SUM2**2
C
C...SCORRE=(DF1-DF2)/SQRT(DF3*DF4)
C
        DFF=SQRT(DF3*DF4)
        IF(DFF.EQ.0) THEN
                SCORRE=1
        ELSE
                SCORRE=(DF1-DF2)/DFF
        ENDIF
C
        RETURN
        END
C
        SUBROUTINE F4(SP,NG,SOS)
C
C-------------------------------------------------------------------------
C
C       THE SUM OF SQUARES: VARIANCE
C       SOS=SUM(SUM(I-U)**2*SP(I,J));
C
C-------------------------------------------------------------------------
C
        DIMENSION SP(32,32)
C
C...THE MEAN U=SUM(SP(I,J)*(I-1))/R; FOR NORMALIZED SPADEP R=1
C
        SUM=0.0
        DO 10 I=1,NG
          DO 10 J=1,NG
            SUM=SUM+SP(I,J)*(I-1)*(J-1)
10      CONTINUE
        U=SUM
C
C...SOS
C
        SUM=0.0
        DO 20 I=1,NG
          DO 20 J=1,NG
            SUM=SUM+(I-1-U)**2*SP(I,J)      ! (I-1) IS CLASS MARK WHEN THE
C                                           ! FIRST GRAY LEVEL=0
20      CONTINUE
C
        SOS=SUM
C
        RETURN
        END
C
        SUBROUTINE F5(SP,NG,SIDM)
C
C-------------------------------------------------------------------------
C
C       THE INVERSE DIFFERENCE MOMENT.
C
C       SIDM=SUM(SUM(SP(I,J)/(1+(I-J)**2))
C
C       NOTE: THE CONNOTATION OF I AND J IS NOT CLEAR YET.
C             THEY WERE TREATED AS CLASS MARK. THUS WHEN THE FIRST
C             GRAY LEVEL IS 0, I=I-1,J=J-1. BUT, (I-1)-(J-1)=I-J
C
C-------------------------------------------------------------------------
C
        DIMENSION SP(32,32)
C
        SUM=0.0
        DO 10 I=1,NG
          DO 10 J=1,NG
            SUM=SUM+SP(I,J)/(1+(I-J)**2)
10      CONTINUE
C
```

```
      SIDM=SUM
C
      RETURN
      END
C
      SUBROUTINE F6(SPXY,NG,SUMAVE)
C
C-----------------------------------------------------------------
*
*     THE SUM AVERAGE.
*
*     SUMAVE=SUM(I*SPXY(I); THE FIRST I EQUALS TO (I+J) IN THE
*                           CALCULATION OF SPXY(I).
*
*
C------------------------------------------------------------------
C
*
      DIMENSION SPXY(64)
*
      SUM=0.0
      DO 10 I=1,2*NG-1
           SUM=SUM+(I+1)*SPXY(I)
10    CONTINUE
*
      SUMAVE=SUM
*
      RETURN
      END
*
      SUBROUTINE F7(SPXY,NG,SUMETP)
C
C------------------------------------------------------------------
*
*     SUM ENTROPY.
*
*     SUMETP= - SUM (SPXY(I) * LOG(SPXY(I));  RANGE: I=1....2*NG-1.
*
*     SINCE LOG(0) IS UNDEFINED AND SPXY(I)=0 IS POSSIBLE;
*     THE FORMULA CHANGE TO:
*
*     SUMETP= - SUM(SPXY(I) * LOG(SPXY(I)+CONST)
*
*     IF SPXY(I) = 0, CONST=1
*
*     THE BASE OF LOG IS 2. SINCE LOG HAS BASE OF (E), THUS:
*
*     SUMETP= - SUM(SPXY(I) * LOG(SPXY(I)+CONST)/LOG(2)
*
C------------------------------------------------------------------------
C
      DIMENSION SPXY(64)
*
      A=2.0
      B=ALOG(A)
*
      SUM=0.0
      DO 10 I=1,2*NG-1
        IF(SPXY(I).EQ.0.0) THEN
           CONST=1.0
        ELSE
           CONST=0.0
        ENDIF
        SUM=SUM+SPXY(I)*(ALOG(SPXY(I)+CONST)/B)
10    CONTINUE
*
      SUMETP=-SUM
*
      RETURN
      END
*
      SUBROUTINE F8(SPXY,NG,SUMF7,SUMV)
C
C-------------------------------------------------------------------
*
*     THE SUM VARIANCE.
*
*     SUMV = SUM (I-SUMF7)**2*SPXY(I); I IS THE CLASS MARK, ?
*
*     WHEN FIRST GRAY LEVEL=0, I=I+1.  NOTE: TRUE FOR ALL SITUATION ?
*
*     SUMF7 IS THE  SUM ENTROPY.
*
C---------------------------------------------------------------------
```

```
C
        DIMENSION SPXY(64)
C
        SUM=0.0
        DO 10 I=1, 2*NG-1
          SUM=SUM+(I+1-SUMF7)**2*SPXY(I)
10      CONTINUE
C
        SUMV=SUM
C
        RETURN
        END
        SUBROUTINE F9(SP,NG,ENTRO)
C
C-----------------------------------------------------------------
*
*       CALCULATE THE ENTROPY.
*
*       ENTRO = - SUM [SP(I,J) * LOG(SP(I,J)+CONSTANT)/LOG(2)].
*
*       IF SP(I,J)=0, CONSTANT=1.0; ELSE, CONSTANT=0.0.
*
C-------------------------------------------------------------------
C
        DIMENSION SP(32,32)
C
        A=2.0
        B=ALOG(A)
        SUM=0.0
        DO 10 I=1,NG
          DO 10 J=1,NG
C
            IF(SP(I,J).EQ.0.0) THEN
                CONST=1.0
            ELSE
                CONST=0.0
            ENDIF
C
            SUM=SUM+SP(I,J)*(ALOG(SP(I,J)+CONST)/B)
C
10      CONTINUE
C
        ENTRO= - SUM
C
        RETURN
        END
C
        SUBROUTINE F10(SPYX,NG,DIFETP)
C
C-------------------------------------------------------------
*
*       DIFFERENCE VARIANCE.
*
*       DIFETP = - SUM(SPYX(I)*(ALOG[SPYX(I)+CONSTANT/ALOG(2)];
*
*       ARRAY SUBSCRIPT: 1, 2, 3, ... NG.
*
C---------------------------------------------------------------
C

        DIMENSION SPYX(32)
C
        A=2.0
        B=ALOG(A)
        SUM=0.0
        DO 10 I=1,NG
C
               IF (SPYX(I).EQ.0.0) THEN
                  CONST=1.0
               ELSE
                  CONST=0.0
               ENDIF
C
               SUM=SUM+SPYX(I)*(ALOG(SPYX(I)+CONST)/B)
C
10      CONTINUE
C
        DIFETP= - SUM
C
        RETURN
        END
C
        SUBROUTINE F11(SPYX,NG.DETP.DIFV)
C
C------------------------------------------------------------------
*
*       DIFFERENCE VARIANCE.
*
*       F10=DETP     (THE DIFFERENCE ENTROPY)
*
*       DIFV= SUM {(I-F10)**2*SPYX(I)}.  NOTE: THE ALGORITHEM IS
*                                              NOT SURE.
*
*       WHEN THE FIRST GRAY LEVEL = 0, I=I-1.
*
*
```

```
C----------------------------------------------------------------------
C
        DIMENSION SPYX(32)
C
        SUM=0.0
        DO 10 I=1,NG
          SUM=SUM+(I-1-DETP)**2*SPYX(I)
10      CONTINUE
C
        DIFV=SUM
C
        RETURN
        END
C
        SUBROUTINE F12(SP,SPX,SPY,NG,HXY,SIMC1,SIMC2)
C
C--------------------------------------------------------------------------
C
*       INFORMATION MEASURES OF CORRELATION.
*
*       SIMC1 = (HXY - HXY1) / MAX(HX,HY);
*
*       SIMC2 = SQRT{(1-exp[-2.0(HXY2 - HXY])}.
*
*      HXY: THE ENTROPY, CACULATED IN F9;
*       HX = - SUM [SPX(I)*(LOG(SPX(I)+CONSTANT)/LOG(2))];ENTROPY OF SPX(I)
*       HY = - SUM [SPY(I)*(LOG(SPY(I)+CONSTANT)/LOG(2))];ENTROPY OF SPY(I)
*     HXY1 = - SUM { SP(I,J) * (LOG[SPX(I)*SPY(I)+CONSTANT]/LOG(2)] };
*     HXY2 = - SUM { SPX(I)*SPY(I)  *  (LOG[SPX(I)*SPY(I)+CONSTANT]/LOG(2))}.
*
C----------------------------------------------------------------------------
C
        DIMENSION SP(32,32),SPX(64),SPY(64)
C
        A=2.0
        B=ALOG(A)
C
C...HX, HY
C
        HX=0.0
        HY=0.0
C
        DO 10 I=1,NG
C
          IF (SPX(I) .EQ. 0.0) THEN
            CONST1=1.0
          ELSE
            CONST1=0.0
          ENDIF
          IF (SPY(I) .EQ. 0.0) THEN
            CONST2=1.0
          ELSE
            CONST2=0.0
          ENDIF
C
          HX=HX+SPX(I)*(ALOG(SPX(I)+CONST1)/B)
          HY=HY+SPY(I)*(ALOG(SPY(I)+CONST2)/B)
C
10      CONTINUE
C
        HX = - HX
        HY = - HY
C
C...HXY1, HXY2
C
        HXY1=0.0
        HXY2=0.0
C
        DO 20 I=1,NG
          DO 20 J=1,NG
C
            IF (SPX(I).EQ.0.0 .OR. SPY(J).EQ.0.0) THEN
              CONST=1.0
            ELSE
              CONST=0.0
            ENDIF
C
            HXY1=HXY1 + SP(I,J) * (ALOG (SPX(I)*SPY(J) + CONST)/B)
            HXY2=HXY2 + SPX(I) * SPY(J)*(ALOG(SPX(I)*SPY(J)+CONST)/B)
C
20      CONTINUE
C
        HXY1 = - HXY1
        HXY2 = - HXY2
```

```
C
C...SIMC1
C
        IF(AMAX1(HX,HY).EQ.0) THEN
                SIMC1=0
        ELSE
                SIMC1 = (HXY - HXY1) / AMAX1(HX,HY)
        ENDIF
C
C...SIMC2
C
        A= EXP((-2.0)*(HXY2-HXY))
C
        SIMC2 = SQRT(1-A)
C
        RETURN
        END
        SUBROUTINE F13(SP,SPX,SPY,NG,SUBMCC)
C
C-----------------------------------------------------------
*
*       MAXIMAL CORRELATION COEFFICIENT.
*
*       SUBMCC = SQRT(SECOND LARGEST EIGENVALUE OF Q)
*
*       Q(I,J)  = SUM(SP(I,K)*SP(J,K)/SPX(I)*SPY(K))
*
*       k= 1, 2, 3, ... NG.
*
*       PROCEDURE:
*         1) CREATE Q(I,J);
*         2) CONVERT Q(I,J) TO HESSENBERG MATRIX;
*         3) CALCULATE EIGENVALUES OF Q(I,J); (IT IS POSSIBLE TO
*            BUILD THE EIGENVECTOR MATRIX WITH THESE EIGENVALUES);
*         4) FIND THE SECOND LARGEST EIGENVALUE OF Q(I,J).
*
*       THE CRITICAL ALGORITHEMS OF THIS PORTION IS BASED ON THE BASIC
*       LANGUAGE PROGRAM DESIGNED BY ZHANG et. al. IN QINGHUA U. CHINA.
*
*       NOTE: THE ACCURACY OF COMPUTATION IS INFLUENCED BY THE SIZE
*       OF THE ORIGINAL MATRIX, i.e. THE NUMBER OF GRAY LEVEL WE
*       DEAL WITH. THE MEANING OF THIS MAXIMAL CORRELATION COEFFICIENT
*       STILL NEEDS TO FIND OUT, SO IS THE GRAPHIC DISPLAY OF THIS
*       MEASUREMENT.
*
*
C-------------------------------------------------------------------------
C
        DIMENSION SP(32,32),SPX(64),SPY(64),Q(36,36),EIGR(32)
C
          DO 10 I=1,NG
            DO 10 J=1,NG
              SUM=0.0
              DO 20 K=1,NG
                IF(SPX(I) .EQ. 0 .OR. SPY(K) .EQ. 0) GOTO 20
                SUM=SUM+(SP(I,K)*SP(J,K))/(SPX(I)*SPY(K))
20      CONTINUE
                Q(I,J)=SUM
10      CONTINUE
C
C...CONVERT Q(I,J) TO HESSENBERG MATRIX
C
        CALL HESSEN(Q,NG)
C
C...CALCULATE EIGENVALUE FOR Q(I,J) CONVERTED TO HESSENBER MATRIX.
C
        CALL EIGEN(Q,NG,EIGR)
C
C...FIND THE SECOND LARGEST EIGENVALUE OF Q(I,J)
C
        DO 60 J=1,NG-1
                DO 50 I=1,NG-1
                  IF(EIGR(I).GT.EIGR(I+1)) THEN
                    TEMP=EIGR(I)
                    EIGR(I)=EIGR(I+1)
                    EIGR(I+1)=TEMP
                  ENDIF
50              CONTINUE
60      CONTINUE
C
        SMAX=EIGR(NG-1)
C
        SUBMCC= SQRT(ABS(SMAX))
C
        RETURN
        END
C
        SUBROUTINE HESSEN(HA,SNG)
C
C-------------------------------------------------------------------------
*
*       PROGRAM TO TRANSFOR GENERAL MATRIX TO THE HESSENBERG MATRIX.
*
*       THE ALGORITHEM IS DESIGNED BY ZHANG et. al IN QINGHUA U. CHINA.
*
*       THE HESSENBERG MATRIX IS TO USED FOR EIGENVALUE CALCULATION WITH
*       THE QR METHOD.
*
*       A(SN,SN+4); B(SN)         ! CURRENTLY A(SN, SN),B(SN);
*                                 ! SN=<86.
C-------------------------------------------------------------------------
```

```
C
        DIMENSION HA(36,36),B(32)
        INTEGER SNG,SN
C
        SN=SNG
C
C...INITIALIZE B(I) WITH 1, 2, 3,....SN
C
        DO 10 I=1,SN
          B(I)=I
10      CONTINUE
C
C...CORE OF THE PROGRAM
C
        L=SN-1
        DO 20 M=2,L
             I=M
             X=0.0
          DO 30 J=M,SN
            IF( ABS(HA(J,M-1)) .LE. ABS(X)) GOTO 30
                X=HA(J,M-1)
                I=J
30      CONTINUE
C
        IF( I .EQ. M) GOTO 9999
C
        Y=B(M)
        B(M)=B(I)
        B(I)=Y
C
        DO 40 J=M-1, SN
           Y=HA(I,J)
           HA(I,J)=HA(M,J)
           HA(M,J)=Y
40      CONTINUE
C
        DO 50 J=1,SN
           Y=HA(J,I)
           HA(J,I)=HA(J,M)
           HA(J,M)=Y
50      CONTINUE
C
9999    IF( X .EQ. 0.0) GOTO 20
C
        DO 60 I=M+1,SN
           Y=HA(I,M-1)
           IF( Y .EQ. 0.0) GOTO 60
           Y=Y/X
           HA(I,M-1)=Y
           DO 70 J=M,SN
              HA(I,J)=HA(I,J)-Y*HA(M,J)
70      CONTINUE
           DO 80 J=1,SN
              HA(J,M)=HA(J,M)+Y*HA(J,I)
80      CONTINUE
60      CONTINUE
C
20      CONTINUE
C
        DO 90 I=1,SN
          DO 90 J=1,SN
            IF( I .LE. 2 .OR. J .GT. I-2) GOTO 90
            HA(I,J)=0.0
90      CONTINUE
C
        RETURN
        END
C
        SUBROUTINE EIGEN(A,SNG,DR)
C
C----------------------------------------------------------------------
*
*       QR METHOD TO CALCULATE THE EIGENVALUE OF MATRIX.
*
*       INPUT MATRIX MUST BE A HESSENGER MATRIX. GENERAL MATRIX CAN
*       BE CONVERTED TO A HESSENGER MATRIX BY THE PROGRAM 'UPPERH.FOR'.
*
*       THE ALGORITHEM IS DESIGNED BY ZHANG et. al AT QINGHUA U. CHINA.
*
*       ARRAY
*           A(N+4,N+4).. INPUT HESSENGER MATRIX;
*           B(N+4)...... RECORD OF SEARCHING TIMES;
*           R(N)........ THE REAL PARTS OF THE EIGENVALUE;
*           I(N)........ THE IMAGERARY PARTS OF THE EIGENVALUE;
*           N........... SIZE OF THE MATRIX;
*           E........... ERROR INDEX, SPECIFIED TO 0.1.
*
C-----------------------------------------------------------------------
C
        DIMENSION A(36,36),B(36),DR(32),DI(32)
        INTEGER SNG
C
        N=SNG
C
        E=0.1
        T=0.0
        T2=N
7160    IF( N .EQ. 0) GOTO 7410
        T1=0
        N1=N-1
C
7166    DO 10 L=N,2,-1
        IF(ABS(A(L,L-1)).LE.E*(ABS(A(L-1,L-1))+ABS(A(L,L))))GOTO 7174
10      CONTINUE
C
        L=1
7174    X=A(N,N)
C
        IF( L .EQ. N) THEN
                DR(N)=X+T
                DI(N)=0.0
                B(N)=T1
                N=N1
                GOTO 7160
        ENDIF
                Y=A(N1,N1)
                W=A(N,N1)*A(N1,N)
         IF( L .EQ. N1) THEN
                P=(Y-X)/2
                Q=P*P+W
                Y=SQRT(ABS(Q))
                B(N)=-T1
                B(N1)=T1
                X=X+T
                IF( Q .LE. 0) THEN
                        DR(N1)=X+P
                        DR(N)=X+P
                        DI(N1)=Y
                        DI(N)=-Y
                ELSEIF( P .GE. 0) THEN
                        Y=P+Y
                        DR(N1)=X+Y
                        DR(N)=X-W/Y
                        DI(N1)=0.0
                        DI(N)=0.0
                ELSE
                        Y=-Y
                        Y=P+Y
                        DR(N1)=X+Y
                        DR(N)=X-W/Y
                        DI(N1)=0.0
                        DI(N)=0.0
                ENDIF
C
                N=N-2
                GOTO 7160               ! STARTING AGAIN.
         ENDIF
C
         IF( T1 .EQ. 10 .OR. T1 .EQ. 20) THEN
                T=T+X
            DO 20 I=1,N
                A(I,I)=A(I,I)-X
20          CONTINUE
                S=ABS(A(N,N1))+ABS(A(N1,N-2))
                X=0.75*S
                Y=0.75*S
                W=(-0.4375)*S*S
         ELSEIF( T1 .EQ. 60) THEN
                WRITE(7,*)  'EIGENVALUE NOT FOUND '
         RETURN
         ENDIF
C
         DO 30 M=N-2,L,-1
                Z=A(M,M)
                R=X-Z
                S=Y-Z
                P=(R*S-W)/A(M+1,M)+A(M,M+1)
                Q=A(M+1,M+1)-Z-R-S
```

```
                  R=A(M+2,M+1)
                  S=ABS(P)+ABS(Q)+ABS(R)
                  P=P/S
                  Q=Q/S
                  R=R/S
                  IF( M .EQ. L) GOTO 7240
         U=E*ABS(P)*(ABS(A(M-1,M-1))+ABS(Z)+ABS(A(M+1,M+1)))
         IF( ABS(A(M,M-1))*(ABS(Q)+ABS(R)) .LE. U) GOTO 7240
C
30       CONTINUE
C
7240     DO 40 I=M+2,N
                  A(I,I-2)=0.0
40       CONTINUE
C
         DO 50 I=M+3,N
                  A(I,I-3)=0.0
50       CONTINUE
C
         DO 60 K=M,N1
                  IF( K .NE. N1) N2=1
                  IF( K .NE. M) THEN
                           P=A(K,K-1)
                           Q=A(K+1,K-1)
                           R=0.0
                           IF(N2 .EQ. 1) R=A(K+2,K-1)
                           X=ABS(P)+ABS(Q)+ABS(R)
                           IF( X .EQ. 0) GOTO 60             !NEXT K.
                           P=P/X
                           Q=Q/X
                           R=R/X
                  ENDIF
C
                  S=SQRT(P*P+Q*Q+R*R)
C
                  IF( P .LT. 0) S=-S
C
                  IF( K .NE. M) THEN
                          A(K,K-1)=(-S)*X
                  ELSEIF( L .NE. M) THEN
                          A(K,K-1)= -A(K,K-1)
                  ENDIF
C
                  P=P+S
                  X=P/S
                  Y=Q/S
                  Z=R/S
                  Q=Q/P
                  R=R/P
C
         DO 70 J=K,N
                  P=A(K,J)+Q*A(K+1,J)
                  IF( N2 .EQ. 0) GOTO 7320
                  P=P+R*A(K+2,J)
                  A(K+2,J)=A(K+2,J)-P*Z
7320              A(K+1,J)=A(K+1,J)-P*Y
                  A(K,J)=A(K,J)-P*X
70       CONTINUE
C
                  IF( K+3 .GE. N) THEN
                           J=N
                  ELSE
                           J=K+3
                  ENDIF
C
         DO 80 I=L,J
                  P=X*A(I,K)+Y*A(I,K+1)
                  IF( N2 .EQ. 0) GOTO 7344
                  P=P+Z*A(I,K+2)
                  A(I,K+2)=A(I,K+2)-P*R
7344              A(I,K+1)=A(I,K+1)-P*Q
                  A(I,K)=A(I,K)-P
80       CONTINUE
60       CONTINUE
C
         T1=T1+1
C
         GOTO 7166                ! LOOP 10.
C
7410     CONTINUE
7416     RETURN
         END
```

```
C
C-----------------------------------------------------------
        PROGRAM TXGG3
C
C       PROGRAM TO PLOT INDIVIDUAL TEXTURAL MEASURES.
C
C       AUG. 9, 1987.
C
C-----------------------------------------------------------
C
        DIMENSION X(12),Y(12),A(7,7)
        CHARACTER*10 INFIL(10),FORM
        DATA FORM/'(''$'',A)'/
        DATA X /1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,0.,0./
        DATA Y /.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,0.,0./
C
        CALL MPIOPS
C
C       WRITE(7,FORM) 'HOW MANY FILES TO COMPARE : '
C       READ(5,*) N
        N=10
        WRITE(7,*) 'ENTER 10 FILE NAMES : '
        DO 5 I=1,N
            READ(5,'(A)') INFIL(I)
5       CONTINUE
C
        CALL SCALE (X,7.0,10,1)
        CALL SCALE (Y,6.0,10,1)
C
        WRITE(7,FORM) 'FEATURE YOU WANT TO PLOT : '
        READ(5,*) TX
        WRITE(7,FORM) '0(1),90(2),45(3),135 (4),RANGE (5),AVERAG (6),
     +VARIANT (7) : '
        READ(5,*) IF
C
        CALL PLOTS (0,0,0)
        CALL PLOT (1.0,1.0,-3)
C
        M=0
888     M=M+1
        OPEN(UNIT=1,FILE=INFIL(M),STATUS='OLD')
C
        READ(1,*) ((A(I,J),J=1,7),I=1,7)
C
        TEMP=A(TX,IF)
        IF(TX.GE.3.AND.TX.LE.5.AND.IF.NE.5.AND.IF.NE.7) THEN
                TEMP=TEMP/10
        ELSE
                TEMP=SQRT(TEMP)
        ENDIF
        Y(M)=TEMP
C
        CLOSE(1)
        IF(M.EQ.N) GOTO 777
        GOTO 888
C
777       CALL AXIS (0.0,0.0,'TEXTURE MEASURES',-16,7.0,0.0,X(11),X(12))
          CALL AXIS (0.0,0.0,'VALUE',5,6.0,90.0,Y(11),Y(12))
          CALL LINE (X,Y,10,1,1,11)
C
        STOP
        END
```

```
C
C-----------------------------------------------------------------
C
      PROGRAM FTXTUR
C
C  PERFORMS TWO DIMENSIONAL FFT, THEN CALCULATE THE REGIONAL ENTROPY.
C  THE CURRENT PROGRAM WORK FOR 32X32 SUBIMAGE. USING THE VIRTUAL
C  MEMORY, ANALYSIS CAN BE PERFORMED ON UP TO 128X128 SUBIMAGE.
C
C  INPUT DATA IS READ FROM THE PICTURE PLANE '0', SUBIMAGE AREA IS
C  SELECTED BY POINTING THE CURSOR TO THE LEFT CORNOR OF THE AREA.
C
C  OUTPUT UNIT 2 CONTAINS THE OUTPUT FILE OF THE GRAY-LEVEL-SCALED
C  FOURIER SPECTRUM TO BE PLOT BY 'PLOTFF.FOR' AS AN IMAGE OR 'FFT3D.FOR'
C  AS A 3-D PLOT.
C
C  THE SIZES OF ENTROPY REGIONS ARE 25 X 25, 17 X 17, 11 X 11, 5 X 5.
C
C  THIS ALGORITHM OF REGIONAL ENTROPY ANALYSIS IS PROPOSED BY
C  M.E. JERNIGAN AND F. D'ASTOUS. IN 'ENTROPY-BASED TEXTURE ANALYSIS IN
C  THE SPPATIAL FREQUENCY DOMAIN', IEEE, TRANSACTIONS ON PATTERN
C  ANALYSIS AND MACHINE INTELLEGENCE, VOL. PAMI-6, NO. 2, MARCH 1984.
C
C  LI BIN, JULY 22, 1987.
C
C-----------------------------------------------------------------
C
      DIMENSION H(32,32),B(32),QB(32,32)
      COMPLEX B,QB
      REAL K
      CHARACTER*10 OUTFL1,OUTFL2,FORM
      DATA FORM/'(''$'',A)'/
C
      CALL MPIOPS
      CALL ERASER
      NN=32
C
      WRITE(7,FORM) 'ENTER FILE NAME FOR THE TXTURE MEASURES : '
      READ(5,'(A)') OUTFL2
C
      OPEN (UNIT=1,FILE=OUTFL2,STATUS='NEW')
C
C  INPUT SECTION
C
9899  WRITE(7,FORM) 'ENTER FILE NAME FOR THE FOURIER SPECTRUM : '
      READ(5,'(A)') OUTFL1
      OPEN (UNIT=2,FILE=OUTFL1,STATUS='NEW',FORM='UNFORMATTED')
C
1     WRITE(7,FORM) 'JOYSTIC (0) OR KEYBORAD (1) COORDINATES ? '
      READ(5,*) ISEL
C
      IF (ISEL .EQ. 0) THEN
              WRITE (7,FORM) 'Hit <BS>'
              CALL CURSOR(IX,IY)
              WRITE(7,*) 'WRITE DOWN THE POSITION : ',IX,IY
      ELSE
              WRITE (7,FORM) 'ENTER X, Y COORDINATES : '
              READ (5,*) IX,IY
      ENDIF
C
      IXSIZE=NN
      IYSIZE=NN
C
      CALL BOXON(IX,IY,IXSIZE,IYSIZE)
C
      WRITE(7,FORM) 'IS THIS AREA ACCEPTABLE (1/0) ? '
      READ(5,*) RESPONSE
      IF( RESPONSE .EQ. 0) THEN
                 CALL BOXOFF(IX,IY,IXSIZE,IYSIZE)
                 GO TO 1
      ENDIF
C
      CALL PEER(0)
            DO 22 IROW=1,IYSIZE
               DO 33 ICOL=1,IXSIZE
                     CALL GRAFIN(IX+ICOL-1,IY+IROW-1,IZ1)
                     H(IROW,ICOL)=IZ1
33             CONTINUE
22             CONTINUE
C
C  FFT SECTION
C
C     WRITE(7,*)'ORIGIN CENTERED FFT ? 1/0'
C     READ(5,*) OPTION
```

```
C
        OPTION = 1
        WRITE(7,*) 'LOG SCALE?(1=Y,0=N)'
        READ(5,*) SCALOG
        IF(SCALOG.EQ.1) THEN
                WRITE(7,*) 'ENTER SCALE FACTOR K'
                READ(5,*) K
        ENDIF
C       WRITE(7,*) 'ENTER GRAY LEVEL'
C       READ(5,*) G
C
        G=255
C
C TRANSFORM THE ROWS OF H(I,J), STORE IN Q(I,L)
C
        T1=SECNDS(0.0)
        DO 10 I=1,NN
                ICOUNT=0
          DO 20 J=1,NN
                ICOUNT=ICOUNT+1
                IF (OPTION.EQ.1) IC=(-1)**(I+J)
                IF (OPTION.EQ.0) IC=1
                B(ICOUNT)=H(I,J)*IC
20        CONTINUE
                CALL FOUREA(B,NN,-1)
                M=0
                DO 30 L=1,NN
                   M=M+1
                   QB(I,L)=B(M)
30              CONTINUE
10      CONTINUE
C
C TRANSFORM THE COLUMNS OF QB(I,J)
C
        DO 50 J=1,NN
                ICOUNT=0
          DO 60 I=1,NN
                ICOUNT=ICOUNT+1
                B(ICOUNT)=QB(I,J)/NN             ! DIVIDED BY 1/N.
60        CONTINUE
                CALL FOUREA(B,NN,-1)
                M=0
                DO 70 L=1,NN
                   M=M+1
                   QB(L,J)=B(M)
70              CONTINUE
50      CONTINUE
        TYPE*,'TIME IN TRANSFORM = ',SECNDS(T1),' SECONDS.'
C
        U=7
        VMIN=1.0E9
        VMAX=-1.0E9
        DO 80 I=1,NN
          DO 80 J=1,NN
            PSD=QB(I,J)*CONJG(QB(I,J))
            QB(I,J)=PSD
            H(I,J)=SQRT(PSD)
            VMAX=AMAX1(H(I,J),VMAX)
            VMIN=AMIN1(H(I,J),VMIN)
            IF(SCALOG.NE.1) GOTO 80
            H(I,J)=LOG(1+K*H(I,J))
80      CONTINUE
C
C.. SUMS OF 4 REGIONS: 25X25, 17X17, 11X11, 5X5
C
        IW1=NN/2+1-12
        IW11=NN/2+1+12
        SUM1=0.0
        DO 81 I=IW1,IW11
          DO 81 J=IW1,IW11
                SUM1=SUM1+QB(I,J)
81      CONTINUE
        IW2=NN/2+1-8
        IW22=NN/2+1+8
        SUM2=0.0
        DO 82 I=IW2,IW22
          DO 82 J=IW2,IW22
                SUM2=SUM2+QB(I,J)
82      CONTINUE
        IW3=NN/2+1-5
        IW33=NN/2+1+5
        SUM3=0.0
        DO 83 I=IW3,IW33
          DO 83 J=IW3,IW33
```

```
                SUM3=SUM3+QB(I,J)
83      CONTINUE
        IW4=NN/2+1-2
        IW44=NN/2+1+2
        SUM4=0.0
        DO 84 I=IW4,IW44
          DO 84 J=IW4,IW44
                SUM4=SUM4+QB(I,J)
84      CONTINUE
C
C ENTROPY
C
        AA=2.0
        BASE=ALOG(AA)
C
        ETP1=0.0
        DO 85 I=IW1,IW11
          DO 85 J=IW1,IW11
                TEMP1=QB(I,J)/SUM1
                TEMP2=ALOG(TEMP1)
                ETP1=ETP1+TEMP1*TEMP2
85      CONTINUE
        ETP1=-ETP1/(ALOG(25.0*25.0)/BASE)
C
        ETP2=0.0
        DO 86 I=IW2,IW22
          DO 86 J=IW2,IW22
                TEMP1=QB(I,J)/SUM2
                TEMP2=ALOG(TEMP1)
                ETP2=ETP2+TEMP1*TEMP2
86      CONTINUE
        ETP2=-ETP2/(ALOG(17.0*17.0)/BASE)
C
        ETP3=0.0
        DO 87 I=IW3,IW33
          DO 87 J=IW3,IW33
                TEMP1=QB(I,J)/SUM3
                TEMP2=ALOG(TEMP1)
                ETP3=ETP3+TEMP1*TEMP2
87      CONTINUE
        ETP3=-ETP3/(ALOG(11.0*11.0)/BASE)
C
        ETP4=0.0
        DO 88 I=IW4,IW44
          DO 88 J=IW4,IW44
                TEMP1=QB(I,J)/SUM4
                TEMP2=ALOG(TEMP1)
                ETP4=ETP4+TEMP1*TEMP2
88      CONTINUE
        ETP4=-ETP4/(ALOG(5.0*5.0)/BASE)
C
C PRINT THE TXTURE MEASURES
C
        WRITE(7,7777) 'ETP1 = ',ETP1,' ETP2 = ',ETP2
        WRITE(7,7777) 'ETP3 = ',ETP3,' ETP4 = ',ETP4
7777    FORMAT(1X,2(A,F10.5))
        WRITE(1,'(4F15.4)') ETP1,ETP2,ETP3,ETP4
C
        VMAX=-1.0E9
        VMIN=1.0E9
        DO 90 I=1,NN
          DO 90 J=1,NN
            IF(I.EQ.NN/2+1.AND.J.EQ.NN/2+1) GOTO 90
            VMAX=AMAX1(H(I,J),VMAX)
            VMIN=AMIN1(H(I,J),VMIN)
90      CONTINUE
C
        RANG=VMAX-VMIN
C
        DO 95 I=1,NN
          DO 95 J=1,NN
            H(I,J)=((H(I,J)-VMIN)/RANG)*G
            IF(I.EQ.NN/2+1.AND.J.EQ.NN/2+1) H(I,J)=255
95      CONTINUE
C
        DO 100 I=1,NN
          WRITE(2) (H(I,J),J=1,NN)
100     CONTINUE
C
        CLOSE(2)
        WRITE(7,FORM) 'NEXT FILE ? (1/0) '
        READ (5,*) NEXT
        IF(NEXT.EQ.0) GOTO 9090
        GOTO 9899
```

```
C
9090    CALL OFF('G')
        STOP
        END
```

```
C
C-------------------------------------------------------------
C
C  SUBROUTINE: FOUREA
C  PERFORMS COOLEY-TUKEY FAST FOURIER TRANSFORM
C
C-------------------------------------------------------------
C
        SUBROUTINE FOUREA(DATA,N,ISI)
C
C THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN ANSI FORTRN
C
C DATA IS A ONE-DIMENSIONAL COMPLEX ARRAY WHOSE LENGTH, N IS A
C POWER OF TWO. ISI IS +1 FOR AN INVERSE TRANSFERM AND -1 FOR A
C FORWARD TRANSFORM. TRANSFORM VALUES ARE RETURNED IN THE INPUT
C ARRAY, REPLACING THE INPUT.
C AFTER PROGRAM BY BRENNER, JUNE 1967.
C
        VIRTUAL DATA(1)
        COMPLEX DATA
        COMPLEX TEMP,W

        PI=4.*ATAN(1.)
        FN=N
C
C PUT DATA IN BIT-REVERSED ORDER
C
        J=1
        DO 80 I=1,N
C
C AT THIS POINT, I AND J ARE A BIT REVERSED PAIR (EXCEPT FOR THE
C DISPLACEMENT OF +1
C
                IF(I-J) 30,40,40
C
C EXCHANGE DATA(I) WITH DATA(J) IF I.LT.J
C
30              TEMP=DATA(J)
                DATA(J)=DATA(I)
                DATA(I)=TEMP
C
C IMPLEMENT J=J+1, BIT-REVERSED COUNTER
C
40              M=N/2
50              IF(J-M) 70,70,60
60              J=J-M
                M=(M+1)/2
                GOTO 50
70              J=J+M
80      CONTINUE
C
C COMPUTE THE BUTTERFLIES
C
        MMAX=1
90      IF(MMAX-N) 100,130,130
100     ISTEP=2*MMAX
        DO 120 M=1,MMAX
                THETA=PI*FLOAT(ISI*(M-1))/FLOAT(MMAX)
                W=CMPLX(COS(THETA),SIN(THETA))
                DO 110 I=M,N,ISTEP
                  J=I+MMAX
                  TEMP=W*DATA(J)
                  DATA(J)=DATA(I)-TEMP
                  DATA(I)=DATA(I)+TEMP
110             CONTINUE
120     CONTINUE
        MMAX=ISTEP
        GOTO 90
130     IF (ISI) 160,140,140
C
C FOR INVERSE TRANSFORM -- ISI=1 -- MULTIPLY OUTPUT BY 1/N
C
140     DO 150 I=1,N
                DATA(I)=DATA(I)/FN
150     CONTINUE
160     RETURN
        END
```

```
$ TY FFT2.FOR

C
C---------------------------------------------------------------
C
C  MAIN PROGRAM: FFT2.FOR
C
C  PERFORMS TWO DIMENSIONAL FFT, THE OUTPUT FILE IS THE UNSCALED FOURIER
C  SPECTRUM WHICH CAN BE DIRECTLY PLOTTED AS 3D SURFACE BY 'FFT3D.FOR'.
C  THE CURRENT PROGRAM WORK FOR 32X32 SUBIMAGE. WITH SLIGHT MODIFICATION,
C  ANALYSIS CAN BE PERFORMED ON UP TO 128X128 SUBIMAGE.
C
C  IF THE VIRTUAL MEMEORY IS TO BE USED, CHANGE ALL REGULAR ARRAYS TO
C  VIRTUAL ARRAYS (INCLUDING ARRAY IN THE SUBROUTINE 'FOUREA.FOR'), THEN
C  LINK THE PROGRAM AS FOLLOWING:
C
C  F77 FFTXR
C  LINK FFTXR/XM,FOUREA/XM,SY:VIRTXM,SY:F77LIB
C
C  NOTE:  SUBROUTINE 'BOXON' AND 'BOXOFF' ARE NOT INCLUDED HERE, THEY CAN
C  BE FOUND IN 'FTXTUR.FOR'.
C
C  LI BIN, JULY 22, 1987.
C
C----------------------------------------------------------------
C
        DIMENSION H(32,32),B(32),QB(32,32)
        COMPLEX B,QB
        REAL K
        CHARACTER*10 OUTFL1,FORM
        DATA FORM/'(''$'',A)'/
C
        CALL MPIOPS
        CALL ERASER
        NN=32
C
C
C  INPUT SECTION
C
9899    WRITE(7,FORM) 'ENTER FILE NAME FOR THE FOURIER SPECTRUM : '
        READ(5,'(A)') OUTFL1
        OPEN (UNIT=2,FILE=OUTFL1,STATUS='NEW',FORM='UNFORMATTED')
C
1       WRITE(7,FORM) 'JOYSTIC (0) OR KEYBORAD (1) COORDINATES ? '
        READ(5,*) ISEL
C
        IF (ISEL .EQ. 0) THEN
                WRITE (7,FORM) 'Hit <BS>'
                CALL CURSOR(IX,IY)
                WRITE(7,*) 'WRITE DOWN THE POSITION : ',IX,IY
        ELSE
                WRITE (7,FORM) 'ENTER X, Y COORDINATES : '
                READ (5,*) IX,IY
        ENDIF
C
        IXSIZE=NN
        IYSIZE=NN
C
        CALL BOXON(IX,IY,IXSIZE,IYSIZE)
C
        WRITE(7,FORM) 'IS THIS AREA ACCEPTABLE (1/0) ? '
        READ(5,*) RESPONSE
        IF( RESPONSE .EQ. 0) THEN
                   CALL BOXOFF(IX,IY,IXSIZE,IYSIZE)
                   GO TO 1
        ENDIF
C
        CALL PEER(0)
              DO 22 IROW=1,IYSIZE
                 DO 33 ICOL=1,IXSIZE
                       CALL GRAFIN(IX+ICOL-1,IY+IROW-1,IZ1)
                       H(IROW,ICOL)=IZ1
33                CONTINUE
22                CONTINUE
C
C  FFT SECTION
C
C       WRITE(7,*)'ORIGIN CENTERED FFT ? 1/0'
C       READ(5,*) OPTION
C
        OPTION = 1
C
C TRANSFORM THE ROWS OF H(I,J), STORE IN Q(I,L)
C
```

```
        T1=SECNDS(0.0)
        DO 10 I=1,NN
                ICOUNT=0
          DO 20 J=1,NN
                ICOUNT=ICOUNT+1
                IF (OPTION.EQ.1) IC=(-1)**(I+J)
                IF (OPTION.EQ.0) IC=1
                B(ICOUNT)=H(I,J)*IC
20        CONTINUE
                CALL FOUREA(B,NN,-1)
                M=0
                DO 30 L=1,NN
                   M=M+1
                   QB(I,L)=B(M)
30              CONTINUE
10      CONTINUE
C
C TRANSFORM THE COLUMNS OF QB(I,J)
C
        DO 50 J=1,NN
                ICOUNT=0
          DO 60 I=1,NN
                ICOUNT=ICOUNT+1
                B(ICOUNT)=QB(I,J)/NN            ! DIVIDED BY 1/N.
60        CONTINUE
                CALL FOUREA(B,NN,-1)
                M=0
                DO 70 L=1,NN
                   M=M+1
                   QB(L,J)=B(M)
70              CONTINUE
50      CONTINUE
        TYPE*,'TIME IN TRANSFORM = ',SECNDS(T1),' SECONDS.'
C
        U=7
        VMIN=1.0E9
        VMAX=-1.0E9
        DO 80 I=1,NN
          DO 80 J=1,NN
            PSD=QB(I,J)*CONJG(QB(I,J))
            QB(I,J)=PSD             ! QB(I,J) STORES THE POWER SPECTRUM.
            H(I,J)=SQRT(PSD)        ! H(I,J) STORES THE FOURIER SPECTRUM.
            IF(I.EQ.NN/2+1.AND.J.EQ.NN/2+1) GOTO 80    !SKIP D.C.VALUE.
            VMAX=AMAX1(H(I,J),VMAX)
            VMIN=AMIN1(H(I,J),VMIN)
80      CONTINUE
C
        H(NN/2+1,NN/2+1)=VMAX
C
        DO 100 I=1,NN
          WRITE(2) (H(I,J),J=1,NN)
100     CONTINUE
C
        CLOSE(2)
        WRITE(7,FORM) 'NEXT FILE ? (1/0) '
        READ (5,*) NEXT
        IF(NEXT.EQ.0) GOTO 9090
        GOTO 9899
C
9090    CALL OFF('G')
        STOP
        END
```

```
C
C--------------------------------------------------------
        PROGRAM COMFTX
C
C       CALCULATE THE DISTANCE AMONG THE ENTROPY VECTORS
C       FOR 10 SUBIMAGES. INPUT FILE SHOULD BE A 10 X 4
C       MATRIX (OUTPUT FROM FTXTUR.FOR)
C--------------------------------------------------------
C
        DIMENSION A(10,4), DIF(10,10,4),DIS(10,10)
        CHARACTER*10 FORM1,FORM2,INFL,OUTFL
        DATA FORM1/'(''$'',A)'/FORM2/'(A)'/
C
        WRITE(7,FORM1) 'ENTER INPUT FILE NAME : '
        READ(5,FORM2) INFL
        WRITE(7,FORM1) 'ENTER OUTPUT FILE NAME : '
        READ(5,FORM2) OUTFL
        WRITE(7,FORM1) 'SELECTION OF OUTPUT DEVICE (2 FOR DISK OUTPUT) : '
        READ(5,*) U
C
        OPEN (UNIT=1,FILE=INFL,STATUS='OLD')
        IF(U.EQ.2) THEN
                OPEN (UNIT=2,FILE=OUTFL,STATUS='NEW')
        ENDIF
C
        READ(1,*) ((A(I,J),J=1,4),I=1,10)
        WRITE(7,'(4F12.5)') ((A(I,J),J=1,4),I=1,10)
C
C DISTANCES CALCULATED FROM INDIVIDUAL ENTROPY REGIONS.
C
        DO 10 K=1,4
          DO 10 I=1,10
            DO 10 J=1,10
              DIF(I,J,K)=ABS(A(I,K)-A(J,K))
10      CONTINUE
C
C DISTANCES AMONG SUBIMAGES
C
           DO 500 I=1,10
             DO 500 J=1,10
                 SUM=0.0
                 DO 600 K=1,4
                         SUM=SUM+DIF(I,J,K)**2
600              CONTINUE
                 DIS(I,J)=SQRT(SUM)
500     CONTINUE
C
        DO 100 K=1,4
                WRITE(U,200) ((DIF(I,J,K),J=1,10),I=1,10)
100             WRITE(U,*)
                WRITE(U,200) ((DIS(I,J),J=1,10),I=1,10)
C
200     FORMAT(10F8.3)
        STOP
        END
```

```
C
C----------------------------------------------------------------
C
C  MAIN PROGRAM: FFT.FOR
C
C  PERFORMS TWO DIMENSIONAL FFT.
C  THIS PROGRAM IS DESIGNED FOR TESTING PURPOSE. THE SUBROUTINE
C  IT SHOULD LINK WITH IS FOUREA.FOR WHICK ALSO USES VIRTUAL
C  MEMORY. THE TESTING DATA SET CAN BE MADE UP BY THE USER.
C
C  H(128,128) IS THE INPUT MATRIX.
C  B(128) STORE ONE ROW (COLUMN) OF H THEN PERFORM FFT ON THIS ARRAY.
C  QB(128,128) THE RESULTANT FORWARD FFT MATRIXE, COMPLEX VARIABLE.
C  HB(128,128) THE RESULTANT INVERSE FFT MATRIXE, COMPLEX VARIABLE.
C
C  SINCE H(128,128) IS REAL, THE REAL PART OF HB(128,128) SHOULD
C  EQUAL TO H(128,128).
C
C-----------------------------------------------------------------
C
        VIRTUAL H(128,128),B(128),QB(128,128),HB(128,128),P(128,128)
        COMPLEX B,QB,HB
        CHARACTER*10 INFILE,FORM1,FORM2*12

        FORM1='(''$'',A)'
        FORM2='(1X,###F7.3)'


999     WRITE(7,FORM1) 'INPUT DATA FILE: '
        READ(5,'(A)') INFILE

        OPEN (UNIT=1,FILE=INFILE,STATUS='OLD',ERR=999)

        WRITE(7,FORM1) 'ENTER SIZE OF MATRIXE (4,8,16,32,64,128): '
        READ(5,*) NN
        WRITE(FORM2(5:7),'(I3)') NN
        WRITE(7,FORM1)'ORIGIN CENTERED FFT (1/0) ? '
        READ(5,*) OPTION
C
C       READ IN DATA
C
        READ(1,*) ((H(I,J),J=1,NN),I=1,NN)
C
C TRANSFORM THE ROWS OF H(I,J), STORE IN Q(I,L)
C
        DO 10 I=1,NN
                ICOUNT=0
          DO 20 J=1,NN
                ICOUNT=ICOUNT+1
                IF (OPTION.EQ.1) IC=(-1)**(I+J)
                IF (OPTION.EQ.0) IC=1
                B(ICOUNT)=H(I,J)*IC
 20       CONTINUE
                CALL FOUREA(B,NN,-1)
                M=0
                DO 30 L=1,NN
                   M=M+1
                   QB(I,L)=B(M)
 30             CONTINUE
 10     CONTINUE
C
C TRANSFORM THE COLUMNS OF QB(I,J)
C
        DO 50 J=1,NN
                ICOUNT=0
          DO 60 I=1,NN
                ICOUNT=ICOUNT+1
                B(ICOUNT)=QB(I,J)/NN          ! DIVIDED BY 1/N.
 60       CONTINUE
                CALL FOUREA(B,NN,-1)
                M=0
                DO 70 L=1,NN
                   M=M+1
                   QB(L,J)=B(M)
 70             CONTINUE
 50     CONTINUE

        DO 80 I=1,NN
          WRITE(7,FORM2) (REAL(QB(I,J)),J=1,NN)
          WRITE(7,FORM2) (AIMAG(QB(I,J)),J=1,NN)
          WRITE(7,*)
 80     CONTINUE
C
C  THE FOURIER SPECTRUM
```

```
C
        WRITE(7,*) 'THE FOURIER SPECTRUM'
        DO 85 I=1,NN
          DO 85 J=1,NN
                TEMP=QB(I,J)*CONJG(QB(I,J))
                P(I,J)=SQRT(TEMP)
85      CONTINUE
C
        WRITE(7,FORM2) ((P(I,J),J=1,NN),I=1,NN)
C
C
C--------------------------------------*
C       INVERSE TRANSFORM              |
C--------------------------------------*
C
C TRANSFORM THE ROWS OF QB(I,J), STORE IN HB(I,L)
C
        DO 100 I=1,NN
                ICOUNT=0
          DO 200 J=1,NN
                ICOUNT=ICOUNT+1
                B(ICOUNT)=QB(I,J)
200       CONTINUE
                CALL FOUREA(B,NN,1)
                M=0
                DO 300 L=1,NN
                   M=M+1
                   HB(I,L)=B(M)
300             CONTINUE
100     CONTINUE
C
C TRANSFORM THE COLUMNS OF HB(I,J)
C
        DO 500 J=1,NN
                ICOUNT=0
          DO 600 I=1,NN
                ICOUNT=ICOUNT+1
                B(ICOUNT)=HB(I,J)           ! NOT DIVIDED BY 1/N.
600       CONTINUE
                CALL FOUREA(B,NN,1)
                M=0
                DO 700 L=1,NN
                   M=M+1
                   IF (OPTION.EQ.1) IC=(-1)**(L+J)
                   IF (OPTION.EQ.0) IC=1
                   HB(L,J)=B(M)*IC*NN   !TIMES N
700             CONTINUE
500     CONTINUE

        WRITE(7,*) 'THE INVERSE FFT:'
        DO 800 I=1,NN
                WRITE(7,FORM2) (REAL(HB(I,J)),J=1,NN)
                WRITE(7,FORM2) (AIMAG(HB(I,J)),J=1,NN)
                WRITE(7,*)
800     CONTINUE


        STOP
        END
```

```
C.................................................................C
C
C       PROGRAM TO PLOT THE FOURIER SPECTRUM.
C
C       R FORTRA                                                  C
C       *FFT3D=FFT3D/W/S                                        C
C                                                                 C
C       R LINK                                                    C
C       *FFT3D=FFT3D,SDCAL(GGCAL),TVLIB/F
C.................................................................C


        PROGRAM FFT3D
        REAL H(32)
        INTEGER IZ(32,32)
        DATA IZ/1024*0/
C
        CALL PLOTS(0,0,0)
        CALL NEWPEN(256)
        CALL SYMBOL(1.5,.5,.20,'FOURIER SPECTRUM',0,16)
C
        OPEN(UNIT=2,NAME='FOR002.DAT',TYPE='OLD',FORM='UNFORMATTED')

        DO 50 I=1,32
             READ(2)  (H(K),K=1,32)
          DO 50 J=1,32
             IZ(I,J)=INT(H(J))
  50    CONTINUE

        WRITE(5,*)'INPUT ANGLE OF ROTATION (MULTIPLE OF 90)'
        READ(5,*) IROT
        ITRANS=IROT/90

        WRITE(5,*)'INPUT AZIMUTH ANGLE '
        READ(5,*) ANG
        CALL ROTATE(IZ,ITRANS)
C
C
  55    YMAX=32.
        PI=3.141592654
        RANG=ANG/180.*PI
        SCALE=.10          !*************
        COSY=COS(RANG)*SCALE
        SINY=SIN(RANG)*SCALE
        ZSCALE=.01         !*************
C
        DO 150 IX=1,32
        DO 175 IY=1,32
        Y=(FLOAT(IY-1)*SINY+FLOAT(IZ(IX,IY))*ZSCALE+1)
        X=(FLOAT(IX-1)*SCALE+FLOAT(IY-1)*COSY+1)
        IF  (IY.NE.1) GOTO 75
C       WRITE(5,*)  X,Y
        CALL PLOT(X,Y,3)
        GOTO 175
C       WRITE(5,*)  X,Y
  75    CALL PLOT(X,Y,2)
 175    CONTINUE
 150    CONTINUE
C
C
        DO 250 IY=1,32
        DO 275 IX=1,32
        Y=(FLOAT(IY-1)*SINY+FLOAT(IZ(IX,IY))*ZSCALE+1)
        X=(FLOAT(IX-1)*SCALE+FLOAT(IY-1)*COSY+1)
        IF  (IX.NE.1) GOTO 230
        CALL PLOT(X,Y,3)
        GOTO 275
 230    CALL PLOT  (X,Y,2)
 275    CONTINUE
 250    CONTINUE
C
C
        Y=1.0
        X=(31.*SCALE+1)
        CALL PLOT  (X,Y,3)
C
        Y=(FLOAT(IZ(32,1))*ZSCALE+1)
        CALL PLOT  (X,Y,2)
C
        Y=(FLOAT(IZ(1,1))*ZSCALE+1)
        X=1
        CALL PLOT(X,Y,3)
C
        CALL PLOT(1.0,1.0,2)
```

```
C
      Y=1.0
      X=(31.*SCALE+1)
      CALL PLOT(X,Y,2)

      Y=(31.*SINY+1)
      X=(31.*SCALE+31.*COSY+1)
      CALL PLOT(X,Y,2)
C
C
      Y=(31.*SINY+FLOAT(IZ(32,32)*ZSCALE+1))
      CALL PLOT(X,Y,2)
      STOP
      END

C
C.....SUBROUTINE THAT CALCULATES THE IMAGE ROTATION
C
      SUBROUTINE ROTATE(IA,ITIME)
C
C
      INTEGER IA(32,32),IB(32,32)
C
      IF(ITIME.EQ.0) RETURN
C
      DO 200 IANGL=1,ITIME
        DO 50 IX=1,32
        DO 50 IY=1,32
          IB(IY,32-(IX-1))=IA(IX,IY)
  50    CONTINUE
C
C
        DO 100 IX=1,32
        DO 100 IY=1,32
          IA(IX,IY)=IB(IX,IY)
 100    CONTINUE
 200  CONTINUE
      RETURN
      END
```

```
C
C--------------------------------------------------
C
       PRGRAM GRPAT.FOR
C
C   PLOT THE GRAY SCALED FOURIER POWER SPECTRUM.
C
C--------------------------------------------------
C
       REAL F(128)
       CHARACTER*10 FILE

       CALL MPIOPS

555    WRITE(7,*)'ENTER FFT FILE NAME'
       READ(5,'(A)') FILE

       OPEN(UNIT=2,FILE=FILE,STATUS='OLD',FORM='UNFORMATTED',ERR=555)

       WRITE(7,*) 'ENTER SIZE OF THE IMAGE (N)'
       READ(5,*) N
       WRITE(7,*)'SELECTION OF PICTURE PLANE (0,1,2)'
       READ(5,*) IP
       WRITE(7,*)'ENTER ORIGIN (IX,IY) FOR DISPLAY <BAS> '
       READ(5,*) IX,IY
C      CALL CURSOR(IX,IY)
       CALL OFF('G')
       CALL ON('P')
       CALL PEER(IP)

       DO 10 I=1,N
               READ(2)(F(J),J=1,N)
         DO 20 K=1,N
               IZ=NINT(F(K))
               CALL GRAFOT(IX+K-1,IY+I-1,IZ)
20       CONTINUE
10     CONTINUE

       STOP
       END
```

```
C
C------------------------------------------------------------------
C
C  MAIN PROGRAM: FFTCOR.FOR
C
C  PERFORMS TWO DIMENSIONAL FFT, AND CROSS CORRELATIONS
C  BETWEEN TWO FUNCTIONS (MUST BE THE SAME SIZE).
C
C-------------------------------------------------------------------
C
        VIRTUAL H1(64,64),B1(64),QB1(64,64),HB(64,64)
        VIRTUAL H2(64,64),B2(64),QB2(64,64)
        COMPLEX B1,QB1,B2,QB2,HB
        CHARACTER*10 INFIL1,INFIL2,FORM1,FORM2*12,FORM3*12
C
        FORM1='(''$'',A)'
        FORM2='(1X,###F5.1)'
        FORM3='(1X,###F5.1)'
C
        WRITE(7,FORM1) 'OUTPUT DEVICE: '
        READ(5,*) U
C
999     WRITE(7,FORM1) 'INPUT DATA FILE #1 : '
        READ(5,'(A)') INFIL1
C
        OPEN (UNIT=1,FILE=INFIL1,STATUS='OLD',ERR=999)
C
888     WRITE(7,FORM1) 'INPUT DATA FILE #2 : '
        READ(5,'(A)') INFIL2
C
        OPEN (UNIT=2,FILE=INFIL2,STATUS='OLD',ERR=888)
C
        WRITE(7,FORM1) 'SIZE OF THE MATRIX (4,8,16,32,64,128) : '
        READ(5,*) NN

        NNT=2*NN
        WRITE(FORM2(5:7),'(I3)') NNT
        WRITE(FORM3(5:7),'(I3)') NN
C
C INITIALIZE H1,H2, SO THAT THE INPUT ARRAY HAS LENGTH OF 2N.
C
        DO 5 I=1,NNT
          DO 5 J=1,NNT
              H1(I,J)=0.0
              H2(I,J)=0.0
5       CONTINUE
C
C       READ IN DATA
C
        READ(1,*) ((H1(I,J),J=1,NN),I=1,NN)
        READ(2,*) ((H2(I,J),J=1,NN),I=1,NN)
C
C TRANSFORM THE ROWS OF H(I,J), STORE IN Q(I,L)
C
        DO 10 I=1,NNT
                ICOUNT=0
          DO 20 J=1,NNT
                ICOUNT=ICOUNT+1
                B1(ICOUNT)=H1(I,J)
                B2(ICOUNT)=H2(I,J)
20        CONTINUE
                CALL FOUREA(B1,NNT,-1)
                CALL FOUREA(B2,NNT,-1)
                M=0
                DO 30 L=1,NNT
                   M=M+1
                   QB1(I,L)=B1(M)
                   QB2(I,L)=B2(M)
30              CONTINUE
10      CONTINUE
C
C TRANSFORM THE COLUMNS OF QB(I,J)
C
        DO 50 J=1,NNT
                ICOUNT=0
          DO 60 I=1,NNT
                ICOUNT=ICOUNT+1
                B1(ICOUNT)=QB1(I,J)/NNT           ! DIVIDED BY 1/N.
                B2(ICOUNT)=QB2(I,J)/NNT
60        CONTINUE
                CALL FOUREA(B1,NNT,-1)
                CALL FOUREA(B2,NNT,-1)
                M=0
                DO 70 L=1,NNT
```

```
                   M=M+1
                   QB1(L,J)=B1(M)
                   QB2(L,J)=B2(M)
70              CONTINUE
50        CONTINUE
C
C-------------------------------------*
C        INVERSE TRANSFORM            |
C-------------------------------------*
C
C CALCULATE THE NORMALIZING FACTOR SUM
C
        SUM1=0.0
        SUM2=0.0
        DO 750 I=1,NNT
          DO 750 J=1,NNT
                SUM1=SUM1+H1(I,J)*H1(I,J)
                SUM2=SUM2+H2(I,J)*H2(I,J)
750     CONTINUE
C
        SUM1=SQRT(SUM1)
        SUM2=SQRT(SUM2)
        SUM=SUM1*SUM2
C
C TRANSFORM THE ROWS OF QB(I,J)*CONJG((QB2(I,J)), STORE IN HB(I,L)
C
C EACH ENTRY IS DIVIDED BY THE NORMALIZING FACTOR.
C
        DO 100 I=1,NNT
                ICOUNT=0
          DO 200 J=1,NNT
                ICOUNT=ICOUNT+1
                B1(ICOUNT)=(QB1(I,J)*CONJG(QB2(I,J)))/SUM
200       CONTINUE
                CALL FOUREA(B1,NNT,1)
                M=0
                DO 300 L=1,NNT
                   M=M+1
                   HB(I,L)=B1(M)
300             CONTINUE
100     CONTINUE
C
C TRANSFORM THE COLUMNS OF HB(I,J)
C
        DO 500 J=1,NNT
                ICOUNT=0
          DO 600 I=1,NNT
                ICOUNT=ICOUNT+1
                B1(ICOUNT)=HB(I,J)*NNT             ! TIMES N   ? .
600       CONTINUE
                CALL FOUREA(B1,NNT,1)
                M=0
                DO 700 L=1,NNT
                   M=M+1
                   IF (OPTION.EQ.1) IC=(-1)**(L+J)
                   IF (OPTION.EQ.0) IC=1
                   HB(L,J)=B1(M)*IC*NNT      !TIMES N
700             CONTINUE
500     CONTINUE

        WRITE(U,'(A,F10.5)') ' THE LARGEST CORRELATION FUNCTION = ',
     +                      REAL(HB(1,1))
        WRITE(U,*) 'THE CORRELATION FUNCTION: '
        DO 800 I=1,NNT
                WRITE(U,FORM2) (REAL(HB(I,J)),J=1,NNT)
                WRITE(U,FORM2) (AIMAG(HB(I,J)),J=1,NNT)
                WRITE(U,*)
800     CONTINUE

        STOP
        END
```

```
C
C-----------------------------------------------------------
        PROGRAM CROCOR
C
C       PERFORMS CROSS CORRELATION BETWEEN TWO FUNCTIONS.
C       ONLY ONE FUNCTION IS COMPUTED IN THIS PROGRAM, IT
C       IS (N/2,N/2). THE INPUT MATRICES MUST BE THE SAME
C       SIZE.
C
C       JULY, 20, 1987.
C-----------------------------------------------------------
C
        DIMENSION A1(30,30),A2(30,30),B1(30,30),B2(30,30)
        CHARACTER*10 FILE1,FILE2,FORM1,FORM2
        DATA FORM1/'(''$'',A)'/

        WRITE(7,FORM1) 'ENTER FILE #1 : '
        READ(5,'(A)') FILE1
        WRITE(7,FORM1) 'ENTER FILE #2 : '
        READ(5,'(A)') FILE2

        OPEN(UNIT=1,FILE=FILE1,STATUS='OLD',FORM='UNFORMATTED')
        OPEN(UNIT=2,FILE=FILE2,STATUS='OLD',FORM='UNFORMATTED')
C
C   INPUT DATA
C
        DO 5 I=1,30
                READ(1)  (A1(I,J),J=1,30)
                READ(1)  (A2(I,J),J=1,30)
                READ(2)  (B1(I,J),J=1,30)
                READ(2)  (B2(I,J),J=1,30)
5       CONTINUE
C
C   SUM
C
        SUM1=0.0
        SUM2=0.0
        SUM3=0.0
        SUM4=0.0
        DO 10 I=1,30
          DO 10 J=1,30
                SUM1=SUM1+A1(I,J)*A1(I,J)
                SUM2=SUM2+A2(I,J)*A2(I,J)
                SUM3=SUM3+B1(I,J)*B1(I,J)
                SUM4=SUM4+B2(I,J)*B2(I,J)
10      CONTINUE
C
        SUM5=0.0
        SUM6=0.0
        DO 100 I=1,30
          DO 100 J=1,30
                        SUM5=SUM5+A1(I,J)*B1(I,J)
                        SUM6=SUM6+A2(I,J)*B2(I,J)
100     CONTINUE
C
        COR1=SUM5/SQRT(SUM1*SUM3)
        COR2=SUM6/SQRT(SUM2*SUM4)
C
        TYPE*,COR1,COR2
C
        STOP
        END
```

References

Ahuja, N. and Rosenfield, A. 1981 "Mosaic models for textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jan., pp. 1-111.

Bartels, P., et. al. 1969 "Cell recognition from line scan transition probability profiles," *Acta Cytol.*, vol. 13, pp. 210-217.

Bell, C. B. 1962 "Mutual information and maximal correlation." *Ann. Math. Statist.*, vol. 43, pp. 587-595.

Chetverikov, D. 1984 "Measuring the degree of texture regularity," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 80-82.

Carlotto, M. J. 1984 "Texture classification based on hypothesis testing approach, " *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 93-96.

Chen, C. H. 1973, *Statistical Pattern Recognition,* Hayden Book Company, Inc., Rochelle Park, N. J.

Chien, Y. P. and Fu, K. S. 1974 "Recognition of X-ray picture patterns," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp. 145-156.

D'Astous, F. and Jernigan, M.E. 1984 "Texture discrimination based on detailed measures of the power spectrum," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 83-86.

Darling, E. M. and Joseph, R. D. 1968 "Pattern recognition from satellite altitudes," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-4, pp. 38-47.

Deguchi, K. and Morishita, I. 1976 "Texture characterization and texture-based image partitioning using two-dimensional linear estimation techniques," presented at *U.S. - Japan Cooperative Science Program Seminar on Image Processing in Remote Sensing* (Washington, D.C.), 1-5.

Egbert, D., McCauley J., and McNaughton, J. 1973 "Ground pattern analysis in the Great Plains," *Semi-Annual ERTS A Investigation Rep.*, Remote Sensing Laboratory, University of Kansas.

Gagalowicz, A. and Ma, S. D. 1984 "Synthesis of natural textures on 3-D surfaces," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. pp. 1209-1212.

Gonzalez, R. C. and Wintz, P. 1977 *Digital image processing.* Addison-Wesley Publishing Company, Inc.

Goodchild, M. F. and Mark, D. M. 1987 "The fractal nature of geographic phenomena," *Annals of the A. A. G.*, vol. 77, pp. 265-278.

Gramenopoulas, N. 1973 "Terrain type recognition using ERTS-1 MSS images," *Rec. Symp. Significant Results Obtained from the Earth Resources Technology Satellite,* NASA SP-327, pp. 1229-1241.

Gross, G. R. and Jain, A. K. 1983 "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jan., pp. 25-39.

Hall, E. L. 1979, *Computer Image Processing and Recognition*, Academic Press.

Haralick, R. M. 1971 "A texture-context feature extraction algorithm for remotely sensed imagery," *Proc. 1971 IEEE Decision and Control Conf.,* pp. 650-657.

Haralick, R. M., Shanmugam K., and Dinstein I. 1973 "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-3, pp. 610-621.

Haralick, R. M., K. Shanmugam 1974 "Combined Spectral and Spatial Processing of ERTS," *Remote Sensing of Environment*, vol. 3, pp. 3-13.

Haralick, R. M. 1979 "Statistical and Structural Approaches to Texture," *Proceedings of the IEEE*, vol. 67, pp. 786-804.

Hord, R. M. 1982, *Digital Image Processing of Remote Sensed Data,* Academic Press.

Hord, R. M. 1986, *Remote Sensing, Methods and Applications*, John Wiley & Sons.

Horning, R. J. and Smith, J. A. 1973 "Application of Fourier analysis to multispectral/spatial recognition," presented at *Management and Utilization of Remote Sensing Data ASP Symposium*, Sioux Falls.

Hsu, S. Y. 1978 "Texture-tone analysis for automated land-use mapping," *Photogrammetric Engineering & Remote Sensing*, vol. 44, No. 11, pp. 1393-1404.

Ikonomopoulos, et. al. 1984 "A directional filtering approach to texture discrimination," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp.87-89.

Jau, Y. C. et. al. 1984 "Time series modeling for texture analysis and synthesis with applications to cloud field morphology study," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 1219-1221.

Jensen, J. R. 1979 "Spectral and textural features to classify elusive land cover at the urban fringe," *The Professional Geographer,* vol. 31, pp. 400-409.

Jensen, J. R. 1986 *Introductory digital processing, a remote sensing perspective.* Englewood Cliffs, N. J.: Prentice-Hall, Inc.

Jensen, J. R. and Toll, D. J. 1982 "Detecting residential land-use development at the urban fringe," *Photogrammetric Engineering and Remote Sensing,* vol. 48, pp. 629-643.

Jernigan, M. E. and D'Astous, F. 1984 "Entropy-based texture analysis in the spatial frequency domain," *Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-6, NO. 2, pp. 237-243.

Kaizer, H. 1955 "A quantification of textures on aerial photographs," Boston University Research Laboratories, Boston University, Boston, MA, Tech. Note 121, AD 69484.

Kashyap, R. L. and Khotanzad, A. 1984 "A stochastic model based techniques for texture segmentation," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp.1202-1205.

Kirvida, L. 1976 "Texture Measurements for the automatic classification of imagery," *IEEE Trans. Electromagnet. Computer,* vol. 18, pp. 38-42.

Kirvida, L. and Johnson, G. 1973 "Automatic interpretation of ERTS data for forest management," *Symp. on Significant Results Obtained from the Earth Resources Technology Satellite,* NASA SP-327.

Kotz, S. 1966, *Recent Result in Information Theory,* Methuen's Review Series in Applied Probability, vol. 5.

Lendaris, G. and Stanley, G. 1969 "Diffraction Pattern sampling for automatic pattern recognition," *SPIE Pattern Recognition Studies Seminar Proc.* June 9-10, pp. 127-154.

Lillesand, T. M. and Keifer, R. W. 1979, *Remote Sensing and Image Interpretation.* John Wiley & Sons.

Ma, S. and Gagalowics, A. 1984 "A parallel method for natural texture synthesis," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 90-92.

Matheron, G. 1967. *Elements Pour Une Theorie Des Milieus Poreux.* Paris, France: Masson.

McCormick, B. H. and Jayaramamurthy, S. N. 1974 "Time series model for texture synthesis," *International Journal of Computer Information Science*, vol. 3, no. 4, pp. 329-343.

Pressman, N. J. 1976 "Markovian analysis of cervical cell images," *J. Histochem. Cytochem.*, vol. 24, no. 1, pp. 128-144.

Ramapriyan, H. K. 1972 "Spatial frequency analysis of multispectral data," *Remote Sensing of Earth Resources*, vol. 1, pp. 621-644.

Rosenfield, A. and Kak, A. C. 1982, *Digital Image Processing,* 2nd ed., vol. 1 and vol. 2, Academic Press.

Rosenfield, A. and Thurston, M. 1971 "Edge and curve detection for visual scene analysis," IEEE Trans. Comput., vol. C-20, pp. 562-569.

Rugg, D. S. 1979, *Spatial Foundations of Urbanism,* second edition. Wm. C. Brown Company Publishers, Dubuque, Iowa.

Sato, M. and Ogata, M. 1984 "Texture analysis by the self-organization method," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 1213-1215.

Schacter, B. J., Rosenfield, A. and Davis, L.S. 1978 "Random Mosaic Models for textures," *IEEE Tran. on Sys., Man, and Cyn.*, Sept., pp. 694-702.

Schowengerdt, R. A. 1983, *Techniques for Image Processing and Classification in Remote Sensing.* Academic Press.

Serra, J. 1974 "Theoretical bases of the Leitz texture analyses system," *Leitz Sci. Tech. Inform.*, Supplement 1, 4, pp. 125-136.

Sun, C. and Wee, W. G. 1983 "Neighboring gray level dependence matrix for texture classification," *Computer Vision, Graphics and Image Processing,* vol. 23, pp. 341-352.

Sutton, R. and Hall, E. 1972 "Texture measure for automatic classification of pulmonary disease," *IEEE Trans. Comput.,* vol. C-20, no. 1, pp. 667-676.

Tamura, H., Mori, S. and Yamawaki, T. 1978 "Texture features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8, No. 6, pp. 461-472.

Toriwaki, J. I. et. al. "Adjacency graphs on a digitized figure set and their application to texture analysis," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 1216-1218.

Tou, J. T. and Chang, Y. S. 1976 "An approach to texture pattern analysis and recognition," *Proc. 1976 IEEE Conf. on Decision and Control.*

Unser, M. 1984 "Local linear transform texture analysis," *Proceedings, Seventh International Conference on Pattern Recognition,* vol. 1, pp. 1206-1208.

Werman, M. and Peleg, S. 1984 "Multiresolution texture signature using min-max operators," *Proceedings, Seventh International Conference on Pattern Recognition,* vol.1, pp. 97-101.

Weszka, C. Dyer, and A. Rosenfield 1976 "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, and Cybern.*, vol. SMC-6, no. 4, pp. 269-285.

Yeates, M. and Garner, B. 1980, *The North American City,* third edition, Harper & Row, Publishers.