ON TEXTURE IMAGE ANALYSIS USING FRACTAL GEOMETRY BASED FEATURES

Nirupam Sarkar

Computer Vision and Pattern Recognition Unit,

Indian Statistical Institute,

203, B. T. Road,

Calcutta 700 035,

INDIA.

A thesis submitted to the *Indian Statistical Institute*

in partial fulfilment of the requirements

of the degree of

DOCTOR OF PHILOSOPHY

Acknowledgments

I owe Prof. B. B. Chaudhuri, my supervisor, a tremendous debt. His guidance and constant encouragement helped me keeping research objectives clear and keeping myself motivated throughout the years of my tenure as a research fellow. His ideas in the subject left a long lasting influence on my own thinking.

Prof. D. Dutta Majumder, Emeritus Professor, allowed me to use the resources in the Knowledge Based Computer Systems (KBCS).

In the course of my research career I was benefited from the knowledge, help, friendship, and encouragement of many people. Dr. Swapan K. Parui, Amar Mukherjee, Ujjwal Bhattacharjee, and Pulak Kundu deserve special thanks for all the help they have extended. I am indebted to Dr. Amarnath Gupta for presenting the Brodatz album which turns out to be an asset for me. I wish to thank one and all of the KBCS Center for the nice environment they provided for the work.

I am grateful to Prof. Anil K. Jain of Michigan State University, USA and Prof. Oliver D. Faugeras of INRIA, Sophia-Antipolis, France for discussions I had with them during their visits to our Institute.

I am thankful to my parents for their constant support and encouragement.

Finally, I wish to thank Dean of Studies for financial support throughout my tenure and Head, Reprography Unit for preparation of the manuscript.

November 24, 1994

NIRUPAM SARKAR
CVPR Unit, ISI
Calcutta.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Visual texture

Texture is a property to characterize a region of a scene. A set of natural texture images is shown in Fig. 1.1. A specific texture may be generated due to certain organization of several objects in a region, or due to the reflectance pattern caused by color variation or unevenness of an object surface. Since texture provides a lot of information of a region, texture analysis and synthesis are important components of digital image processing.

It is difficult to provide a formal definition of texture although we perceive and recognize texture rather easily. According to Sklansky [152] "A region in an image has a constant texture if a set of local statistics or other local properties of the picture functions are constant, slowly varying, or approximately periodic". Haralick *et al.* [59] stated "Texture is an innate property of virtually all surfaces. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment". For textured regions we may get different interpretations at different distances and at different degrees of visual attention. At standard distance with normal attention, a notion of macro-regularity is obtained that is characteristic of the particular texture. Even ho-

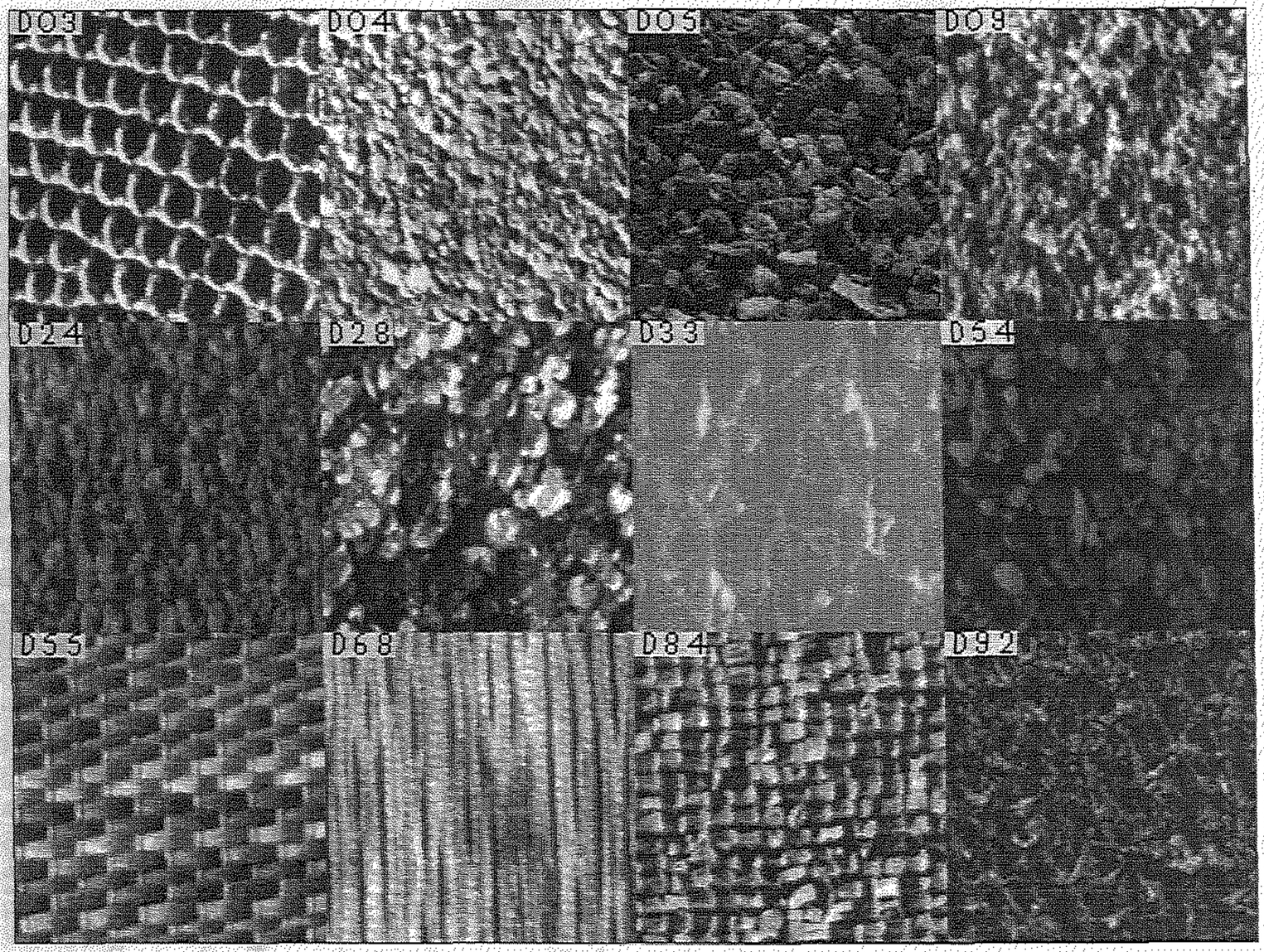


Figure 1.1: Natural textures (from Brodatz [17] album)

mogeneous regions and edges, when viewed closely and attentively are sometimes found to constitute of meaningful objects. For tonal regions, on the other hand, the interpretations remain largely unaffected with the distance and degree of attention. Moreover, tonal regions have edges at the 'borders' of the regions, while edges are normally abundant in the 'interior' of a textured region.

Early researchers characterized a textured pattern as one exhibiting many repetitive variations of similar 'basic elements' each of which is small relative to the size of the textured region; or as large number of elements, each to some degree visible, and, on the whole, densely and evenly placed over the field of view. The 'basic elements' which are supposed to be the main constituent of texture is called 'texture elements' or *texels*.

Properties of texture : Surface texture is a combination of 2-D and 3-D properties of the objects in a scene. In general, image texture possesses qualities like uniformity, density, coarseness, roughness, regularity, linearity, directionality, repetition frequency, and phase. These qualities play important roles in describing a texture. Some of these qualities are often interrelated. From the quantitative point of view, six basic textural properties proposed by Tamura *et al.* [156] are described below.

(a) *Coarseness* : The coarseness is an important textural feature and has been investigated since early studies [143,62]. It is a measure of relative size of the texels of a given texture. For example, in Fig 1.1, D05, D28 and D54 are coarse textures while D9 and D84 are fine textures. Larger the size of the texels, the coarser is the texture. Hayes *et al.* [62] developed a method to provide a measure of coarseness of a given texture. The essence of this method is to pick up a large size as the best when both coarse and micro textures are present, but to pick a small size when only fine texture is present. This procedure can be summarized by the following steps.

(i) Take averages at every point over neighborhoods whose sizes are powers of two. The average over the neighborhood of size $2^k \times 2^k$ at the point

(x, y) is

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} f(i,j)/2^{2k}$$

where $f(i,j)$ is the gray value at (i,j).

(ii) For each point, take differences between pairs of averages corresponding to pairs of non-overlapping neighborhoods just on opposite sides of the point in both horizontal and vertical orientations. For example, the difference in the horizontal direction is

$$E_{k,h}(x,y) = \mid A_k(x+2^{k-1},y) - A_k(x-2^{k-1},y) \mid$$

(iii) At each point, pick the best size which gives the highest output value of E. The coarseness of the texture is measured as the average of the best sizes over the whole image.

(b) *Contrast* : Contrast is a measure of sharpness i.e. variation of gray values among neighboring pixels. Contrast of an image depends on the following factors.

(i) dynamic range of gray-levels,

(ii) polarization of the distribution of black and white on the gray-level histogram,

(iii) sharpness of the edges,

(iv) period of repeating patterns.

Haralick *et al.* [59] proposed the estimate of contrast as :

$$f_2 = \sum_{k=0}^{n} n^2 \left[\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p(i,j) \right] \qquad \mid i-j \mid = n$$

where $p(i,j)$ is the (i,j)th entry of the $n \times n$ gray-level spatial dependence matrix. Here f_2 is the moment of inertia about the main diagonal of the matrix $p(i,j)$. As noted by Weszka *et al* [170] it can also be obtained from the histogram of the frequencies of gray-level differences. However, Tamura *et al.* proposed the measure of contrast as a function of the variance and the kurtosis of the gray level histogram. In Fig 1.1, D03 and D84 have high contrast on the other hand D33 and D92 have low contrast.

(c) *Directionality* : Textures which exhibit some degree of orientation specificity at each point are called directional textures. Directional textures can be of two types, monodirectional and multidirectional. In addition, textures may have a slowly space varying dominant direction. D68 and D24 are monodirectional textures while D03, D84 and D55 are multidirectional textures with two dominant directions.

Directionality can be estimated by computing the gradient represented by the magnitude $|\Delta G|$ and direction θ defined as

$$\begin{aligned} |\Delta G| &= (|\Delta_H| + |\Delta_V|)/2 \\ \theta &= tan^{-1}\left(\frac{|\Delta_H|}{|\Delta_V|}\right) + \frac{\pi}{2} \end{aligned}$$

where Δ_H and Δ_V are the horizontal and vertical differences measured by Sobel, Prewitt or Canny edge operator.

Tamura *et al.* estimated the directionality in the following way. At first, a direction histogram is obtained by quantizing θ and counting the points with gradient magnitude $|G|$ exceeding the threshold value t. Then the peak(s) of the direction histogram are located. The directionality measure is given as a function of number of peaks and peak strength of each of them. Chaudhuri *et al.* [21] provided a simple measure by gathering directional evidence as in Hough transform.

(d) *Line-Likeness* : By "line-likeness" we mean an element of texture that is composed of lines. The line-like structures are

(i) thin,

(ii) piecewise linear,

(iii) the gray level contrast on either side of the structure is significant and local edge directions of either side of the structure are 180° out of phase with each other.

(iv) the gray levels of the pixels lying on the structure are locally homogeneous (with respect to edge gradient and local edge direction).

To estimate line-likeness, at first, the linear structures are to be found. Then direction co-occurrence matrix is constructed (the directions should be quantized at some specific interval). An element of the direction co-occurrence matrix $P_d(i,j)$ is defined as the relative frequency with which two neighboring cells separated by a distance d, one with the direction i and the other with direction j. The measure of like-likeness can be given as a function of moment of inertia about the main diagonal of $P_d(i,j)$. More the moment of inertia, less will be the line-likeness. D68 and D03 have more line-likeness than other textures in the mosaic of Fig 1.1.

(e) *Regularity* : This is a property depicting variation of placement of texture elements (texels). It is evident that the variations of placement and the texels (size and shape) of the given texture will reduce its regularity. D84 and D55 are regular textures. Several textural features may be considered to measure regularity. Tamura *et al.* used coarseness, contrast, directionality and line-likeness for this purpose. The image is partitioned into subimages (overlapping or non-overlapping) and variation of each of the features over the subimages are considered. A linear combination of the variations is used as the measure of regularity.

(f) *Roughness* : Roughness of a texture is a measure of unevenness of the image gray value surface. D03, D04, D09 and D84 are rough textures where as D33 and D54 are the smooth ones. Tamura *et al.* emphasized on coarseness and contrast, and approximated a measure of roughness as the sum of the aforesaid features. Pentland [129] made an observation that the fractal dimension (FD) shows a strong correlation with human judgment of surface roughness. Different aspects of Fractals and a variety of methods of estimating fractal dimension (FD) are studied in Chapter 2. A new method for estimation of FD is also presented there.

The six basic properties described above are not independent. Tamura *et al.* has shown that some of them are highly correlated. To compute the correlation between two properties they used Spearman's coefficient of rank correlation r_s which can be written as

$$r_s = 1 - \frac{6}{n^3 - n} \sum_{i=1}^{n} d_i^2$$

where d_i is the difference between the ranks assigned to the i-th texture in two measurements and n is the number of textures.

There exists considerable correlation between coarseness, contrast, and roughness as well as between directionality and line-likeness. The correlation coefficient between coarseness with contrast and roughness are 0.658 and 0.535, respectively and that of between contrast and roughness is 0.504. Directionality and line-likeness have a high correlation of 0.923 [156].

1.2 Texture related problems and applications

1.2.1 Texture related problems

The texture related problems can be grouped broadly as follows:

1. *Texture Classification* : The goal of texture classification is to classify textures into one of the several learned classes. Texture classification involves feature selection and extraction as well as design of classification algorithm. The features used for texture classification is described in Section 1.3. A variety of algorithms are available for classification. Many classification methods are based on probability distribution models for the classes of interest. Recently, neural network and genetic algorithm based techniques are being tried. Irrespective of the algorithm chosen, the following steps are taken.
   (a) The set of texture classes are decided.
   (b) Representative pixels from each class are chosen from known scene. These pixels constitute the training data. The features are detected and parameters for the particular classifier are estimated from each of the training data sets.

(c) Using trained classifier, an unknown sample is classified into one of the desired classes. A tabular representation called confusion matrix, which summarizes the result of classification is produced.

Perhaps this is the area of texture analysis where maximum research publications are available [59,170,34,174,30,128].

2. *Texture Segmentation* : The task of texture segmentation is to segment and label a given image into homogeneously textured regions. These segmented and labeled regions are then used for representation, description, and recognition. There exist two main classes of approaches of texture segmentation namely region-based approach and edge-based approach. In region-based approach, regions with uniform textures are identified and labeled [125,36,66,72]. In edge-based approach, the edges due to difference in texture are detected [82].

   This thesis deals mainly with texture segmentation using a region based approach. In general, texture segmentation involves three major steps : i) Feature selection and extraction, ii) feature domain smoothing, *and* iii) feature domain segmentation. A brief review of feature selection is presented in Section 1.3, while feature domain smoothing and feature domain segmentation are described in Chapter 3.

3. *Texture Synthesis* : Computer-generated images can appear highly realistic if the surfaces are rendered with adequate textures. Texture synthesis can be useful in computer graphics and animation, design of textiles, wallpaper and laminate finishing, camouflaging techniques, texturing of missing or incomplete portions etc. There are several approaches to texture synthesis. They are based on mathematical model or derived from texture analysis techniques.

   Use of formal grammar is an approach of texture synthesis. Fu and Lu [102] used a stochastic tree grammar approach for synthesis and analysis of texture. Another approach is the growth model introduced by Yokoyama and Haralick [173]. Ahuja [2], Ahuja and Rosenfeld [3] and Schacter *et al.* [147]

used mosaic models for the texture modelling and synthesis. Texture synthesis based on computer drafting was investigated by Mezei *et al* [113]. Yokoyama and Haralick [172] as well as Cross and Jain [38] used Markov process for texture generation. The underlying model is called Markov Random Field(MRF) model. A variation of this model using Gaussian Markov Random Field (GMRF) was described by Chellappa *et al.* [25] and Chellappa [24]. Another powerful class of texture synthesis is based on the fractal geometry [106,165,126].

4. *Shape from Texture* : Normally, a digital image is the 2-D projection of a 3-D scene. When a texture-rich surface is inclined, the surface texture is apparently distorted according to the inclination. The role of texture as a basis for recovery of surface orientation was first investigated by Gibson [52]. The method to estimate this surface orientation from the apparent texture distortion is called *shape-from-texture.*

   The shape-from-texture algorithms are based on different surface cues such as texture gradient, converging lines, normalized texture property map and shape distortion. Bajcsy and Lieberman [9], Ohta *et al.* [122] and Alloimonos and Swain [4] estimated shape from texture using the texture gradient cue, whereas Nakatani *et al.* [120] and Kender [88] recovered shape information from converging lines computed from the given scene. Kender [87] and Ikeuchi [69] investigated the potential of Normalized Texture Property Map (NTPM). On the other hand, the shape distortion parameters were used by Witkin [171] as well as Walker and Kanade [167].

5. *Texture Perception* : Human texture perception is mostly concerned with identifying factors that mediate preattentive texture discrimination. Preattentive visual systems cannot process complex forms. Yet they can almost instantaneously, without effort or scrutiny, detect differences in a few local conspicuous features, regardless of where they occur. Julesz [77] first proposed that the first- and the second-order statistics are major criteria for texture discrimination. Later Julesz [76] proposed the theory of *textons*, where he stated that the textons are the main features for preattentive tex-

ture discrimination. Textons are basically elongated blobs (e.g., rectangles, ellipses, or line segments) with specific properties, including color, angular orientation, width, and length. The end-of-lines (terminators) and crossings of line segments are also considered as textons. Many researchers studied and developed this concept [78,12,75] in connection with visual perception of texture.

While most psychological experiments on texture perception focus on the preattentive aspects, many practical problems may require attentive analysis of texture. Thus it is important to look for high level features for texture description. Tamura *et al.* [156] performed some experiments with the textures of the Brodatz album. A psychological experiment was carried out where the subjects were asked to compare pairs of textures based on the six features described above. Their results and the results obtained by Amadasun and King [5] and Rao and Lohse [137] reveal that the psychovisual and computer simulated results are highly correlated.

1.2.2 Applications of texture analysis

Texture analysis methods have been utilized in a variety of application domains. Here we review briefly the role of texture analysis in automated inspection, medical image processing, document processing, and remote sensing.

The applications of texture processing to automated inspection problems include defect detection in textiles, automated carpet wear, and automobile paint. Dewaele *et al.* [41] used signal processing methods to detect point defects in texture images. Chetverikov [29] defined a simple window differencing operator to the texture features obtained from simple filtering operation. Chen and Jain [27] used a structural approach to detect defect in textural images. In the area of quality control of texture images, Siew *et al.* [151] proposed a method for assessment of carpet wear. Jain *et al.* [73] used texture features from a bank of Gabor filters to classify the quality of painted metallic surface.

Recently, texture analysis approaches are being used in electronics industry. Different types of defects in semiconductor wafer are detected using texture analysis methods [136].

Several applications of texture analysis are reported in the area of medical image processing. Sutton and Hall [155] discuss the classification of pulmonary disease using texture features. They used isotropic contrast measure, directional contrast measure and Fourier domain energy sampling to distinguish normal lung and diseased lung. Harmes *et al.* [61] used structural features like texture microedges and "textons" along with other features to diagnose leukemic malignancy in samples of stained blood cells. Landeweerd and Gelsema [94] extracted various first- and second-order statistics to classify different types of white blood cells. Chen *et al.* [26] used fractal texture features to classify ultrasound images of liver and also to do edge enhancement in chest X-rays. Lundervold [103] used fractal texture features in combination with other features to analyze ultrasound images of the heart.

Another area of application of texture analysis is in document processing, where the applications range from postal address recognition to analysis and interpretation of maps.

Texture analysis has been extensively used to classify remotely sensed images. Land use classification where homogeneous regions with different types of terrain should be identified is an important application area. Haralick [59] as well as Rignot and Kwok [140] analyzed SAR images using textural features based on gray-level co-occurrence matrices. Features based on fractal dimension, Gabor filters, and MRF were also used in the classification of remotely sensed images.

1.3 Texture analysis : A survey

There exist a large number of papers on texture analysis including several survey papers. Haralick [57] compiled a comprehensive survey of statistical and structural approaches. Van Gool *et al.* [161] and Tuceryan and Jain [160] also published sur-

veys of texture analysis techniques. Texture analysis approaches are popularly grouped into two classes, *viz.*, 1) Statistical approaches and 2) Structural approaches. Which are surveyed below. Section 3.2 contains some complementary materials.

1.3.1 Statistical approaches

Histogram based approach

The gray level histogram is a plot of global frequency of occurrence of gray level i in the image n_i against i. The normalized frequency h_i is estimated by dividing n_i by the number of pixels in the image.

Histogram may also be computed on other properties of the image such as edge strength against direction, run length against direction and so on. Several statistical features, such as mean, standard deviation, kth moment, and entropy, may be computed from the image histogram.

One simple way of comparing two images is through their (gray value) histogram similarity. The similarity is measured by the Kolmogorov-Smirnov (KS) test on the cumulative distribution functions. Nagao *et al.* [119] used KS test for agricultural land use classification of aerial photographs. Muerle [116] used KS test for image segmentation.

Co-occurrence Statistics based approach

Spatial gray level co-occurrence can estimate the image properties related to second order statistics. The co-occurrence $P(i,j)$ of gray levels i and j for an image I is defined as the number of pixel pairs having gray levels i and j, respectively, where the pixels are in a fixed spatial relationship, say a fixed distance apart, along a fixed direction. Julesz [77] first used these properties in texture discrimination experiments. A brief outline of the features based on co-occurrence statistics [59]

are described in Section 3.2.

Haralick *et al.* [59], Weszka *et al.* [170] and Zucker and Terzopoulos [177] used features from co-occurrence matrix for texture classification. Pavlidis and Chen [125], and Conners *et al* [36] achieved good segmentation results using the features obtained from co-occurrence statistics.

Fourier Transform based approach

Textures can be analysed in the frequency domain by Fourier transform of the image. Let $f(x,y)$ be a texture image of size $M \times N$. Its 2-D discrete Fourier transform $F(u,v)$ is defined by

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) exp^{2\pi\sqrt{-1}(ux/M+vy/N)} \quad ; \quad u,v = 1,2,\ldots,M-1$$

The power spectrum which represents the strength of each spatial frequency is obtained by

$$P(u,v) = \mid F(u,v) \mid^2$$

Lowitz [101] and Carlotto [18] suggested the use of local histogram for feature extraction. Bajcsy [8] computed two features from the power spectrum $P(r,\theta)$ in the polar coordinates and later used them to estimate the coarseness and directionality of the texture. Bajcsy and Liebermann [9] used power spectrum in polar co-ordinate system to recover shape from texture. Pentland [129] computed the discrete Fourier transform of each block of 8×8 pixels of an image and determined the power spectrum, in order to use it to estimate the fractal dimension of the image.

Although Fourier transform is the most popular transform, Discrete Cosine, Walsh, Hadamard, and Haar transforms can also be used for texture analysis purpose. Kirvida [90] compared fast Fourier, Hadamard, and Slant transform based textural features for analysis of aerial images.

Autoregressive Models

The linear dependency in textures can be conveniently exploited by autoregressive (AR) models. AR models were first used by McCormick and Jayaramamurthy [111] to synthesize textures. Here textures are viewed as two dimensional local variations of intensity levels. These variations are considered to have varying degrees of randomness associated with them. One can quantify such characterization by modeling the spatial interaction among pixels using two-dimensional models like Simultaneous Autoregressive (SAR) [83] and Circular Autoregressive (CAR) model [81].

Kashyap and Khotanzad [82] as well as Khotanzad and Chen [89] used several parameters from the SAR and CAR models for texture segmentation. Wan *et al.* [168] proposed an extension of CAR model. Mao and Jain [108] studied the performance of different SAR (second and higher order neighbor) models for feature extraction and later used them for texture segmentation.

Texture energy approaches

Laws [95] proposed an approach to measure texture energy in the spatial domain. His approach of texture characterization consists of two steps. First the image is convolved with a set of filters having a small region of support in the spatial domain. These filters are called microtexture masks, and their outputs are called microtexture features. These microtexture masks are designed to act as matched filters for certain types of quasiperiodic variations commonly found in textured images. The convolution masks are intended to be sensitive to visual structure such as edges, ripples, and spots. Next, macrostatistical features are obtained over very large window. A detailed description of this method is provided in Section 3.2.

Pietikainen *et al.* [134] used Laws' masks for texture classification. Hsiao and Sawchuk [66,67] used these features for texture segmentation.

Autocorrelation approaches

An important property of *many* textures is the repetitive nature of the placement of texture elements in the image. The autocorrelation function of an image can be used to assess the amount of the regularity as well as the fineness/coarseness of the texture present in the image. *This function* is related to the size of the texture primitive. If the texture is coarse, then the autocorrelation function will drop slowly; otherwise, it will drop very rapidly.

Mathematical morphology based approaches

Mathematical morphology is a powerful approach for texture analysis. The morphological approach to the texture analysis of binary images was proposed by Matheron [109] as well as Serra and Verchery [150]. Their approach is the basis of the systems called Leitz texture analyzer and Cyto computer [149,117]. Morphological texture analysis has found wide applications in material science and biomedical image processing.

Mathematical morphology involves some set operations on the image. The operations are done using a 'structuring element'(H) which is a set of pixels having a specific shape and size. Two basic morphological operations are 'dilation'($F \oplus H$) and 'erosion'($F \ominus H$) of the image $F(x,y)$ by H.

Composition of erosion and dilation determine two other important morphological operations which are idempotent and are duals of one another : The openings of F by H is defined by $(F \ominus H) \oplus H$ and the closing of F by H is defined by $(F \oplus H) \ominus H$.

Sternberg [154] has extended the morphological definition of erosion and dilation to gray-tone image I by gray-tone structuring element H as

$$J(r,c) = \min_{i,j}\{I(r+i,c+j) - H(i,j)\} = (I \ominus H)(r,c)$$
$$J(r,c) = \max_{i,j}\{I(r+i,c+j) + H(i,j)\} = (I \oplus H)(r,c)$$

As in the two tone case, the gray tone opening is defined as gray tone erosion followed by gray tone dilation. Similarly, gray tone closing is defined as gray tone dilation followed by gray tone erosion. Commonly used gray-tone structuring elements include rods, cones, paraboloids, and hemispheres.

Peleg *et al.* [128] used gray-tone erosion and dilation to estimate fractal dimension of an image that is subsequently used for texture classification. Werman and Peleg [169] gave a fuzzy set generalization to the morphological operators for texture classification. They used line segments of various lengths and orientations as structuring elements and defined five textural features, *viz.*, 'sum of the entire matrix', 'sum of rows over all angles', 'sums of columns over all lengths', 'a row measure', 'a column measure', and 'a gradient measure'. These features were derived from the morphologically operated images. Mayer [112] as well as Lipkin and Lipkin [99] used morphological features for biomedical texture analysis. Di Gesu and Tripiciano [43] used Minimum Erosion, Avaraged Erosion, Δ-Erosion, and Exact Erosion for texture segmentation.

Fractal based features

Many natural surface have roughness and self-similarity at different scales. Fractal geometry is useful in modeling properties in images. Mandelbrot [105] proposed fractal geometry and noticed its existence in the natural world. He first proposed a method to estimate fractal dimension (FD) while measuring the lengths of irregular coastlines. Peleg *et al.* [128] adopted Mandelbrot's idea and extended it to surface area calculation. In this extension, the image can be viewed as a hilly terrain surface whose height from the normal ground is proportional to the image gray value. Pentland [129] suggested a method to estimate FD using Fourier power spectrum.

Mandelbrot stated that one criteria of a surface being fractal is its self-similarity. A bounded set A in Euclidean n-space is said to be self-similar when it is the union of N_r distinct (non-overlapping) copies of itself each of which is similar to A scaled down by a ratio r. Fractal dimension D of A can be derived as

the ratio of $\log(N_r)$ and $\log(1/r)$. Fractal dimension is the principal feature used in fractal based texture analysis approaches.

Rigaut [139], Chaudhuri *et al.* [23] and Keller *et al.* [86] used fractal based features for texture segmentation. Ohanian and Dubes [121] have studied the performance of various texture features including fractal dimension. They compared performance of four fractal features along with other features for texture classification. Mandelbrot [106] and Peitgen [126] generated synthetic textures using fractal geometry.

Markov random field models

Markov random field(MRF) is popular for modeling images. They are able to capture the local(spatial) contextual information in an image. These models assume that the intensities at each pixel in an image depends on the intensities of only the neighboring pixels [38].

Chellappa [24] as well as Chatterjee and Chellappa [19] used features based on MRF model for texture segmentation.

Yokoyama and Haralick [172], Cross and Jain [38], Chellappa *et al.* [25] and Chellappa [24] used Markov Random Field models for texture synthesis.

Gabor filters

The Fourier transform is an analysis of the global frequency content in the signal. For texture analysis tasks, such as texture segmentation, the frequency content in small regions around each pixel is required. One way to localize the estimation of frequencies is to use a window Fourier transform. This is done by introducing spatial dependency into the Fourier analysis. When the spatial dependency function is Gaussian, the transform becomes a Gabor transform [49]. Gabor filters are complex sinusodial gratings modulated by 2–D Gaussian functions. A canonical Gabor filter in the spatial domain is given by

$$h(x,y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\}\cos(2\pi u_0 x + \phi)$$

where u_0 and ϕ are the frequency and phase of the sinusodial plane wave along the x-axis, respectively, and σ_x and σ_y are the space constants of the Gaussian envelope along the x- and y-axis, respectively.

Bovik *et al.* [16], Jain and Farrokhnia [72], and Farrokhnia [47] used Gabor filter in texture segmentation. Bovik *et al.* demodulated the filtered images to compute the feature images, while Jain and Farrokhnia [72] used a non-linear sigmoidal transformation to the filtered image to obtain the corresponding feature image.

Gray level run length

A gray level run is a maximal collinear connected set of pixels all having the same gray-tone. Properties of a run can be the run length, gray level, and angular orientation of the run. Galloway [50] used four directions : $0°, 45°, 90°$ and $135°$, and for each of these directions the joint probability of gray tone of the run and the run length were computed. The joint probability matrix is denoted by $p(i,j)$, where an element of $p(i,j)$ is the number of times there is a run of length j and having gray tone i. Useful statistics of the $p(i,j)$ are 'short run emphasis', 'long run emphasis', 'gray level *nonuniformity*', 'run length nonuniformity', and 'fraction of image in run'. Galloway used these features along with one feature derived from co-occurrence matrix for texture classification.

1.3.2 Structural Approaches

Pure structural models of texture are based on the view that textures are made up of primitives which appear in near regular repetitive spatial arrangements. To describe the texture, we must describe the primitives and the placement rules [142].

The choice of primitive and the probability of chosen primitive being placed at a particular location, can be a strong or weak function of location or of the primitives in the location. Textures here are classified as weak textures or strong textures. Weak textures are those which have weak spatial-interaction between primitives.

Primitives

A primitive is a connected set of resolution cells characterized by a list of attributes. The simplest primitive is a pixel with its gray tone attribute. Features, which will lead to identification of the primitives are either edge-based or region-based.

Vilnrotter *et al.* [162] used edge-based feature in order to recover a symbolic structure description of an ordered texture. They proposed the use of an edge repetition array.

In region based approaches, Tomita *et al.* [158] assumed the texture elements to be regions of homogeneous gray level. Simple region growing techniques were used to isolate these homogeneous regions. A simpler method is used by Lee and Wee [97], where the original textured image is thresholded to yield a binary image. Voorhees and Poggio [163] regarded blobs as dual of edges, *i.e.*, as areas in the image which do not correspond to edges. They presented a technique to estimate the amount of noise in the image, which allows automatic setting of the threshold to remove connections between blobs due to noise.

Spatial Relationship

There are several techniques to find spatial relationships among the primitives. Lu and Fu [102] adopted a syntactic approach to analyze the structured textures. The primitive element is not defined explicitly, but is actually a single pixel. Structure is extracted by determining whether a given texture belongs to the language generated by some *apriori* grammar. This grammar then defines the given texture.

Davis *et al.* [40] presented a method which involves a heuristic procedure

to measure repetitiveness in the histogram of the distance between a point and its k-nearest neighbors. Lee and Wee [97] used a variant of the nearest neighbor histogram.

Edge per unit area

Rosenfeld and Troy [143] proposed that the edge density, *i.e.*, the number of edge element per unit area can be used as a textural feature. The primitive here is the pixel and its property is the magnitude of its gradient. The gradient can be calculated by any one of the gradient neighborhood operators.

Local extrema

The use of the number of extrema per unit area for a texture measure has been suggested by Rosenfeld and Troy [143]. They defined extrema in one-dimension only along a horizontal scan. Mitchel *et al.* [114] used this method on smoothed version of the texture in order to eliminate extrema due to noise. Kundu and Chaudhuri [93] used a fuzzy version of extrema based method for texture classification.

Generalized co-occurrence matrix

Davis *et al.* [40] defined the concept of generalized co-occurrence matrices (GCM) which is essentially co-occurrence between relative extrema. Because of their invariance under any monotonic gray scale transformation, relative extrema primitives are likely to be very important.

Dyer *et al.* [46] and Tamura *et al.* [156] used features from GCM for texture classification, while Terzopolous and Zucker [157] used GCM for texture segmentation.

1.4 Scope and layout of the thesis

In this thesis we have initially considered feature detection and segmentation problems of multi-textured images. Fractal dimension related features are chosen for the purpose. At first, we presented a method of estimating Fractal Dimension which is simple, fast, and inexpensive. We also explored the idea of using multifractal dimension to textural image analysis. These features, when subject to an efficient clustering technique, results in an impressive texture segmentation.

Next, we described a procedure for automatic taxonomical classification of texture. We noticed that fractal based features are not enough for such purpose and other features such as texture directionality should be measured. We proposed a technique to find directionality that is based on evidence accumulation as in Hough Transform. Another problem addressed here is to estimate the minimum area of a texture image that should be examined to get a clear notion of the texture.

Finally, we focus on analysis of reservoir rock using textural clues. Sandstone and limestone are the two types of rocks examined here. Porosity and permeability are the important parameters of a sandstone reservoir rock, whereas identification of the bioclast is important in detecting limestone reservoir rock. The sandstone sample images are segmented using texture segmentation technique to estimate the porosity. Then using grain size analysis we estimate the permeability of the rock. In limestone samples the bioclast and non-bioclast areas are segmented using texture segmentation technique and then using directionality description, the presence of a particular type of bioclast is identified.

The chapterwise breakup of the thesis is as follows.

Chapter 2: In this chapter, at first, we made an extensive survey on the methods for estimation of fractal dimension. We found that some of these methods are accurate and cover the full dynamic range of Fractal Dimensions, but are computationally expensive. Others are computationally attractive but do not cover the

full dynamic range. We proposed a method which is computationally more attractive and covers the full dynamic range of Fractal Dimensions as well. We also compared the proposed method with the other methods. Estimation of generalized dimension of texture images is also considered in this chapter.

Chapter 3: The problem of feature detection and texture segmentation is addressed in this chapter. For this purpose we proposed six fractal based features which are intended to capture roughness, coarseness and directionality of the textures. For segmentation we considered K-means clustering approach. A new iterative method has been proposed to compute initial seed points. These seed points lead to a fast convergence in clustering stage. Later, we compared the performance of our textural features with the other popular textural features. We also made a comparative study of our clustering based approach with other classification techniques such as k-NN, minimum-distance classifier.

Chapter 4: In this chapter, at first, we proposed a direction measure for detection and gradation of oriented textures. With the help of this directional measure and fractal features we have done taxonomical classification of textures into different groups namely structural, directional and random. In this chapter an algorithm is also presented for finding the minimum size of a texture field that gives the perception of the particular texture.

Chapter 5: Analysis and gradation of reservoir rocks are dealt in this chapter. Both the sandstone and limestone samples (microphotographs) were considered for this purpose. For sandstone we estimated porosity and permeability from the given microphotographs. For finding porosity we used texture segmentation technique. Permeability has estimated indirectly from grain size analysis. Analysis of limestone samples are done in two stages. In first stage we segmented the given images to separate bioclast from the background. Foraminifera is an important bioclast to study further. We use the directionality cues to identify them.

Chapter 2

Fractal and multifractal dimensions

2.1 Introduction

Simple objects can be described by the ideal shape primitives, such as cubes, cones or cylinders. But most of the natural objects are so complex and erratic that they cannot be described in terms of simple primitives. On the other hand, the concept of self-similarity seems to play an important role in the description of nature. The complex and erratic shape description in terms of self-similarity was introduced by Mandelbrot [105], who proposed the fractal geometry of nature.

The concept of fractal dimension (FD) can be useful in the measurement, analysis and classification of shape and texture. Pentland [129,130] noticed that the fractal model of imaged three dimensional (3D) surfaces can be used to obtain shape information and to distinguish between smooth and rough textured regions. Rigaut [139] used the concept for image segmentation. Some of the other applications involve sedimentology and particle morphology [123,85], image data compression [10,11,71], speech wave graphs [132] and computer graphics [175].

There exist several approaches to estimate the FD of an image. For example, Peleg [128] used the ϵ -blanket method which is 2D generalization of the original approach suggested by Mandelbrot [105]. Pentland [129] considered the image intensity surface as fractal Brownian function (fBf) and estimated FD from Fourier power spectrum of fBf. Gangepain and Roques-Carmes [51] as well as Keller *et al.* [86] used variations of box-counting approach to estimate FD.

Fractal dimension is used to characterize systems with self-similarity of simple and homogeneous fractals. FD is not enough for characterization of sets having non-isotropic and inhomogeneous scaling properties. For such a characterization one has to generalize the analysis using the concept of multifractals which implies a continuous spectrum of exponents for the characterization of the system. In this generalization, an inhomogeneous fractal set is considered to be interwoven with infinitely many sub-fractal sets of different dimensions.

Blacher *et al.* [14] used multifractals to characterize the morphology of polymer alloys and granular discontinuous thin film. Kanmani *et al.* [80] used multifractal formalism in describing the inhomogeneity of stress corrosion crack patterns.

This chapter deals with the methods of estimating fractal dimension and generalized dimension of gray tone images. The generalized dimension is one of the ways of representing multifractals. The fractal dimension and its estimation approaches are reviewed in Section 2.2. It is useful to compare the approaches and suggest, if possible, an improved one that is computationally attractive and yeilds accurate result. A differential box counting approach is proposed in Section 2.3 for this purpose. This approach is compared with existing ones in terms of computer complexity and accuracy. Experimental results on various texture images are presented. Section 2.4 deals with the method of estimating generalized dimension of gray tone image.

2.2 Fractal Dimension and estimation approaches

2.2.1 Basic definitions

A set is called a fractal set if its Hausdorff-Besicovitch dimension is strictly greater than its topological dimension. Mandelbrot [105] coined the term fractal from the Latin word *fractus*, which means irregular segments.

The Hausdorff–Besicovitch dimension of a bounded set A in $\mathbb{R}^n$ is a real number used to characterize the geometrical complexity of A.

Consider the metric space $(\mathbb{R}^n, d)$, where n is a positive integer and d denotes the Euclidean metric defined on $\mathbb{R}^n$. Let $A \subset \mathbb{R}^n$ be bounded. The diameter of A is defined as

$$diam(A) = \sup\{d(x,y) : x, y \in A\}$$

Let $0 < \varepsilon < \infty$; $0 \leq p < \infty$. Let $\mathcal{A}$ denote the set of sequences of subsets $\{A_i \subset A\}$, such that $A = \bigcup_{i=1}^{\infty} A_i$. Then we define the *Hausdorff p-dimensional outer measure* of A as

$$\mu_{p,\varepsilon}(A) = \inf\left\{\sum_{i=1}^{\infty} [diam(A_i)]^p : \{A_i\} \in A, \text{ and } diam(A_i) < \varepsilon \text{ for } i = 1,2,3...\right\}$$

Note that $\mu_{p,\varepsilon}$ is non-decreasing as ε decreases.
Let $\mu_p(A) = \lim_{\varepsilon \downarrow 0} \mu_{p,\varepsilon}(A)$
The Hausdorff–Besicovitch dimension (D_H) is defined as

$$D_H = \sup\{p \in \mathbb{R} : p > 0, \mu_p(A) = \infty\}$$

where it is assumed that $\sup \phi = 0$

The Hausdorff–Besicovitch dimension is often called the fractal dimension (FD). Mandelbrot [105] defined FD in the following way.

Let $A \subset I\!R^n$. For each $\epsilon > 0$ let $\mathcal{N}(A, \epsilon)$ denote the smallest number of closed balls of radius ϵ needed to cover A. If for a constant C

$$\mathcal{N}(A, \epsilon) = C\epsilon^{-D} \tag{2.1}$$

exists, then D is called the *fractal dimension* of A.

2.2.2 Estimation Approaches

Mandelbrot described an approach to calculate FD while estimating the length of a coastline. Consider all points with distances to the coastline of no more than ϵ. These points form a strip of width 2ϵ, and the suggested length $L(\epsilon)$ of the coastline is the area of the strip divided by 2ϵ. As ϵ decreases $L(\epsilon)$ increases. Mandelbrot studied that for many coastlines the following formula holds good.

$$L(\epsilon) \;=\; F\epsilon^{1-D} \tag{2.2}$$

where F and D are constants for a specific coastline. He called D the FD of the coastline. D can be derived from least square linear fit of log–log plot of $L(\epsilon)$ and ϵ. If m is the slope of the fitted line then the FD of curve(coastline) will be $1 - m$. Note that m is always negative.

Peleg *et al.* [128] adopted Mandelbrot's idea and extended it to surface area calculation. In this extension, the image can be viewed as a hilly terrain surface whose height from the normal ground is proportional to the image gray value. Then all points at distance ϵ from the surface on both sides create a blanket of thickness 2ϵ. The estimated surface area is the volume of the blanket divided by 2ϵ. For different ϵ, the blanket area can be iteratively estimated as follows. The covering blanket is defined by its upper surface u_ϵ and the lower surface b_ϵ. Given the gray level function $g(i, j)$, initially, $u_o(i, j) = b_o(i, j) = g(i, j)$. For $\epsilon = 1, 2, 3...$, the blanket surfaces are defined as follows.

$$\begin{aligned} u_\epsilon(i,j) &= \max\{u_{\epsilon-1}(i,j) + 1, \max_{d(i,j,m,n)\leq 1} u_{\epsilon-1}(m,n)\} \\ b_\epsilon(i,j) &= \min\{b_{\epsilon-1}(i,j) - 1, \min_{d(i,j,m,n)\leq 1} b_{\epsilon-1}(m,n)\} \end{aligned}$$

where $d(i,j,m,n)$ is the distance between pixels (i,j) and (m,n).
Volume of the blanket is given by

$$v_\epsilon = \sum_{i,j}(u_\epsilon(i,j) - b_\epsilon(i,j))$$

while the surface area is measured as

$$A(\epsilon) = \frac{(v_\epsilon - v_{\epsilon-1})}{2}$$

The area of fractal surface behaves according to the expression

$$A(\epsilon) = F\epsilon^{2-D} \tag{2.3}$$

Fractal dimension can be derived from least square linear fit of log–log plot of A(ϵ) and ϵ, with the help of equation (2.3).

The recent results in the field of fractal model and the fractal geometry point out that most of the natural objects are not ideal but semi-fractal. As a result often two linear regions appear in the plot as two fractional elements [37]. The first element characterizes the first edge or surface effects of the analyzed image and is called *textural fractal dimension.* The second element describes the object at a higher resolution level and is called *structural fractal dimension.* These two quantities can be defined and calculated in the same way, applying the same algorithm but for different image resolution.

Pentland [129] suggested a method of estimating FD by using Fourier power spectrum of image intensity surface. It can be shown that Fourier power spectrum $P(f)$ of fractal Brownian function(f) is proportional to f^{-2h-1}, where $h = 2 - D$, and D is the FD. From least square fit of log-log of $P(f)$ and f, one can estimate FD of an image intensity surface, provided image intensity surface can be modeled as a fractal Brownian function.

Mandelbrot stated that one criteria of a surface being fractal is its self-similarity. Self-similarity can be explained as follows. Consider a bounded set A in Euclidean n-space. The set is said to be self-similar when A is the union of N_r

distinct (non-overlapping) copies of itself each of which is similar to A scaled down by a ratio r. Fractal dimension D of A can be derived from the relation [105].

$$1 = N_r r^D \text{ or } D = \frac{\log(N_r)}{\log(1/r)} \tag{2.4}$$

However, natural scenes practically do not exhibit deterministic self-similarity. Instead, they exhibit some statistical self-similarity. Thus, if a scene is scaled down by a ratio r in all n dimensions, then it becomes statistically identical to original one, so that equation (2.4) is satisfied.

It is difficult to compute D using (2.4) directly. An approximate method used by Gangepain and Roques-Carmes [51], called the *reticular cell counting* approach is as follows. Consider the 3-D space where two co-ordinates (x, y) represent 2-D position and the third(z) co-ordinate represents the image intensity. For a given scale L, partition the 3-D space into boxes of sides $L \times L \times L'$, where L can be multiple of sidelength of a pixel in (x, y) and L' can be multiple of gray level unit in z-direction. If G is total gray levels and $M \times M$ is the size of image then $L' = \lfloor L \times G/M \rfloor$. Let for $L = 1$ the box be called *space-intensity cell* or *spicel*. Then for $L = 3$, the box contains $3 \times 3 \times 3 = 27$ spicels. Suppose we can cover the 3-D space by a 3-D box of size L_{max}. Then $L = r \times L_{max}$. Changing parameter from r to L we have, from equation (2.4)

$$N_L = \frac{1}{r^D} = \left[\frac{L_{max}}{L}\right]^D$$

i.e.

$$N_L \propto L^{-D} \tag{2.5}$$

If a box contains at least one sample of gray level intensity surface, we can call it a countable box. For a fixed L, let the total number of countable boxes are N_L. Several values of the L are chosen and least square linear fit of log N_L versus log L gives the value of $-D$. But when the actual FD of an image is very high, points on image intensity surface become widely spaced in the z-direction, effectively lowering the estimated FD. From Fig. 2.5 it is seen that FD estimated by

this method saturates at about 2.5. However, this method is faster than Pentland's [129] method since no Fourier transform computation is required.

Keller *et al* [86] proposed a modification of a method due to Voss [164]. Let $P(m, L)$ be the probability that there are m intensity points within a box of size L centered about an arbitrary point of image intensity surface. For any value of L we have

$$\sum_{i=1}^{N} P(m, L) = 1$$

where N is the number of possible points in the box of size L. In fact, N equals the number of spicels the box contains. Suppose that the image is of size $M \times M$. If one overlays the image intensity surface with boxes of side L, then the number of boxes needed to cover the whole image is

$$N_L = M^2 \sum_{i=1}^{N} (1/m) P(m, L)$$

Since M^2 is constant for an image, let it be dropped from the expression, *i.e.*

$$N_L = \sum_{i=1}^{N} (1/m) P(m, L) \tag{2.6}$$

Using the equations (2.5-2.6) we can estimate D. This method has the same limitation as in Gangepain and Roques-Carmes method. To avoid this Keller *et al.* [86] devised a new version of probability estimation. In this refinement, the fractal surface between the center point of a box and its neighbors are approximated by linear interpolation. The newly interpolated surface is intersected with the box and number of points m in the box of side L is recorded. N_L is calculated using equation (2.6). This method takes little more time but gives satisfactory results except for the image intensity surfaces whose FD are very high. Fig. 2.5 shows that FD estimated by this method saturates at 2.75.

2.3 Differential Box Counting Approach [144,145]

2.3.1 Basic approach

We have a basic equation of FD given by

$$D = \frac{\log(N_r)}{\log(1/r)}$$

In our proposed method, N_r is counted in a different manner than those in [51,86]. Consider that the image of size $M \times M$ pixels has been scaled down to a size $s \times s$ where $M/2 \geq s > 1$ and s is an integer. Then we have an estimate of $r = s/M$. Now, as in previous techniques, consider the image as a 3-D space with (x, y) denoting 2-D position and the third co-ordinate (z) denoting the gray-level. The (x, y) space is partitioned into grids of size $s \times s$. On each grid there is a column of boxes of size $s \times s \times s'$. If the total number of gray levels is G then $\lfloor G/s' \rfloor = \lfloor M/s \rfloor$. See for example Fig. 2.1, where $s = s' = 3$. Assign numbers $1, 2, \ldots\ldots$ to the boxes as shown. Let the minimum and maximum gray level of the image in $(i, j)th$ grid fall in box number k and l, respectively. In our approach,

$$n_r(i, j) = l - k + 1 \tag{2.7}$$

is the contribution of N_r in $(i, j)th$ grid. For example, in Fig. 2.1, $n_r(i, j) = 3 - 1 + 1$. (Though in this figure, for simplicity, smooth image surface is taken, digital image surface will show discrete steps). Because of the differential nature of computing n_r we call our method as *differential box counting* (DBC) approach. Taking contributions from all grids, we have

$$N_r = \sum_{i,j} n_r(i, j) \tag{2.8}$$

N_r is counted for different values of r (*i.e.* different values of s). Then using (2.4) we can estimate D, the fractal dimension, from the least square linear fit of $\log(N_r)$ against $\log(1/r)$.

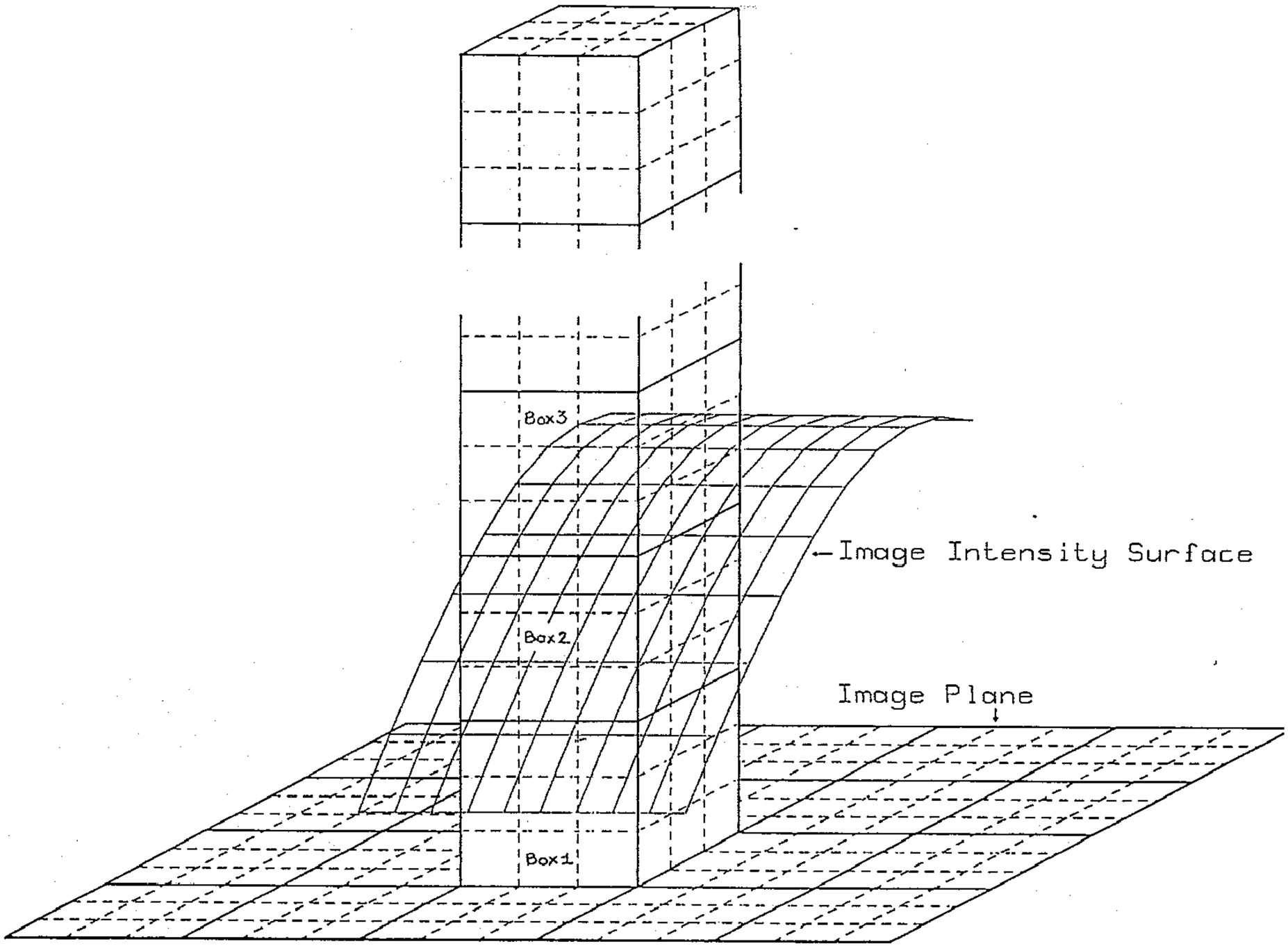


Figure 2.1: Determination of n_r by the proposed method

The reason of counting N_r in this manner is that it gives a better approximation to the boxes intersecting the image intensity surface, which is quantized in space and gray value. This is particularly so when there is sharp gray level variation in neighboring pixels in the image. In a sense, the DBC approach makes a digital approximation of thickness of the blanket that covers the image intensity surface at a particular resolution. Box counting in the other methods [51,86] does not cover the image surface so well and hence cannot capture the fractal dimension for rough textured surface. The other problems of box counting are discussed later in this section.

A typical plot of $\log(N_r)$ vs. $\log(1/r)$ of the image D33 (Brodatz [17]) is shown in Fig. 2.2. Let $y = mx + c$ be the fitted straight line, where y denotes $\log(N_r)$ and x denotes $\log(1/r)$. Then error of fit E can be expressed as the root mean square distance of the points from the fitted line.

$$E = \frac{\sqrt{\sum_{i=1}^{n} \frac{(mx_i+c-y_i)^2}{(1+m^2)}}}{n} \tag{2.9}$$

The error provides a measure of fit so that the lower the value of E, the better is the fit.

2.3.2 Modification of the approach

Although the DBC method produces very good estimate of FD, there exists a possibility of error due to quantization nature of the approach, especially when the image intensity surface is very smooth. As an example, consider an image (S01 in Fig. 2.3) whose gray level range is between 116 and 119. If we directly use DBC method we note that as long as $28 \geq s > 1$, all the gray values in this range will fall in a single box and the estimated FD will be 2.0, which in principle should be (slightly) more than 2.0. The same error will be introduced by other box-counting approaches as well.

We propose a modification that can take care of this problem. In the

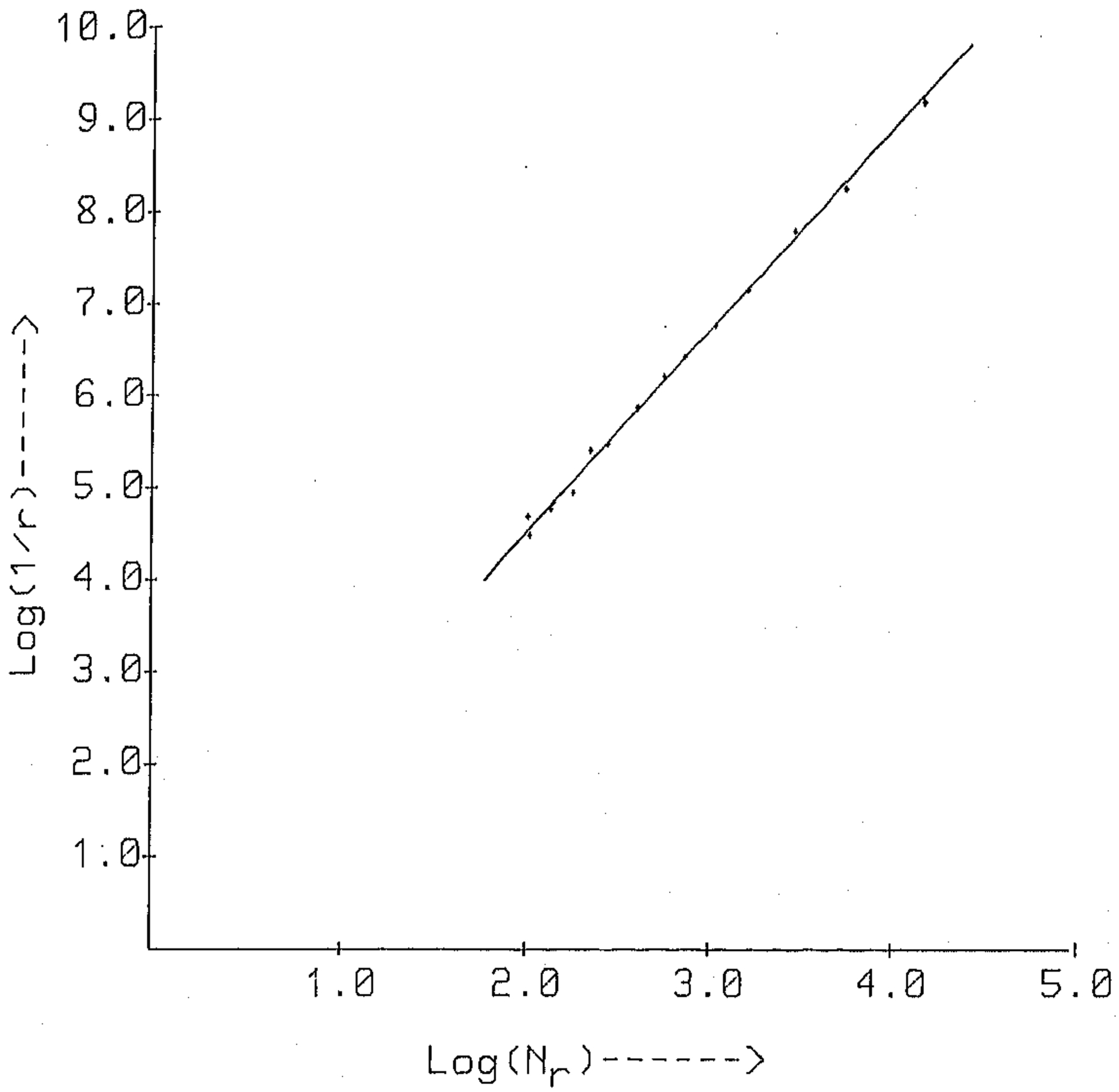


Figure 2.2: Plot of $\log N_r$ vs $\log(1/r)$ of texture image (D33 from Brodatz album)

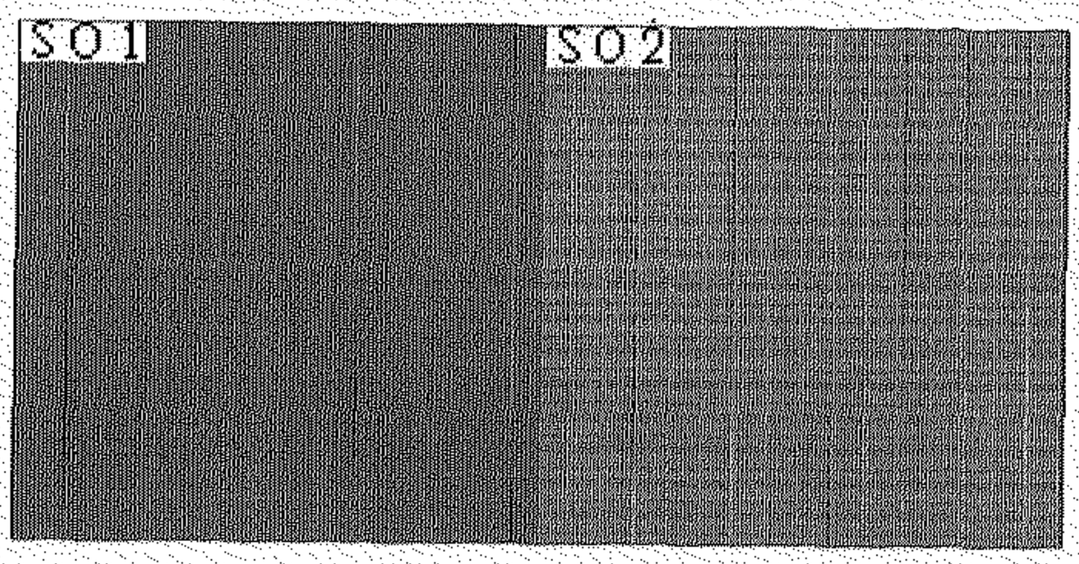


Figure 2.3: Smooth textures

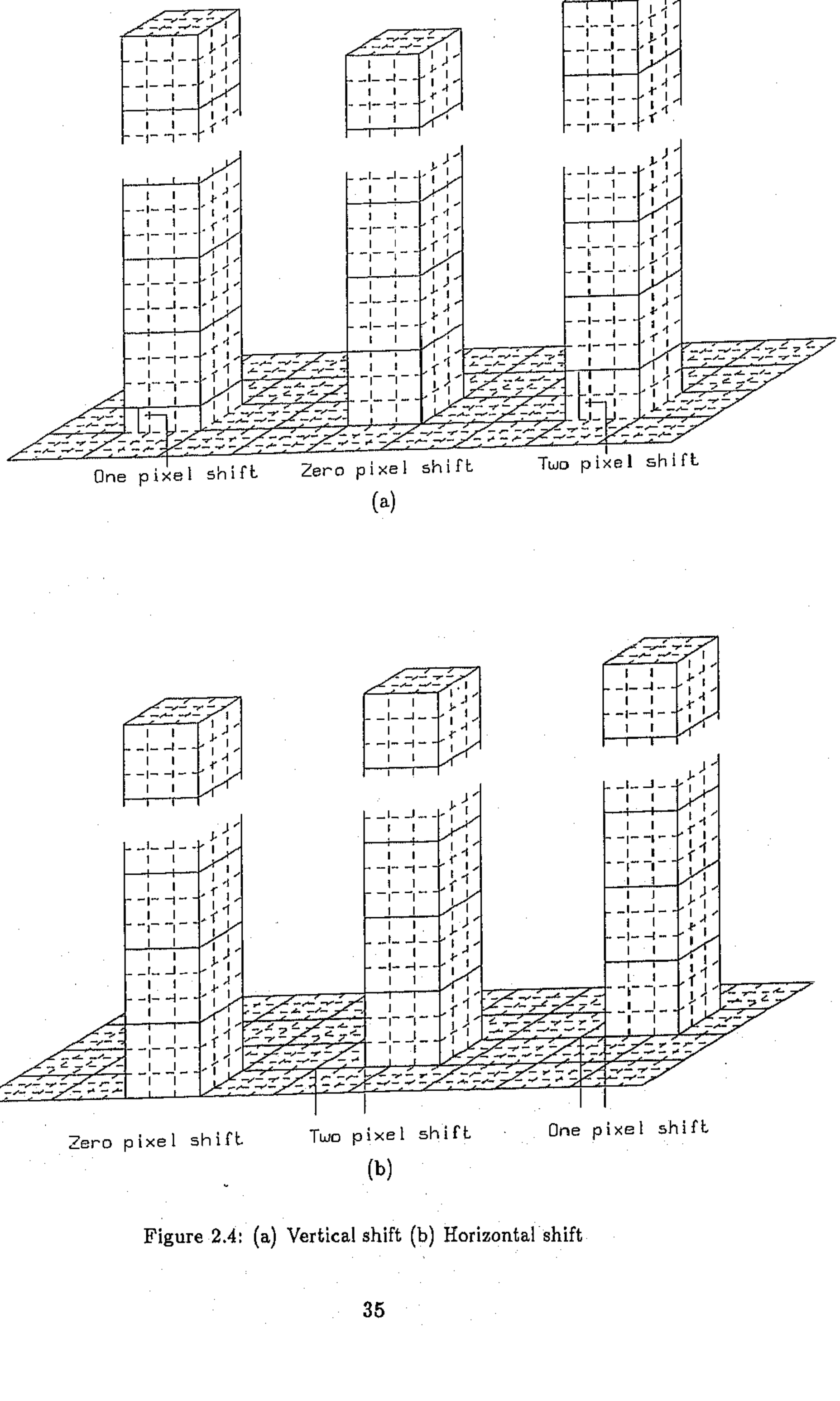


Figure 2.4: (a) Vertical shift (b) Horizontal shift

original DBC approach, the column of boxes are placed so that the height of the *j-th* box in the *i-th* column is the same as the height of the *j-th* box in any other column. In the modified approach, the columns are given random shift in vertical *i.e.*, z-direction so that the height of the *j-th* box in two columns may not be the same. The random shifts are given in the multiples of gray level height. See for example, Fig. 2.4(a). It may be understood that the small variation of gray value in an image can be captured by the modified approach. In fact, as seen from Table-2.1 the image S01 whose FD was estimated as 2.0, now has a modified FD=2.16 by random shifted box placement.

Table 2.1: Fractal dimension of textures with and without shift.

Texture	FD Estimated by using DBC		
Images	*Without shift*	*Predetermined shift*	*Random shift*
D06	2.32	2.36	2.35
D33	2.26	2.31	2.30
D34	2.20	2.26	2.25
D82	2.46	2.48	2.48
S01	2.00	2.19	2.16
S02	2.17	2.32	2.31

Like random shift in z-direction, there is a need for placement of the column of boxes by shifting them in the image plane, *i.e.*, in x or y direction. See, for example, Fig. 2.4(b). Such a necessity may be explained by the synthetic image S02 of Fig. 2.3. In this case, the spatial period of variation in gray level can beat with the spatial period of box columns so that any gray level difference does not fall in a single column of box, and hence are not counted. The beating effect can be overcome by x or/and y shift of columns of boxes. The shifts are in the multiples of pixel length/breadth. In Table-2.1, it is seen that the FD which without shift, is estimated to be 2.17 changes to 2.32 with shift.

The random shifting of column needs some additional computation in-

volving random number generation and differential box counting in each column. Instead of random shifts, we can make pre-determined shifts that reduce the computational overhead to a great extent. The difference in FD estimation by predetermined and random shifts are seen to be small in Table-2.2.

Table 2.2: Comparison of complexity of FD estimation by DBC method.

	Complexity of DBC method		
	Without shift	Predetermined shift	Random shift
Comparison	$2M^2$	$2M^2$	$2M^2$
Addition	$4/r^2$	$6/r^2$	$6/r^2$
Subtraction	$2/r^2$	$2/r^2$	$2/r^2$
Multiplication	$3/r^2$	$4/r^2$	$4/r^2$
Division	$3/r^2$	$4/r^2$	$4/r^2$
Random number generation			$1/r^2$

2.3.3 Results and Discussions

We choose five algorithms including DBC [144,145] for comparative study. The other four algorithms are due to Pentland [129], Peleg *et al.* [128], Gangepain and Roques-Carmes [51] and Keller *et al.* [86]. At first, the algorithms are tested on the synthetic images. It is expected that the FD will increase if the noise σ increases, and beginning at 2.0 the FD will asymptotically go towards a value of 3.0. A good method should reflect this desirable feature.

The results are plotted in Fig. 2.5. It is seen that the methods due to Pentland, Peleg *et al.*, and DBC approach give satisfactory result. On the other hand, the methods due to Gangepain and Roques-Carmes' and Keller *et al.* give satisfactory results upto certain level of roughness of the image intensity surface. After a certain value of σ, the slope of curve nearly goes to zero and hence these

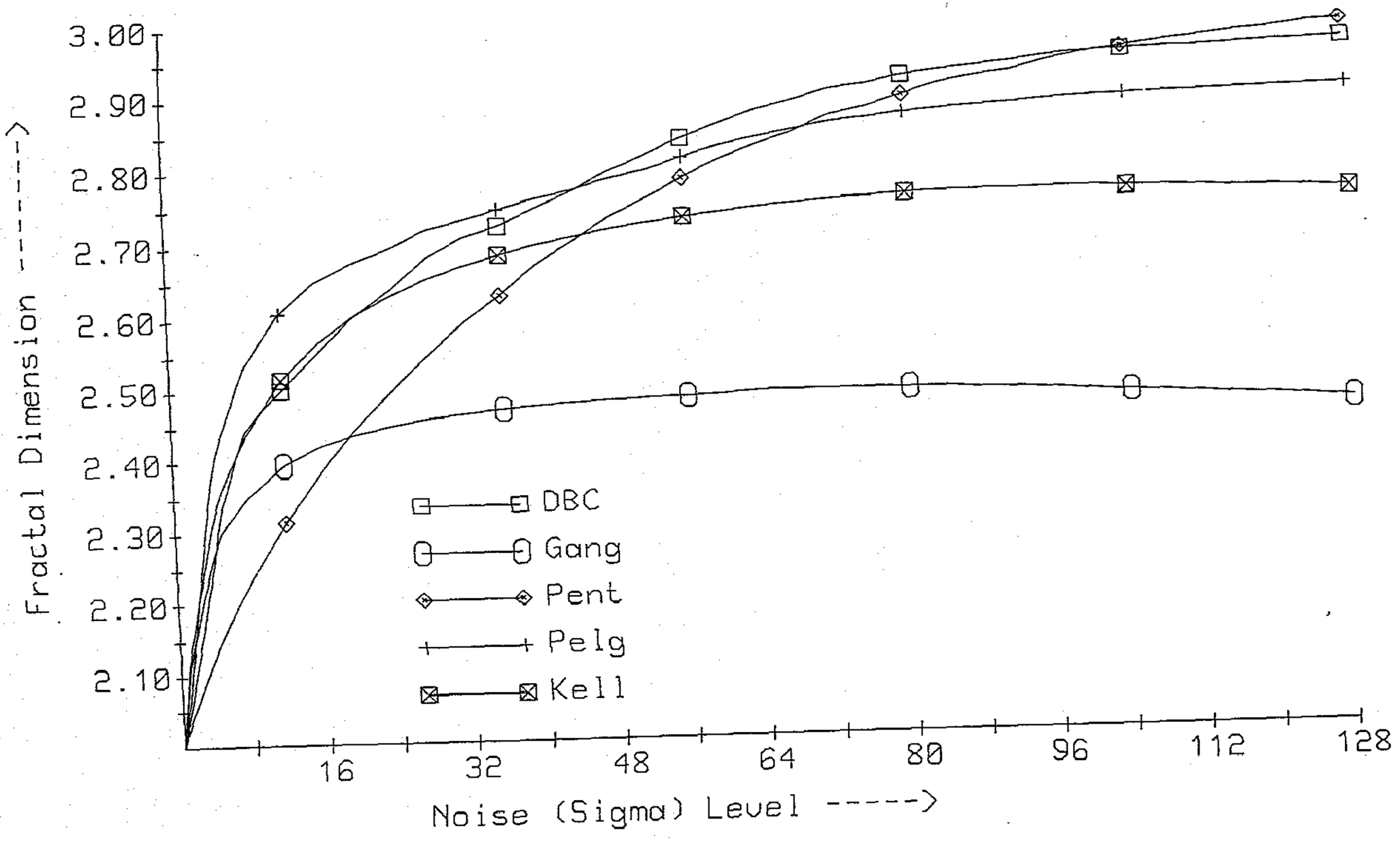

3.00
2.90
2.80
2.70
2.60
2.50
2.40
2.30
2.20
2.10
Fractal Dimension ------>
DBC
Gang
Pent
Pelg
Kell
16
32
48
64
80
96
112
128
Noise (Sigma) Level ----->
Figure 2.5: Fractal dimensions of synthetic images by different methods

Table 2.3: Comparison of complexity of three methods.

	DBC	Gang	Kell
Comparison	$2M^2$	$1/r^4$	r^2M^4
Addition	$4/r^2$	$M/r + 4/r^2$	$2M^2 + M^2L^2$
Subtraction	$2/r^2$		M^2
Multiplication	$3/r^2$	$2/r^2$	
Division	$3/r^2$	M^2	$r^3M^3 + M^2L^2$

methods do not cover full dynamic range of FD. In the method due to Keller *et al.* the acceptable range of FD is 2.0–2.75, while in Gangepain and Roques-Carmes' method the range is 2.0–2.5.

Next, we compare the computational complexity of different methods. Our method is readily comparable with the methods due to Gangepain and Roques-Carmes' [51] as well as Keller *et al.* [86], because in all these cases the computations are done taking boxes of different sizes. Relation between r in our method and L in the methods due to Gangepain and Roques-Carmes' as well as Keller *et al.* is given by $L = s = r \times M$, where $M \times M$ is the size of the image. The number of computations required in terms of M and r for each method is shown in Table-2.3(Computations required by DBC without shift is considered here). Computational superiority of our method is readily observed from this table.

In the approach of finding FD using the method due to Peleg *et al.* [128], the upper and lower blanket are computed in each iteration. To calculate each blanket, we used $10M^2$ comparisons, $6M^2$ additions, $6M^2$ subtractions, one multiplication and one division. Initially, we needed $4M^2$ subtraction to calculate zeroth blanket, before the start of the iteration.

Computational complexity of Pentland's [129] method is very high. It needs Fourier transform to find Fourier power spectra. To calculate FFT of an $M \times M$ image by conventional algorithm, one needs $2M \times \log_2 M$ operations,

Table 2.4: Comparison of number of computations.

	DBC	Gang	Kell	Pelg	Pent
Comparison	464142	417040	2687181	2457600	45056
Addition	37524	445183	418935	1474560	933376
Subtraction	18762	0	144060	1540096	636928
Multiplication	28143	18762	0	15	825856
Division	28143	232071	261630	15	101888
Square root	0	0	0	0	16384
Sine/cos	0	0	0	0	3584

where each of such operations consists of $M + M \log_2 M$ additions, $M + M \log_2 M$ subtractions, $4M$ multiplications, 3 divisions, one *sine* and one *cosine* operation. As other overheads one needs $4M^2$ comparisons and M^2 square roots.

Computation required for regression is not considered in any of the methods because regression is common to all. A comparative study on the actual number of operations required for real texture image (D03 of Brodatz [17]) is shown in Table-2.4, where 15 iterations are taken in methods other than Pentland's. Note that iterations are not necessary in Pentland's method. Again one can see that DBC method is computationally cheaper than the other methods.

Next, we have taken a set of 12 texture images from Brodatz [17] album to compare the FD obtained by different methods. The images are shown in Fig. 1.1, while results are presented in Table-2.5. It can be seen that the methods due to Pentland [129], Peleg *et al.* [128] and DBC give identical results. Also it may be noted that the FD due to DBC method is intermediate between that due to Pentland and Peleg *et al.* for all texture images. In Fig. 2.5 also, FD due to DBC method is intermediate between that due to Pentland and Peleg *et al.* in the range 2.23–2.58 which is roughly the range of FD for these texture images.

Table 2.5: Fractal dimension of natural textures using different methods (Image numbers correspond to Brodatz's album).

Texture	DBC		Pent		Pelg		Gang		Kell	
Images	FD	E	FD	E	FD	E	FD	E	FD	E
D03	2.60	0.032	2.54	0.004	2.69	0.011	2.40	0.074	2.63	0.036
D04	2.66	0.026	2.55	0.012	2.72	0.008	2.45	0.072	2.68	0.030
D05	2.56	0.032	2.48	0.003	2.63	0.012	2.39	0.057	2.62	0.025
D09	2.59	0.028	2.49	0.003	2.65	0.009	2.43	0.066	2.65	0.026
D24	2.45	0.022	2.36	0.002	2.59	0.007	2.39	0.048	2.57	0.014
D28	2.55	0.033	2.48	0.007	2.61	0.012	2.41	0.066	2.62	0.031
D33	2.23	0.007	2.21	0.002	2.34	0.003	2.26	0.024	2.36	0.008
D54	2.39	0.023	2.31	0.011	2.53	0.008	2.35	0.044	2.51	0.015
D55	2.48	0.031	2.37	0.006	2.60	0.010	2.39	0.057	2.59	0.023
D68	2.52	0.024	2.44	0.008	2.63	0.007	2.40	0.054	2.60	0.019
D84	2.60	0.029	2.47	0.001	2.68	0.009	2.43	0.067	2.65	0.028
D92	2.50	0.023	2.38	0.007	2.59	0.007	2.41	0.052	2.59	0.018

The problem of FD estimation is yet to be solved completely. We synthesized some Fractal surfaces by midpoint displacement algorithm [126]. The algorithm, when supplied with a value of FD, would return a fractal surface. If it is agreed that this surface is a true fractal with supplied dimension then an estimation method should find nearly the same value for the FD. As seen from the Table 2.6, the results are not so satisfactory. The methods overestimate (except for Peleg *et al.*) for low values and underestimate for high values of FD. The discrepancy in estimation is very large at higher values of FD. This discrepancy may be partly attributed to the limited resolution and quantization effect of digital image. Also, the choice of box size (*i.e.*, the range of s to be considered for line fitting in the log-log plot) seems to be important in estimating FD correctly.

Table 2.6: Estimated dimension of fractal images using different techniques.

Synthetic Texture	Estimated FD			
FD	DBC	Gang	Kell	Pelg
2.1	2.12	2.12	2.17	2.06
2.2	2.27	2.23	2.29	2.25
2.3	2.30	2.28	2.38	2.35
2.4	2.37	2.32	2.43	2.41
2.5	2.45	2.37	2.51	2.46
2.6	2.52	2.39	2.55	2.50
2.7	2.55	2.40	2.57	2.54
2.8	2.57	2.41	2.59	2.58
2.9	2.61	2.42	2.61	2.61

Gang : method due to Gangepain and Roques-Carmes [51], Kell: method due to Keller *et al.* [86], Pent : method due to Pentland [129], Pelg : method due to Peleg *et al.* [128]

However, from Table 2.6 it is seen that the estimated dimension is monotonic with the supplied dimension. This property suggests that FD can be used as a feature in classification and segmentation problem. Moreover, our estimation methods provides a good measure of surface roughness and hence can be applied for texture segmentation problems.

The error of fit of each method is computed using the eq (2.9). The comparative study of error of fit shows that error in DBC method is less than that due to Gangepain and Roques-Carmes [51] as well as Keller *et al.* [86], but somewhat more than that due to Pentland and Peleg *et al.*. However, since the results are throughout consistent, the error has negligible effect on the computed FD, which has been checked on 20 other texture images.

2.4 Multifractals and generalized dimension [146]

The properties of self-similarity of a set of points can be characterized by fractal dimension. Note that only for simple cases can this be considered as an exhaustive characterization. For non-isotropic or inhomogeneous scaling properties, the sets require a more general treatment and one has to generalize the analysis using the concept of multifractal which implies a continuous spectrum of exponents for the system characterization. During past few years it has been demonstrated that the multifractals are necessary to characterize the properties of various physical phenomena like turbulence, strange attractors in general, fractal growth models, multiplicative processes, and others.

There exists two approaches to compute multifractal [98]. The first one is based on defining a measure in terms of Hölder exponent. In the other approach some sort of 'moments' or generalized dimensions are estimated. These two descriptions are dual in nature.

2.4.1 Multifractal characterization using Hölder exponent

Let f be a summable function from $[a, b]$ to $\mathbb{R}$. Let l be a natural number. Consider normalization of the interval so that $(b - a) = 1$ and assume that we divide $[a, b]$ into ℓ intervals of equal length $1/\ell$. Let μ be any measure defined from f on each interval k_i. For example, we may have

$$\mu(k_i) = \int_{k_i} f(x)dx$$

The measure μ is called multifractal iff:

(i) For sufficiently large ℓ, i.e. for $1/\ell \to 0$

$$\forall k_i, \exists \alpha_{k_i} \in \mathbb{R} \mid \mu(k_i) \sim \left(\frac{1}{\ell}\right)^{\alpha_{k_i}}$$

Here, α is called Hölder exponent.

(ii) Let $E_{\alpha 0} = \{k_i \mid \alpha_0 - d\alpha < \alpha_{k_i} < \alpha_0 + d\alpha\}$
for a certain (small) $d\alpha$
Let

$$N_\alpha = card(E_\alpha)$$

where $card(x)$ denotes cardinality of x. Then

$$N_\alpha \sim d\alpha \left(\frac{1}{\ell}\right)^{-g(\alpha)}$$

where g is a function of α. Note that ℓ is equivalent to $1/r$, where r is defined in Section 2.3.

The above expression signifies that when we group all 'points' with the same α, we get a set E_α, whose fractal dimension is given by $g(\alpha)$.

The $(\alpha, g(\alpha))$ description is obtained by splitting the interval $[a, b]$ into subsets where all the points have the same Hölder exponent and then characterizing each subset by its fractal dimension. In this approach both local and global informations are contained via α and g, respectively.

2.4.2 Generalized Dimension

In computing the box counting dimension, one counts the non-empty boxes without any regard to the number of points in a box, *i.e.*, no weighting is done to the count according to the number of points in the box. This approach describes the geometry of the structure but ignores the underlying measure *i.e.* the density of the point distribution over it. A measure defined over a set describes the varying density of a positive scalar quantity like mass.

The generalized dimension introduced by Grassberger [55] takes into account of the number of points in a box. Let B_i denote the i-th box. Then $u(B_i)$ is the measure defined for the i-th box and is equal to the number of points within the box. Also, $u(A)$ is the total measure of the set A and is equal to the total number of points in the set A. Then $P_i = u(B_i)/u(A)$ is the normalized measure

of the ith box. The generalized dimension is given as

$$D_q = \frac{1}{q-1} \lim_{r \to 0} \frac{\log \sum_i^{N_r} P_i^q}{\log r} \tag{2.10}$$

where $-\infty \leq q \leq \infty$ and N_r denotes the non-empty boxes.

For a homogeneous fractal, with all the P_i being equal, one obtains the generalized dimension D_q, which does not vary with q. But in case of an inhomogeneous fractal, D_q usually decreases with increasing q. Note that D_0 computed using eq 2.10 is equivalent to the FD of the set A.

Coleman and Pietrinero [33] defined a partition function as

$$\chi(q,r) = \sum_i P_i^q \quad \propto \quad r^{\tau(q)}$$

The role of this function is to evaluate all the possible singularities of the distribution by studying all the q moments of the coarse grained distribution for different sized coarse graining. The exponent $\tau(q)$ defines the small r behavior of $\chi(q,r)$. Even a regular distribution can be characterized in this way. Their non-fractal properties would arise from a particular behavior of $\tau(q)$. The generalized dimension D_q can be defined as

$$(q-1)D_q \equiv \tau(q) = \lim_{r \to 0} \frac{\log \chi(q,r)}{\log r} \tag{2.11}$$

Applying L'hospital's rule in eq (2.10) we get

$$D_1 = \lim_{r \to 0} \frac{\sum_i P_i \log P_i}{\log r}$$

D_1 is called the information dimension. Note that the numerator term on the expression of D_1 is identical to classical information measure if P_i denotes probability. D_2 is called the correlation dimension which is a measure of correlation between pair of points inside a box. For $q = 3, 4 \ldots$ one can define a set of generalized dimensions D_3, D_4 associated with higher order correlations between triples, quadruples, of points in each box [14]. It is clear that the description given by (q, D_q) is global since we sum the contribution over the whole interval.

However, as mentioned above, the descriptions $(\alpha, g(\alpha))$ and $(q, \tau(q))$ are dual. For example, the $(\alpha, g(\alpha))$ description can be found from generalized dimension as

$$\alpha = \frac{d\tau}{dq}$$

$$g(\alpha) = q\alpha - \tau(q)$$

2.4.3 Box Counting Approach

As described in Section 2.2, the image is viewed as a 3-D map. Where the z-direction denotes the gray value. Now, the image intensity surface is approximated from the given image by connecting the neighboring points by straight lines. For a given box size s the image is partitioned into boxes of size $s \times s \times s'$. Let B_{ijk} denote the (i, j, k)-th box. Let $u(B_{ijk})$ be the number of points counted at B_{ijk}. The total number of points of the image intensity surface is given by

$$u(A) = \sum_{\forall i,j,k} u(B_{ijk})$$

$$\text{then} \quad P_{ijk} = u(B_{ijk})/u(A)$$

Let $\chi(q, r) = \sum_{ijk} P_{ijk}^q$. Then the 3–D generalization of the eq (2.10) can be written as

$$D_q = \frac{1}{q-1} \lim_{r \to 0} \frac{\log \chi(q, r)}{\log r} \quad (q \neq 1) \tag{2.12}$$

$$D_1 = \lim_{r \to 0} \frac{\sum_i P_i \log P_i}{\log r}$$

where $r = s/M$ as given in Section 2.3.

For a given value of q, $\chi(q, r)$ is computed for different values of r and D_q is estimated from the plot of $\log \chi(q, r)$ against $\log r$.

The image intensity surface approximation and counting of $u(B_{ijk})$ are the most important steps of this method. The approximation is done at first for all columns of the image where the neighboring gray value points (in 3-D space)are

joined by straight lines. Then the same process is repeated for all rows. These lines intercept the 3-D image cells. For counting, at first the 3-D space is partitioned into boxes of size $s \times s \times s'$. The number of image cells intercepted by the lines in the box B_{ijk} gives the count $u(B_{ijk})$. Note that for a given value of s the maximum count of $u(B_{ijk})$ is $s \times s \times s'$.

2.4.4 Results and Discussion

Let the image function be $f(x, y)$. Consider the plot $x, y, f(x, y)$ in a 3-D space for all x and y. Now, if the generalized dimension of the above space is estimated without the space approximation described in Section 2.4.3, then for $q = 0$ the estimated value of D_0 will be equal to the FD estimated using Gangepain and Roques-Carmes [51] method. On the other hand, if the surface approximation is made, then the estimated D_0 is similar to the FD estimated by DBC method. However, because of surface approximation, the number of non-empty boxes are more than N_r and hence D_0 is higher than FD estimated by DBC method. See Fig. 2.6 in 2-D where in column no. 5, the highest and lowest gray values are in box no. 7 and 4, respectively. Hence for this column the count $n_r = 7 - 4 + 1 = 4$. On the other hand, the straight lines joining the gray values intercept 5 boxes.

However, to save computer memory the image intensity surface approximation of the image should be done locally. For example, if we want to calculate $u(B_{ijk})$ where the box size is s, then the following steps can be taken.

Step(1): join $f(i*s+l, j*s+m)$ and $f(i*s+l, j*s+m+1)$ for $0 < l < s, 1 < m < s$

Step(2): join $f(i*s+l, j*s+m)$ and $f(i*s+l+1, j*s+m)$ for $1 < l < s, 0 < m < s$

Step(3): count $u(B_{ijk})$ as the number of points in box (i, j, k).

Blacher *et al.* [14] studied multifractal behavior of granular evaporated thin gold films of different gold concentration. They made the following observations from the plot of D_q versus q for different gold concentration. For the smallest concentration, with non-connected morphology, the generalized dimen-

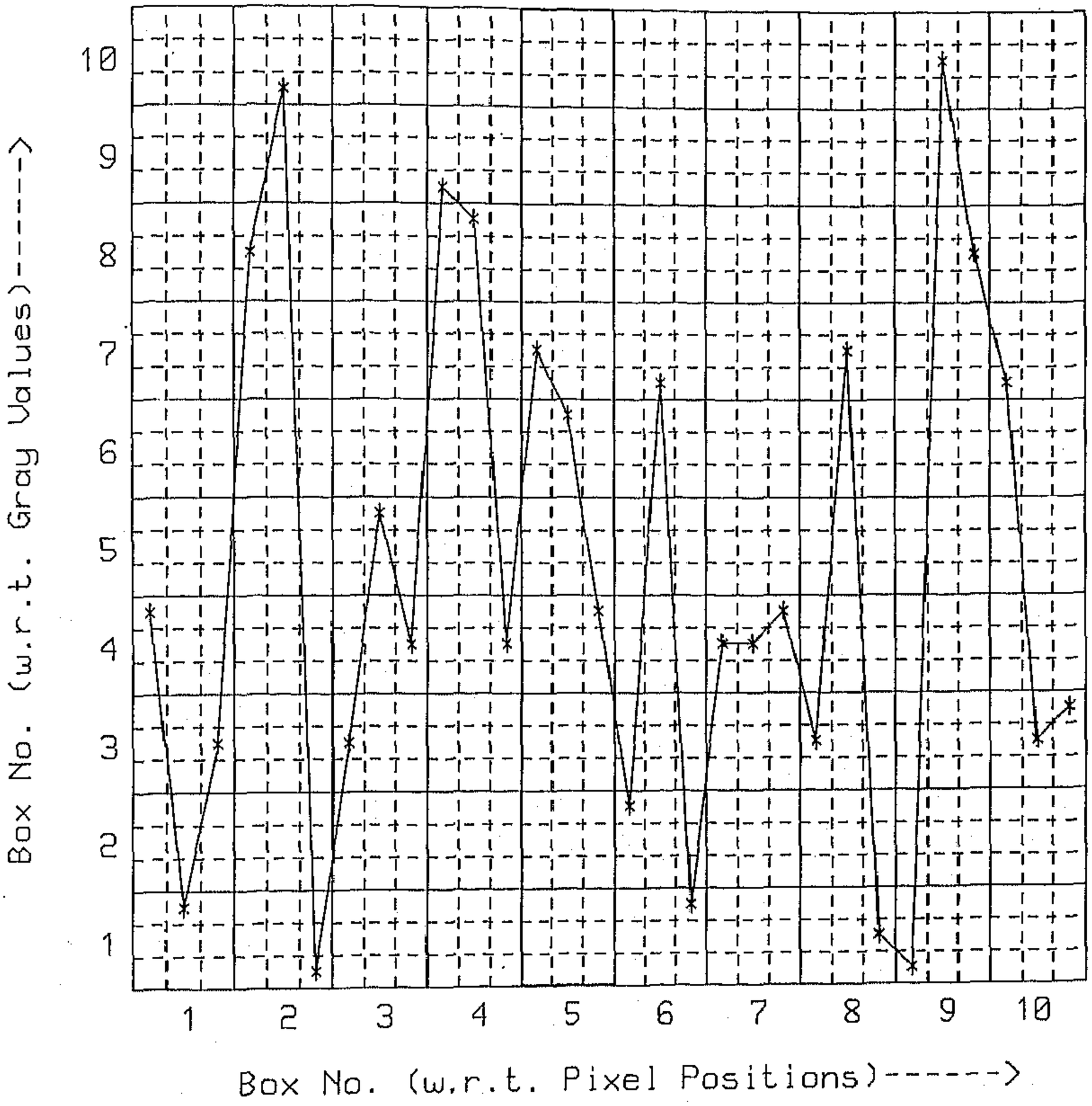


Figure 2.6: Surface approximation in one dimension

sion decreases linearly with q. For high concentration the curve tends toward a constant D_q indicating a more uniform feeling of the plane. Multifractality means non-uniform fractal and the greater the difference between D_0 and other dimension D_q, the more non-uniform is the structure. Kanmani *et al.* [80] studied the multifractal behavior of stress corrosion cracks. They observed that the plot of D_q versus q indicates the inhomogeneity of the cracks. Multifractal behavior of a textural image is somewhat more complicated than binary image. Textures can be described by its basic properties such as coarseness, uniformity, roughness and directionality. Some of these properties can be described by the generalized dimension of the texture. We stated earlier that FD reflects the roughness of a texture. Usually, the value of $D_0 - D_2$ reflects the coarseness of the grain of the texture. We have used feature related to D_2 for texture segmentation in Chapter 3. It has also been observed that as we increase the value of q, D_q saturates to a certain value for a given texture, indicating the limit of fractal non-uniformity of texture image. It is observed that for fine texture the curve of D_q versus q is more or less straight. Similar behavior is observed by Blacher *et al.* [14] on the image of gold film with low gold concentration. See, Table 2.7 and Fig. 2.7 for the results on Brodatz album textures shown in Fig. 1.1. We believe that the multifractal and generalized dimensions can also reveal other physical aspects of an image.

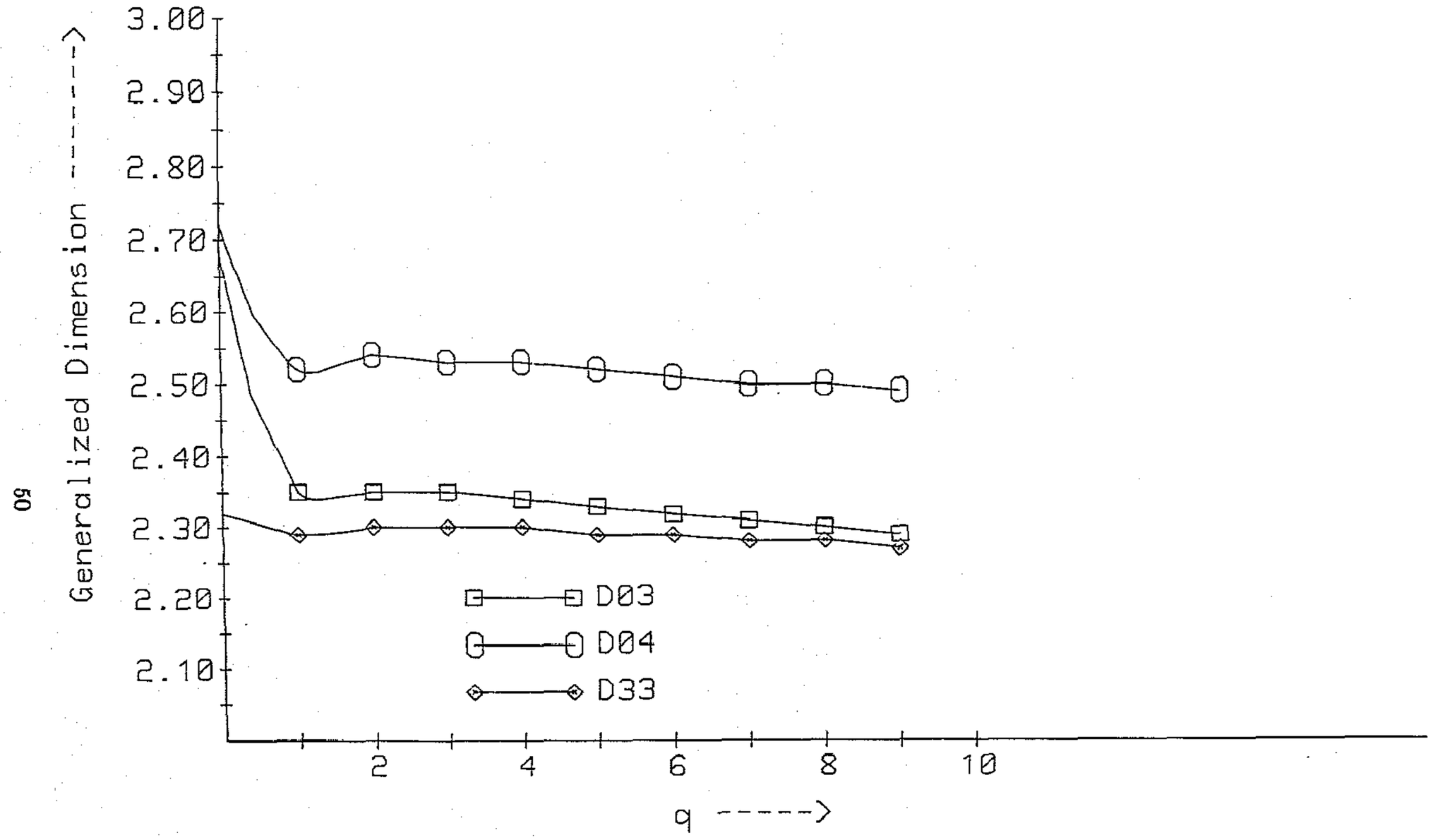


Figure 2.7: Generalized dimension for texture images

Table 2.7: Generalized dimensions of different texture images.

Image	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
D03	2.68	2.35	2.35	2.35	2.34	2.33	2.32	2.31	2.30	2.29
D04	2.72	2.52	2.54	2.53	2.53	2.52	2.51	2.50	2.50	2.49
D05	2.63	2.46	2.48	2.48	2.48	2.47	2.46	2.46	2.45	2.44
D09	2.66	2.44	2.47	2.47	2.47	2.47	2.46	2.46	2.45	2.45
D24	2.53	2.41	2.44	2.45	2.45	2.44	2.44	2.43	2.43	2.42
D28	2.63	2.33	2.37	2.38	2.39	2.39	2.39	2.38	2.38	2.38
D33	2.32	2.29	2.30	2.30	2.30	2.29	2.29	2.28	2.28	2.27
D54	2.47	2.31	2.33	2.34	2.34	2.34	2.33	2.33	2.33	2.32
D55	2.56	2.32	2.35	2.36	2.36	2.36	2.35	2.35	2.34	2.34
D68	2.59	2.44	2.47	2.48	2.48	2.48	2.47	2.47	2.47	2.46
D84	2.67	2.43	2.45	2.45	2.44	2.44	2.43	2.42	2.41	2.41
D92	2.57	2.45	2.48	2.49	2.49	2.48	2.48	2.47	2.47	2.46

Chapter 3

Feature Evaluation and Segmentation

3.1 Introduction

An important task in image analysis is to segment the given image into meaningful regions and label them. The presence of texture in a scene makes the task more difficult. This chapter deals with the segmentation of a scene containing regions of different textures.

There are two important aspects of a segmentation procedure namely, *feature selection* and *segmentation criteria selection*. The texture features used in our work are based on the fractal geometry of images. The choice was initially motivated by the observation that fractal dimension (FD) shows a strong correlation (> 0.9) with human judgment of surface roughness [129]. Peleg *et al.* [128] found 48 features from the original and smoothed versions of the image and used them for texture classification. Keller *et al.* [86] estimated FD using a box-counting approach and used it for texture segmentation.

In this chapter we show that accurate and computationally efficient texture

segmentation is possible with six FD-based features although according to some comparative study [44] FD alone can perform texture segmentation with limited success. For this purpose, the modified box-counting method (DBC) described in previous chapter is used to estimate the FD [144,145]. The FDs of the original as well as the high gray valued and low gray valued versions of the image are normalized in the range [0,1], and used as three features. The fourth and the fifth features are derived from the FD of horizontally and vertically smoothed versions of the image. The sixth feature, based on the concept of multifractal, is computed by a higher moment of box-counting approach [55,33,98,146]. These features also lie in the interval[0,1]. To the best of our knowledge, this is the first time where the multifractal concept has been used for texture segmentation. The features are described in Section 3.3, while other non-fractal features are briefly reviewed in Section 3.2. Some of these features are used for a comparative study with our method.

Next, the feature space representation of the scene is smoothed to get rid of spurious segmentation [74]. This aspect is described in Section 3.4. Texture segmentation is done through clustering in the feature space and mapping back the results in image space. It is assumed that the number of texture classes are known apriori. Other supervised classification methods, such as the k-nearest neighbor and minimum distance classifier are also considered. These approaches are treated in Section 3.5. Various mosaics generated from the texture images of the Brodatz album [17] were used to test the segmentation scheme. The experimental results are presented in Section 3.6.

3.2 Popular Textural Features

Perhaps the most widely used texture features are based on gray level co-occurrence matrix (GLCM). Let S be the set of all pairs of pixels in a given spatial relation. Then the co-occurrence $P(i,j)$ of gray levels i and j for an image

I can be defined as

$$P(i,j) = \mid \{((m,n),(k,l)) \in S \mid I(m,n) = i \text{ and } I(k,l) = j\} \mid$$

The co-occurrence matrix can be normalized by dividing each entry by the sum of all entries in the matrix. Let $p(i,j)$ be the normalized value at the (i,j)th position of the matrix.

Julesz [77] first used gray tone spatial dependence co-occurrence statistics in texture discrimination experiments. Haralick *et al.* [59] suggested a set of 14 textural features which can be extracted from each of the GLCM. The most commonly used ones are energy, entropy, contrast, correlation, and homogeneity. They are defined as

$$\text{Energy} \quad : \text{ASM} : \quad \sum_i \sum_j p^2(i,j)$$

$$\text{Entropy} \quad : \text{ENT} : \quad -\sum_i \sum_j p(i,j) \log_2 p(i,j)$$

$$\text{Contrast} \quad : \text{CON} : \quad \sum \sum (i-j)^2 p(i,j)$$

$$\text{Correlation} \quad : \text{COR} : \quad \sum_i^i \sum_j^j (ijp(i,j) - \mu_x \mu_y)/(\sigma_x \sigma_y)$$

$$\text{Homogeneity} \quad : \text{HOM} : \quad \sum_i \sum_j \frac{p(i,j)}{1+\mid i-j \mid}$$

where μ_x, μ_y, σ_x, and σ_y are means and standard deviations of row sums and column sums of $p(i,j)$, respectively.

Pavlidis and Chen [125], Pietikainen [133], Conners *et al.* [36] and He *et al.* [63] obtained satisfactory texture segmentation results using features based on co-occurrence matrix.

Laws [95] proposed an approach to measure texture energy in the spatial domain. In this approach, at first, the image is convolved with a set of filters called microtexture masks, and their outputs are called microtexture features. These masks are derived from simple vectors of length three namely, $L3 = [1,2,1], E3 = [-1,0,1]$, and $S3 = [-1,2,-1]$, which represent the 1–D operations of center-weighted local averaging, edge detecting, and spot detecting. If

these vectors are convolved among themselves or each other, five vectors of length 5 are obtained as follows

$$\begin{aligned} L5 &= [1,4,6,4,1] = L3 * L3 \\ S5 &= [-1,0,2,0,-1] = E3 * E3 = L3 * S3 \\ R5 &= [1,-4,6,-4,1] = S3 * S3 \\ E5 &= [-1,-2,0,2,1] = L3 * E3 \\ W5 &= [-1,2,0,-2,1] = E3 * S3 \end{aligned} \tag{3.1}$$

where $L5$ is again a local average, $S5$ and $E5$ are, respectively, spot and edge detectors and $R5$ and $W5$ can, respectively, be regarded as *ripple* and *wave* detectors. Sets of larger vectors can be defined by repeating this convolution process. Laws obtained 2–D masks by convolving row vectors with the column vectors of the same length. Mask obtained by $L3^t * S3$ (where t indicates transpose) is denoted as L3S3. The two dimensional convolution of the image $I(i,j)$ and mask $h(i,j)$ of size $(2a+1) \times (2a+1)$ is given by

$$\begin{aligned} x(i,j) &= (h * f)(i,j) \\ &= \sum_{k=-a}^{a} \sum_{l=-a}^{a} h(k,l).I(i+k,j+l) \end{aligned} \tag{3.2}$$

where $a = 1,2,3$.

Laws' microtexture masks are designed to act as matched filters for certain types of quasiperiodic variations commonly found in textured images. Typically, these masks are of maximum size 7×7. In most of the cases, the sum of the elements of the mask is zero. Most widely used convolution masks are $E5L5$, $R5R5$, and $L5S5$.

The macrostatistical features are obtained over large windows (e.g., 15×15 to 31×31). The most useful statistics are the sample variance or a sample mean deviation of the microtetxure features. The sample mean deviation within a $(2n+1) \times (2n+1)$ window at point (i,j) is given by

$$s(i,j) = \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} | x(k,l) - m(i,j) |$$

where the mean $m(i,j)$ is given by

$$m(i,j) = \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} x(k,l)$$

Hsiao and Sawchuk [66,67] used these features for texture segmentation. They achieved results better than those obtained by Laws' by little post processing of the feature image.

Texture features based on Gabor filters and Autoregressive(AR) models are also used for texture segmentation. Bovik *et al.* [16], Jain and Farrokhnia [72], and Farrokhnia [47] used Gabor filter in texture segmentation. Kashyap and Khotanzad [82], Khotanzad and Chen [89], Wan *et al.* [168], and Mao and Jain [108] used different AR models for feature extraction and later used them for texture segmentation.

Some experiments were done using fractal geometry for texture classification and segmentation, although they are not popular. Peleg *et al.* [128] used fractal dimension along with 48 fractal signatures for texture classification. Keller *et al.* [86] used fractal dimension and lacunarity measure for texture segmentation. Dubuisson and Dubes [45] used different fractal based features for texture segmentation.

3.3 Proposed Fractal Dimension Based Features [22,23]

As stated before, our choice of fractal based features is motivated by the fact that fractal dimension is a measure of surface roughness. However, Mandelbrot and Van Ness [107] and Voss [164] have pointed out that different textures may have the same FD. This may be due to the combined effect of coarseness and directionality (dominant orientation and the degree of anisotropy) in the texture images. We propose to use six features in order to discriminate these aspects.

These features are the FD of: (1) the original image, (2) the high gray-valued image, (3) the low gray-valued image (4) the horizontally smoothed image (5) the vertically smoothed image and (6) the multi-fractal dimension of the original image. For any feature f we ensure that $f \in [0,1]$.

Feature 1. FD of the original image I_1 is computed on overlapping windows of size $(2W+1) \times (2W+1)$. Thus, at point (i,j) the first feature value $F_1(i,j)$ is defined as

$$F_1(i,j) \;=\; FD\{\; I_1(i+l, j+k) \;;\; -W \leq l,k \leq W \;\} \tag{3.3}$$

where FD is the fractal dimension derived by the method described in Section 2.2. Since $2.0 \leq F_1(i,j) \leq 3.0$, we define the normalized feature as

$$f_1(i,j) \;=\; F_1(i,j) - 2 \;\text{ so that }\; 0 \leq f_1(i,j) \leq 1$$

Feature 2-3. Consider two modified images called high and low gray-valued images I_2 and I_3 defined respectively, as

$$I_2(i,j) \;=\; \begin{cases} I_1(i,j) - L_1 & \text{if } I_1(i,j) > L_1 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

$$I_3(i,j) \;=\; \begin{cases} 255 - L_2 & \text{if } I_1(i,j) > (255 - L_2) \\ I_1(i,j) & \text{otherwise} \end{cases} \tag{3.5}$$

where $L_1 \;=\; g_{min} + av/2 \;;\; L_2 \;=\; g_{max} - av/2$ while g_{max}, g_{min} and av denote the maximum, minimum and average gray value in I_1, respectively. If two images I_1 and J_1 have the same FD, their high gray-valued images I_2 and J_2 may not have identical roughness and their FDs would be different. Similar conclusion may be drawn for I_3 and J_3. The normalized features f_2 and f_3 are computed from I_2 and I_3 in the same way as the computation of f_1 from I_1.

Feature 4-5 It is to be noted that FD of an image is related to its roughness and hence it is reduced by gray value smoothing. Smoothing is normally done on a window that is running along a particular direction. For a highly oriented texture, the FD will be least affected if the texture is smoothed along the direction

of its dominant orientation. But if the direction of smoothing is perpendicular to the above, the FD will be substantially reduced. On the other hand, the texture having low degree of orientation will have identical effect on FD irrespective of the smoothing direction.

Any orientation can be decomposed into two components in horizontal and vertical direction. So, we have considered images smoothed in horizontal and vertical direction and computed their FD as the fourth and fifth feature, respectively. Feature 4 and 5 are also useful in discriminating two regions having the same texture where one is a rotated version of the other.

Horizontally and vertically smoothed version of the image are defined as

$$I_4(i,j) = \frac{1}{2w+1} \sum_{k=-w}^{w} I(i,j+k) \tag{3.6}$$

$$I_5(i,j) = \frac{1}{2w+1} \sum_{k=-w}^{w} I(i+k,j) \tag{3.7}$$

The FD of these images are obtained and the normalized features f_4 and f_5 corresponding to I_4 and I_5 are computed in a manner similar to that of f_1.

Feature 6. The basics of multifractals and generalized dimension are described in Section 2.4. For the sixth feature the generalized dimension of order two, *i.e.*, D(2) of I_1 is computed on overlapping windows of size $(2W+1) \times (2W+1)$. The partition function used for the purpose is

$$\chi(q,r) = \sum_{i,j} [n'_r(i,j)]^q \simeq r^{\tau(q)}$$

where r is described in Section 2.3 and $n'_r(i,j) = n_r(i,j)/N_r$, $n_r(i,j)$ and N_r are given by eq (2.7) and eq (2.8), respectively.

The sixth feature f_6 is

$$f_6 = TD - D(2) \tag{3.8}$$

where TD is the topological dimension, which is 2 in this case.

3.4 Feature domain smoothing

If the features described above are directly used for segmentation, considerable mis-classification can occur in the interior of a region and the border between two regions. A feature domain smoothing can reduce the mis-classification to a large extent. Smoothing by ordinary moving window averaging can result in considerable blurring at the border and hence an edge preserving smoothing technique [20,74,118] should be used.

The locally linear minimum mean square error (LLMMSE) estimator suggested by Kuan *et al.* [92] has the property that it smoothes out noise in flat regions and leaves the observation unchanged in the vicinity of the edges. The performance of the local filter strongly depends on how the local windows are chosen and how the local statistics are calculated.

A more sophisticated edge preserving smoothing algorithm was proposed by Nagao and Matsuyama [118]. Nine windows, each containing a few pixels, were used to estimate local statistics at each pixel. The most homogeneous neighborhood among the nine regions was then selected. The average gray level of the selected neighborhood is used as the estimate for the pixel under consideration. Noise in an image is removed by repeated use of this method, while the edge remains sharp. For complex-shaped regions, this method yields better result than quadrant window does, at the expense of higher computation cost.

Jiang and Sawchuk [74] suggested a quadrant method to estimate the local statistics. This method works very well for large and simple shaped regions. Here quadrant windows around each pixel are used to estimate local statistics and the averaged gray level of the most homogeneous neighborhood is used to replace the pixel under consideration. This method, called Edge Preserving Noise Smoothing Quadrant (EPNSQ) filtering approach, is used here for smoothing.

The method is explained by Fig. 3.1. Consider four quadrants within a $2W \times 2W$ neighborhood around the candidate at (i,j). The mean and variance of the m-th feature are computed on each window as follows

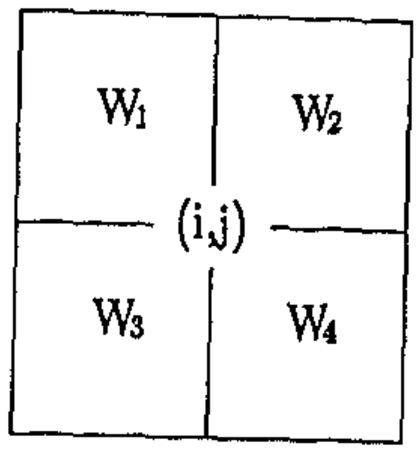


Figure 3.1: Schematic representation of the quadrants

$$A_{m,W_1}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} f_m(i-k,j-l), \tag{3.9}$$

$$A_{m,W_2}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} f_m(i-k,j+l), \tag{3.10}$$

$$A_{m,W_3}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} f_m(i+k,j-l), \tag{3.11}$$

$$A_{m,W_4}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} f_m(i+k,j+l), \tag{3.12}$$

$$V_{m,W_1}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} [f_m(i-k,j-l) - A_{m,W_1}(i,j)]^2, \tag{3.13}$$

$$V_{m,W_2}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} [f_m(i-k,j+l) - A_{m,W_2}(i,j)]^2, \tag{3.14}$$

$$V_{m,W_3}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} [f_m(i+k,j-l) - A_{m,W_3}(i,j)]^2, \tag{3.15}$$

$$V_{m,W_4}(i,j) = \frac{1}{w^2}\sum_{k=0}^{w-1}\sum_{l=0}^{w-1} [f_m(i+k,j+l) - A_{m,W_4}(i,j)]^2, \tag{3.16}$$

where $A_{m,W_n}(i,j)$ and $V_{m,W_n}(i,j)$; $n = 1,...,4$ represent the average and the variance computed on the quadrant W_n around (i,j), respectively.

If $V_{m,W_k}(i,j)$ is the minimum of $V_{m,W_n}(i,j)$, $n = 1,..,4$ then according to EPNSQ approach the feature value at (i,j) is replaced by $A_{m,W_k}(i,j)$.

3.5 Texture Segmentation

The general problem of image segmentation into homogeneous regions is one of the many intriguing topics in image processing. Regions resulting out of a good image segmentation technique should be uniform and homogeneous with respect to some characteristic such as gray level or texture; adjacent regions of segmentation should have significantly different values with respect to the characteristics on which they are uniform; region interiors should be simple and without small holes; and boundaries of each segment should be simple, not ragged, and must be spatially accurate [60].

There exist two major approaches to image segmentation : edge-based and region-based. In an edge-based segmentation approach the edges are detected by taking two adjacent windows and deciding whether the pixels in the windows belong to the same texture. If it is decided that the textures in the two windows are different, the point is marked as a texture boundary or edge pixel. In region-based methods, areas of the image with homogeneous properties are found, which in turn define the boundaries. The presence of texture causes difficulty for conventional edge- and region-based methods. Texture homogeneity is different from gray level homogeneity while boundary between two different textured regions is not detected by conventional edge detection methods. For both edge- and region-based texture segmentation the following ingredients are essential : a set of texture features having good discriminatory power; a segmentation algorithm having spatial constraints. The segmentation results not only depend on the qualities of feature extraction and segmentation algorithms, but these qualities interfere among themselves as well.

Here, our concern is on the region based segmentation algorithm, which may be supervised or unsupervised. In a supervised segmentation algorithm one or more representative pixels of each class are supplied. The number of classes are also known. In the case of unsupervised segmentation, hardly any apriori image knowledge or operation guidance is available, the problem is more difficult. For our purpose we used an unsupervised algorithm which is essentially the K-means

clustering algorithm with minor variation. The unsupervised and supervised techniques used here are described below.

3.5.1 Unsupervised segmentation

In the unsupervised situation we assume that only the number of texture classes K is known and there exist no training samples with known class label. The clustering algorithm used here is essentially a K-means algorithm. It is known that the results of K-means algorithm is dependent on the choice of seed points. A histogram based approach of seed point selection is described here. The histogram of each feature computed on the scene is obtained. For the i-th feature f_i, let H_i be the histogram. The dominant peaks of the histogram are found. Let N_i denote the number of dominant peaks in H_i and let the peaks occur at $f_{i1}, f_{i2}, ..., f_{iN_i}$. An average non-overlapping span(S_i) of the ith feature is calculated as $(f_{iN_i} - f_{i1})/(2N_i)$. The cut-off span ($S_c$) in the feature domain is defined as

$$S_c = \sqrt{\sum_{i=1}^{6} S_i^2}$$

The average radius R of a cluster in the spatial domain is calculated as

$$R = \sqrt{\frac{A}{\pi \times K}}$$

where A is the number of points of the feature image and K is the number of clusters.

The feature vector of the point at the location $(0, 0)$ in the feature image is chosen as the first initial seed point. All the remaining points are scanned one by one. The feature vector of a scanned point p is considered as a new seed point if

1. the Euclidean distance between the feature vector of the scanned point p and each of the other chosen seed points exceeds S_c.

and 2. p is spatially at a distance greater than R from the position of any other chosen points.

This process is continued until all the points are considered. In this way we obtain $K_0 > K$ seed points. Each seed point generates a separate cluster. These

K_0 clusters are then iteratively reduced to K clusters by redistributing the data of smallest cluster among the other clusters. Once this K initial clusters are obtained, the K-means algorithm [159] is run to get the final clustering. The K-means algorithm has the following steps.

*Step*1: Choose K initial cluster centers $z_1(1), z_2(1), \ldots, z_K(1)$. These are arbitrary and are usually the first K samples of the given sample set.

*Step*2: At the kth iterative step the datum x is assigned to a cluster, using the relation,

$$x \in S_j(k) \quad \text{if} \quad \|x - z_j(k)\| < \|x - z_i(k)\| \forall i = 1, 2, \ldots K, i \neq j$$

where $S_j(k)$ denotes the set of samples whose cluster center is $z_j(k)$.

*Step*3: From the result of step 2, compute the new cluster centers $z_j(k+1), j = 1, 2, \ldots, K$, such that the sum of the squared distances from all points in $S_j(k)$ to the new cluster center is minimized. In other words, the new cluster center $z_j(k+1)$ is given by

$$z_j(k+1) = \frac{1}{N_j} \sum_{x \in S_j(k)} x, j = 1, 2, \ldots, K$$

where N_j is the number of samples in $S_j(k)$.

*Step*4: If $z_j(k+1) = z_j(k)$ for $j = 1, 2, \ldots, K$ the algorithm has converged and the procedure is terminated. Otherwise, go to Step 2.

The results, when mapped from feature space to image space, yield the texture segmentation.

3.5.2 Supervised segmentation

The essential practical steps for supervised segmentation are:

- Choose a representative set of data for each class. The representative data are called prototype or training samples.

- Use the training samples to estimate the parameters of the particular classifier algorithm. The classifier is now trained for the particular problem.
- Using the trained classifier, label or classify every pixel in the image into one of (here textures) the classes.
- Connected set of pixels of single label denote a segmentation of the image. To obtain a connected component of reasonable size and shape, some post-processing may be necessary.

We have done the following post processing in our experiment. If a pixel P belonging to a texture class say X and seven or more of its immediate neighbors belong to another class Y then P is assigned to Y.

To compare the discriminatory power of different classification approaches, we considered two supervised techniques namely, minimum distance classifier and k-nearest neighbor classifier.

Minimum distance classifier

Consider M pattern classes and assume that these classes are represented by prototype patterns $z_1, z_2, \ldots, z_M$. The Euclidean distance between an arbitrary pattern vector x and the ith prototype is given by

$$D_i = \|x - z_i\| = \sqrt{(x - z_i)^t (x - z_i)} \tag{3.17}$$

A minimum distance classifier computes the distance from a pattern x of unknown classification to the prototype of each class, and assigns the pattern to class to which it is closest.

$$x \in \omega_i \quad \text{if} \quad D_i < D_j \quad \forall j \neq i$$

Instead of single prototype we may consider multiple prototypes for a given class, say for example, $z_{i_1}, z_{i_2}, \ldots, z_{iN_i}$ where N_i is the number of prototypes

in of class i. The distance between an arbitrary pattern vector x and class ω_i is given by

$$D_i \doteq \min_l \|x - z_{il}\| \quad l = 1, 2, \ldots, N_i \tag{3.18}$$

As before, the distances $D_i, i = 1, 2, \ldots, M$ are computed and the pattern is assigned to the class to which it is closest i.e.,

$$x \in \omega_i \quad \text{if} \quad D_i < D_j \quad \forall j \neq i$$

k-NN classifier

Although the ideas of small numbers of prototypes and familiar Euclidean distances are geometrically attractive, they are not limiting factors in the definition of the minimum-distance classification concept. In order to explore further the general properties of this scheme, let us consider a set of sample patterns of known classification $\{s_1, s_2, \ldots, s_N\}$, where we may assume that each pattern belongs to one of the classes $\omega_1, \omega_2, \ldots, \omega_M$. We may define the *nearest neighbor* (NN) classification rule which assigns a pattern x to the class of its nearest neighbor. We say that $s_i \in \{s_1, s_2, \ldots, s_N\}$ is the nearest neighbor to x if

$$D(s_i, x) = \min_l \{D(s_l, x)\}, \quad l = 1, 2, \ldots, N \tag{3.19}$$

where D is any distance measure definable over the pattern space.

This method is called 1-NN classification method. The similarity of 1-NN and minimum distance classifier with multiple prototype can be noted. In general, we may have k-NN classifier which consists of determining the k nearest neighbors to x, and using *majority* of equal classifications in this group as the classification of x.

3.6 Experimental results and discussion

To test the performance of the proposed technique, different texture images were chosen from the Brodatz album [17]. Mosaic of several textures may

exist in a natural scene. Our experience is that a region of reasonable size normally contains two to four different textures in its neighborhood, although the whole scene may contain more. We created different sets of mosaics with two or four textures. We describe the results on mosaics of four textures since our results on two-texture mosaics are consistently better than those of four-texture ones. To see the robustness of our algorithm, mosaics of 5, 6, 9, and 16 textures were also generated and tested.

In Fig. 3.2($a_1 - f_1$), six mosaics of four textures are shown. Each mosaic is of size 256 × 256 pixels so that each field texture is of size 128 × 128 pixels. The images are quantized in 256 gray levels. The FD of the images of I_1, I_2, I_3, I_4, and I_5 are computed on overlapping windows of size 17 × 17. On the otherhand, $f_6(i,j)$ is obtained using eq (3.8) on overlapping window of size 17 × 17 pixels.

The feature domain smoothing has been done on windows of size 7 × 7. Next, all of the three classification approaches are used to segment the textures. Of these, the results of the unsupervised approach on Fig. 3.2 ($a_1 - f_1$) are shown in Fig. 3.2 ($a_2 - f_2$). It is noted that the mis-classification occurs mainly at the texture border and on an average, the border error depth is about 6 pixels. The border error is partly at some regions where the two textures are visually indistinguishable. For example, see the enclosed region between D33 and D28 in Fig. 3.2(a). Another reason of mis-classification is the window size effect. The border mis-classification should not exceed half the window dimension, *i.e.*, $\lceil 17/2 \rceil = 9$ pixels away from the border, which is satisfied in our experiments. Interior regions, where significant mis-classification occurs, are visually similar in texture belonging to another class. For example, in Fig. 3.2(f_1), the encircled regions of d09 has texture similar to d04, while the encircled region of D28 has texture similar to D24. Similarly, in Fig. 3.2(e_1), the texture of encircled region of D09 is visually identical to the texture of D04. Note also the texture similarity between encircled region of D28 and D54 in the same mosaic.

We tested the Laws' microtextural features as well as the co-occurrence based features on the same texture mosaics. Among the Laws' features, windows of various sizes were considered to calculate the macrostatistical features. It was

noted that the best results were obtained for a window of size 31×31. To calculate the microtextural features we considered E5L5, R5R5, L5S5 and E5S5 described in [95,96] as microtextural masks. In the co-occurrence based technique we used ASM, ENT, CON, and COR described in [57,59] as the based features. Here also we considered windows of different sizes and the features calculated using windows of size 31×31 yielded the best results. The feature domain smoothing and segmentation technique are the same as our fractal based methods. These results are presented in Table 3.1. The results show that our results are more consistent than the other two methods. The window size was varied in both the other methods and the best results are presented.

The accuracy of classification due to different techniques are presented in Table 3.2. Here, the number of training samples used to compute the centroid for minimum distance classifier is 50. In the k-nearest neighbor technique, the chosen values of k are 1 and 5 while the number of training samples are 25 and 50 (all combinations are taken). From Table 3.2 it is seen that the methods yeild identical classification accuracy on the mosaics. Also, we tested our algorithm on the synthetic four-texture mosaic generated by Gaussian Markov random field model [25] for which a correct classification of 96.3% was achieved. See Fig. 3.3(a–b).

Next, to see the effect of increasing the number of textures, we considered mosaics of 5, 6, 9 and 16 textures. They are shown in Fig. 3.4($a-b$), Fig. 3.5($a-b$), Fig. 3.6($a-b$) and Fig. 3.7($a-b$), respectively. The results of segmentation appear to be quite satisfactory in these cases also. The classification accuracy for 5 and 6 texture mosaics are 95.82% and 94.41%, respectively. The classification accuracy with 9 texture mosaic is 93.7% and for 16 texture mosaic it is 91%.

Keller *et al.* [86] considered texture mosaics similar to those of ours. Their segmentation approach is also based on FD of original image and a 'lacunarity' measure. They have not presented any result of quantitative error analysis. We were inspired by the work in [86]. However, a comparative study [44] of different methods showed that FD alone perform poorly in texture segmentation. So we

Table 3.1: Classification accuracy for different sets of features.

Texture Mosaic	Fractal	Laws'	Co-occurrence
3.2(a)	94.93	92.50	92.27
3.2(b)	96.83	95.03	93.75
3.2(c)	97.00	94.53	82.18
3.2(d)	95.40	92.47	91.82
3.2(e)	91.11	82.77	81.10
3.2(f)	91.63	80.93	79.10
Average	94.49	89.98	86.70

Table 3.2: Classification accuracy for different techniques.

Texture Mosaic	Unsupervised	1NN		5NN		Min-dist
		A	B	A	B	
3.2(a)	94.93	87.00	87.00	85.81	86.90	85.28
3.2(b)	96.83	96.60	96.66	96.62	96.95	96.97
3.2(c)	97.00	96.89	96.67	97.05	96.68	97.10
3.2(d)	95.40	94.68	94.68	94.65	94.72	94.68
3.2(e)	91.11	92.55	92.55	91.60	92.26	91.40
3.2(f)	91.63	90.76	91.00	90.38	90.46	90.39
Average	94.49	93.08	93.09	92.70	93.00	92.62

A – taking 25 training samples, B – taking 50 training samples

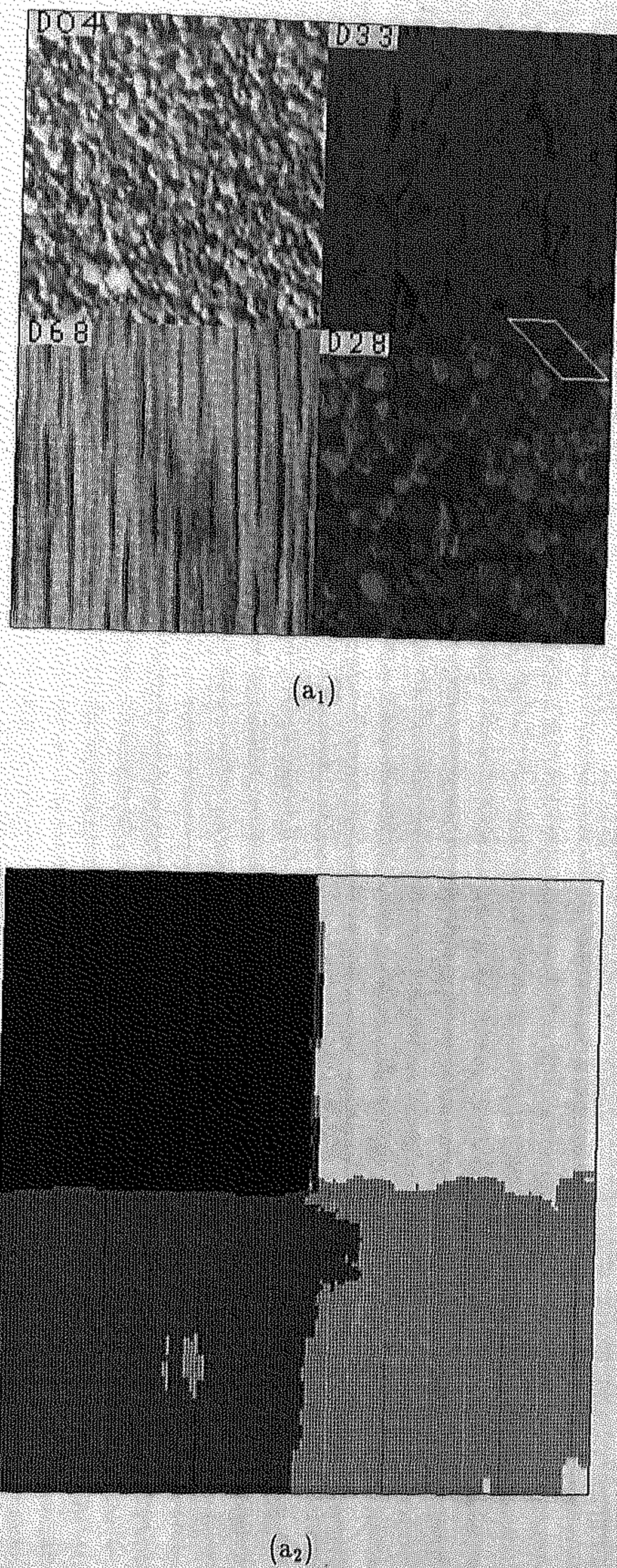
used several FD-based features, along with feature domain smoothing and clustering to obtain excellent segmentation.

Among other methods Hsiao and Sawchuk [66,67] used Law's [95] texture energy as the texture feature. In [66] they used Bayesian classifier for clustering and reported 95%, 93%, and 88% accuracy for the mosaics of 4, 5, and 8 textures, respectively. Using K-means algorithm for clustering [67], they obtained 95.7% and 92.5% correct classification for 4 and 5 texture mosaics, respectively.

Jain and Farrokhnia [72] and Farrokhnia [47] used the Gabor filterbanks for extracting the texture features. Their classifier was based on the feed-forward neural network model, among other techniques. For the synthetic four-texture mosaic generated by Gaussian Markov random field model [72,47], an accuracy of nearly 97% has been reported. For a mosaic of 5 textures using neural net 94% and 96% correct classification were achieved after 10 and 100 training cycles, respectively. Similarly, for a mosaic of 16 textures they obtained an accuracy of 87% and 92.5% using 10 and 100 training cycles, respectively.

To improve our results further, a post-segmentation processing may be employed. One approach is to use the relaxation technique, as presented in [127,58]. Our experience is that relaxation model of Peleg [127] improves the results at the interior of a region by nearly 1%, but the mis-classification at the border between textures are not reduced by this approach. A border curve smoothing criterion may be incorporated to improve the results at the border region. If apriori knowledge about the area of a texture is available, a small mis-classified area may be assigned to the class of its surroundings.

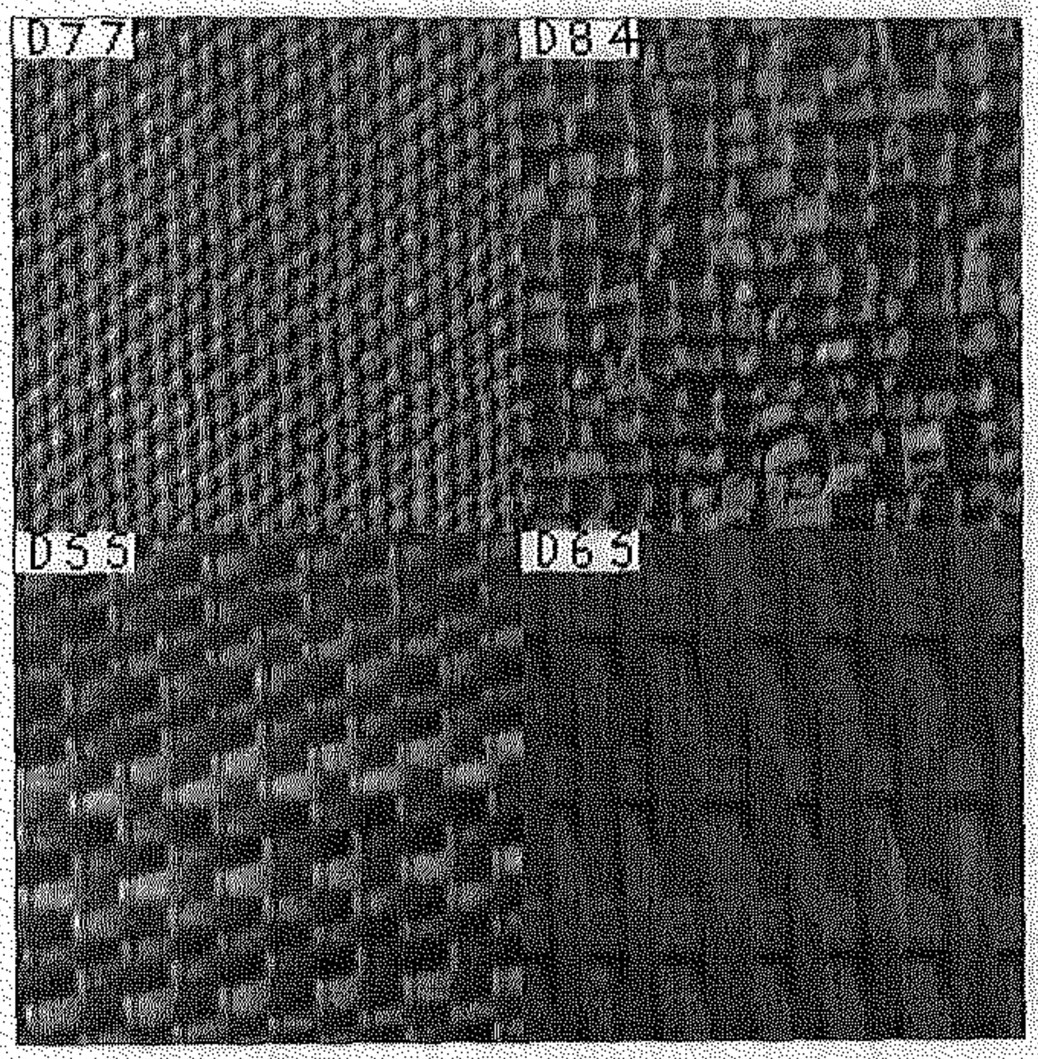
It is instructive to look for more powerful fractal features for segmentation. The concept of multi-fractal is another subject of further investigation. It may be possible that somewhere in the multi-fractal spectrum of FD, we can get the best set of features for texture segmentation.



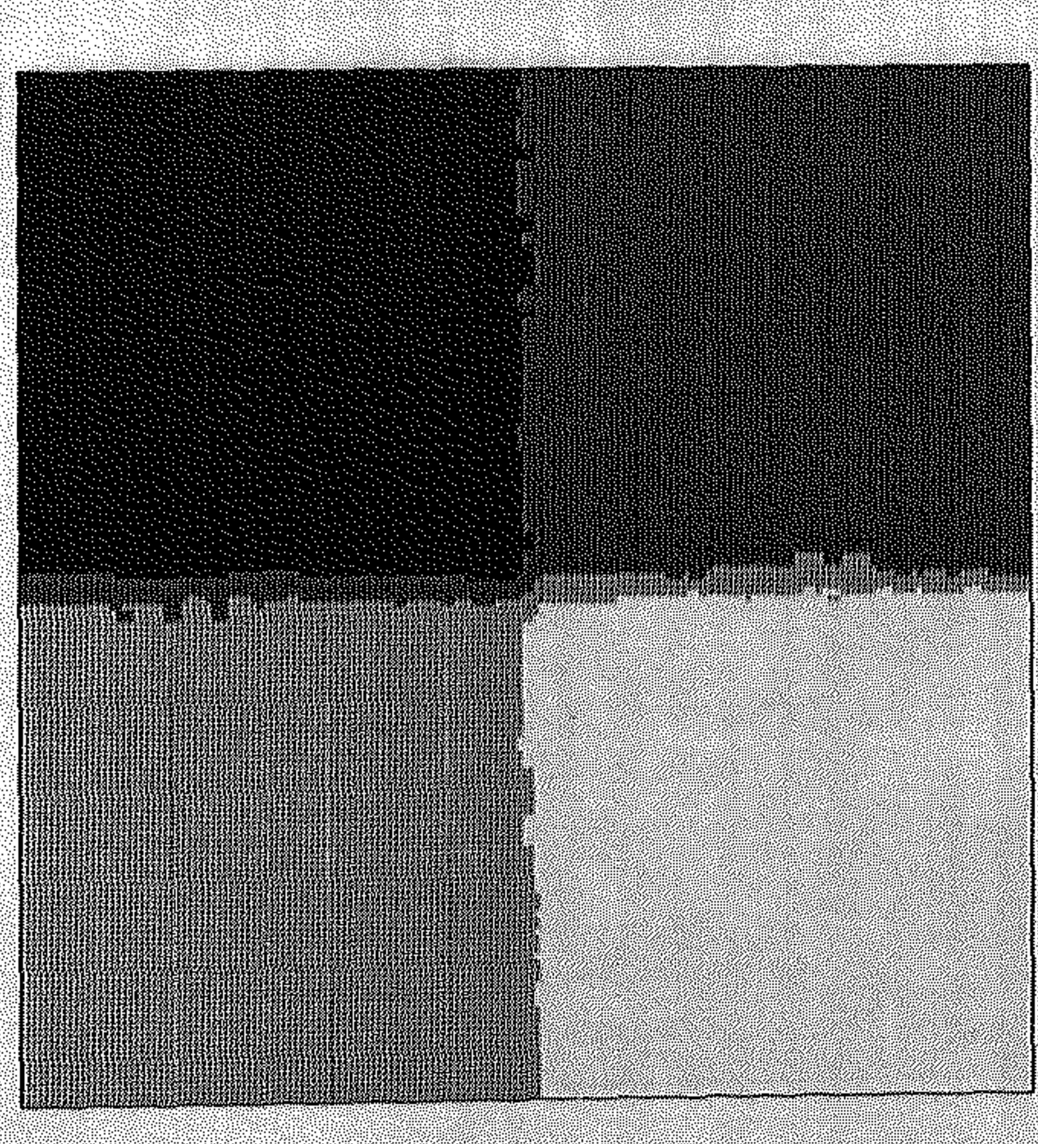
(a_1)

(a_2)

Figure 3.2a : (a_1) Four texture mosaic. (a_2) Segmentation mapping of (a_1).

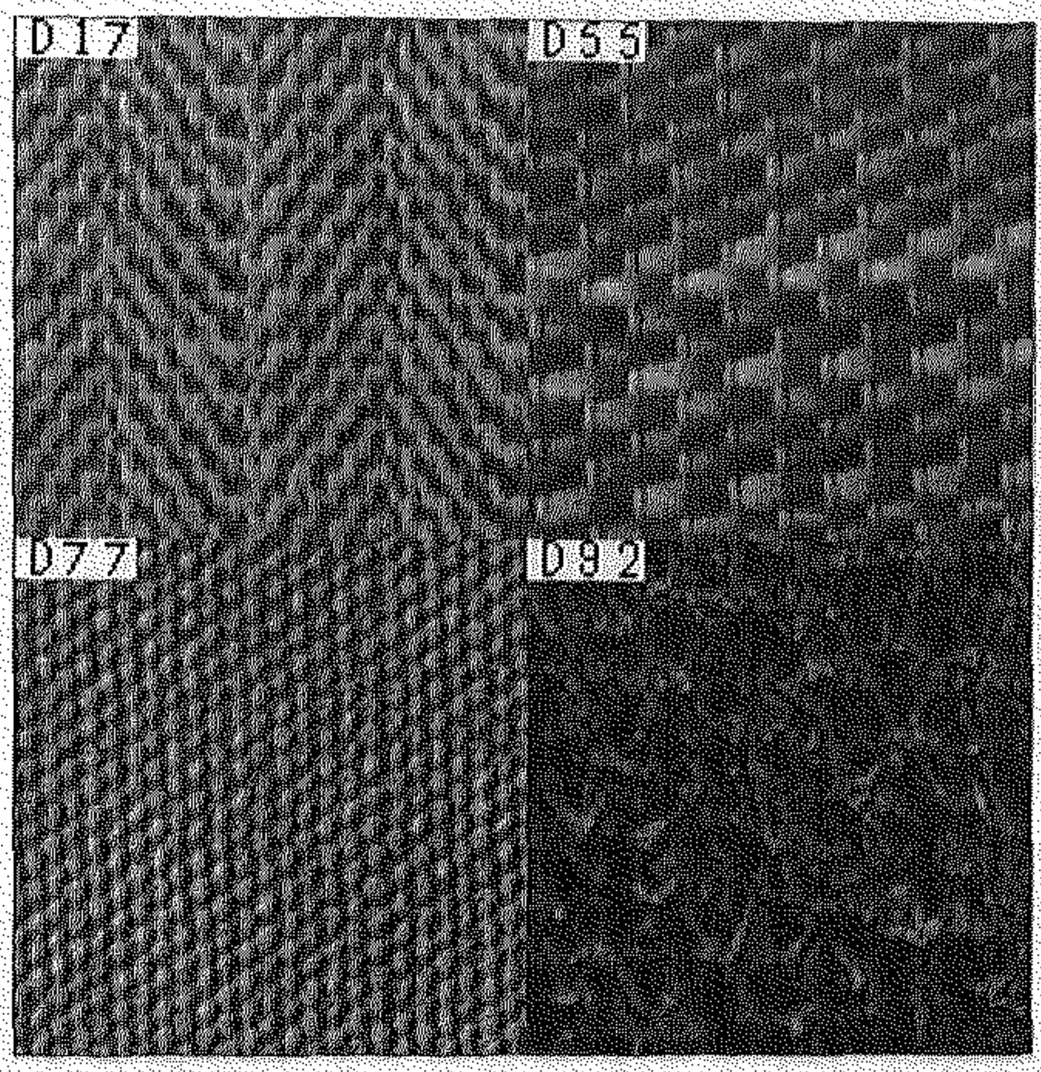


(b_1)

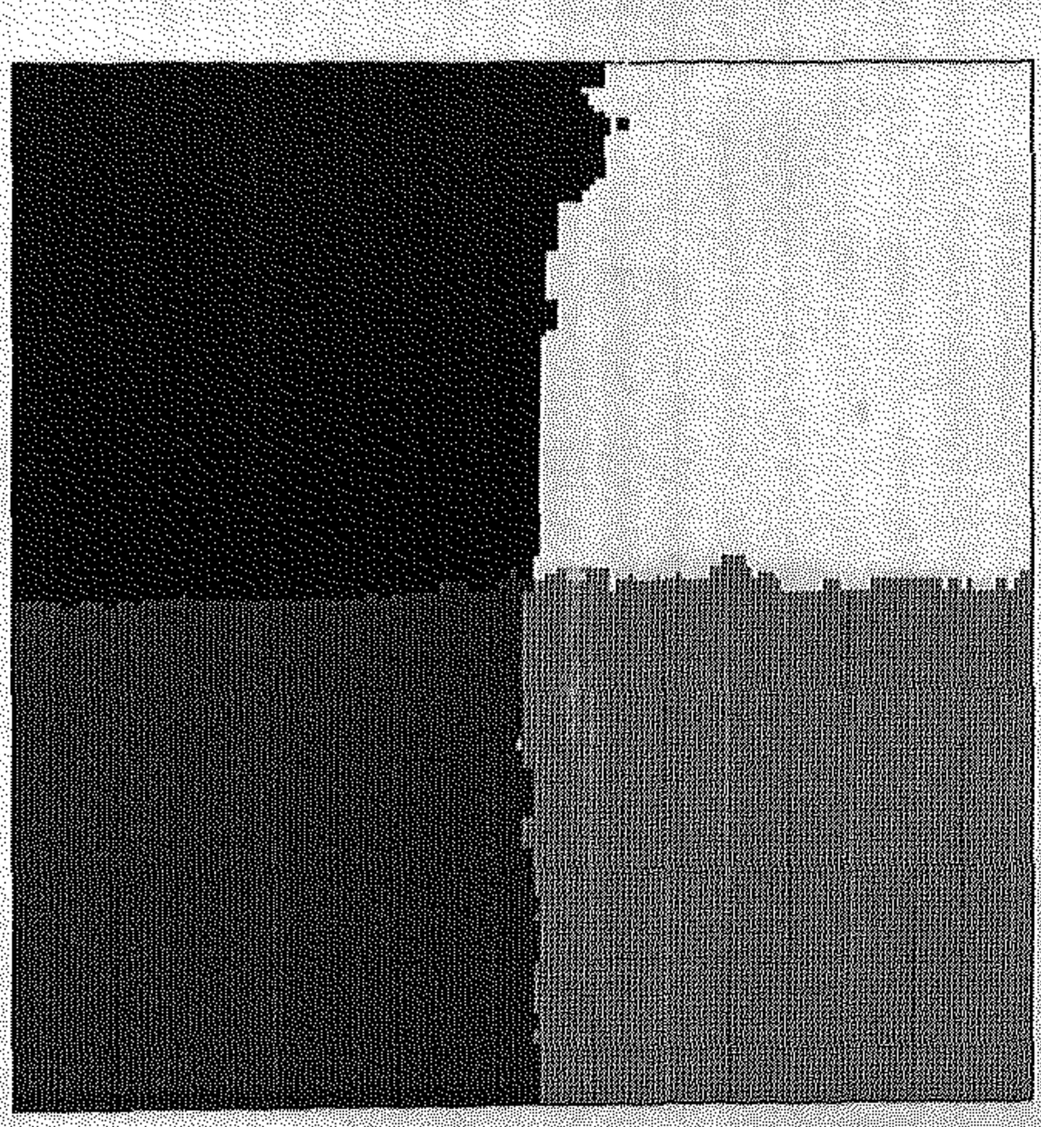


(b_2)

Figure 3.2b : (b_1) Four texture mosaic. (b_2) Segmentation mapping of (b_1).

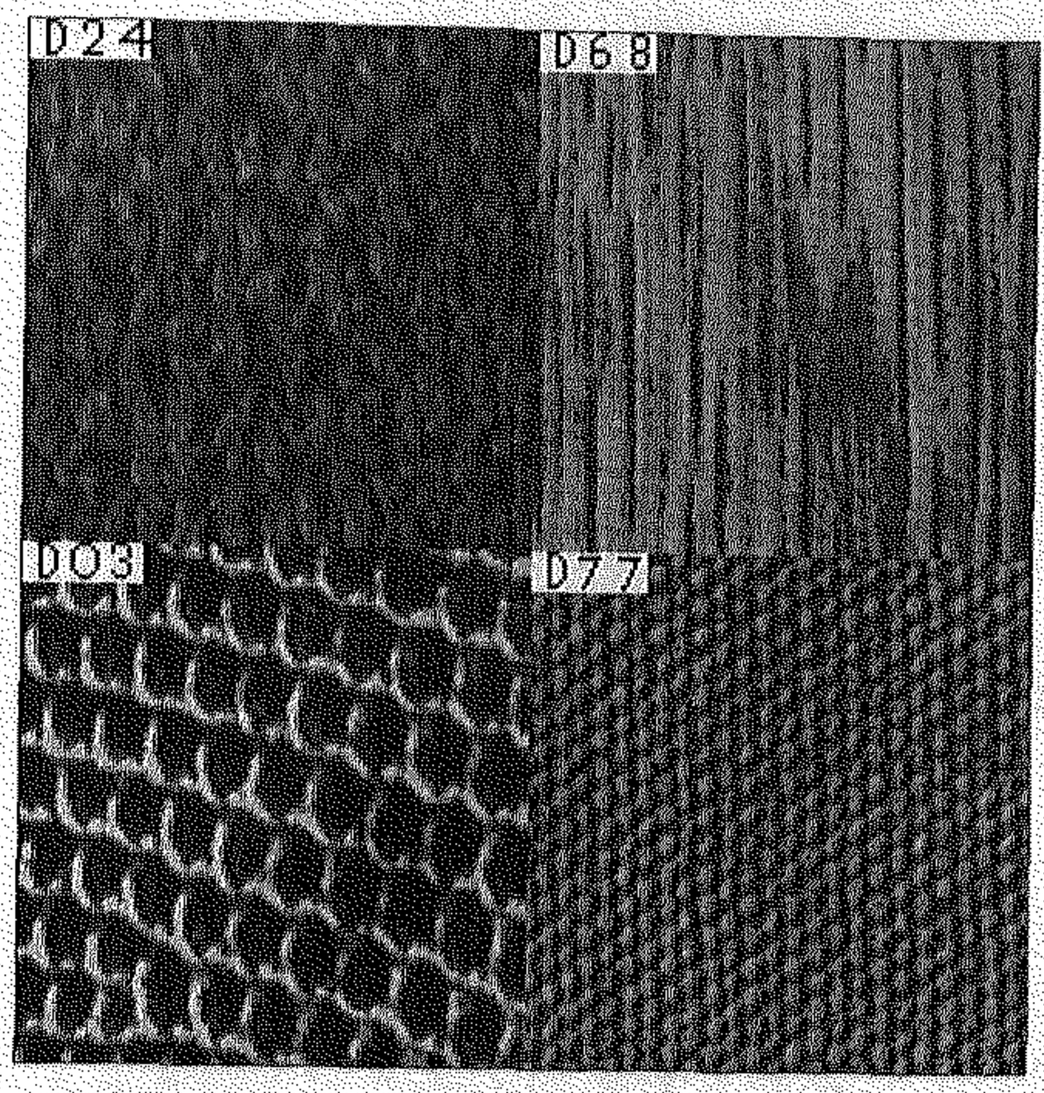


(c_1)

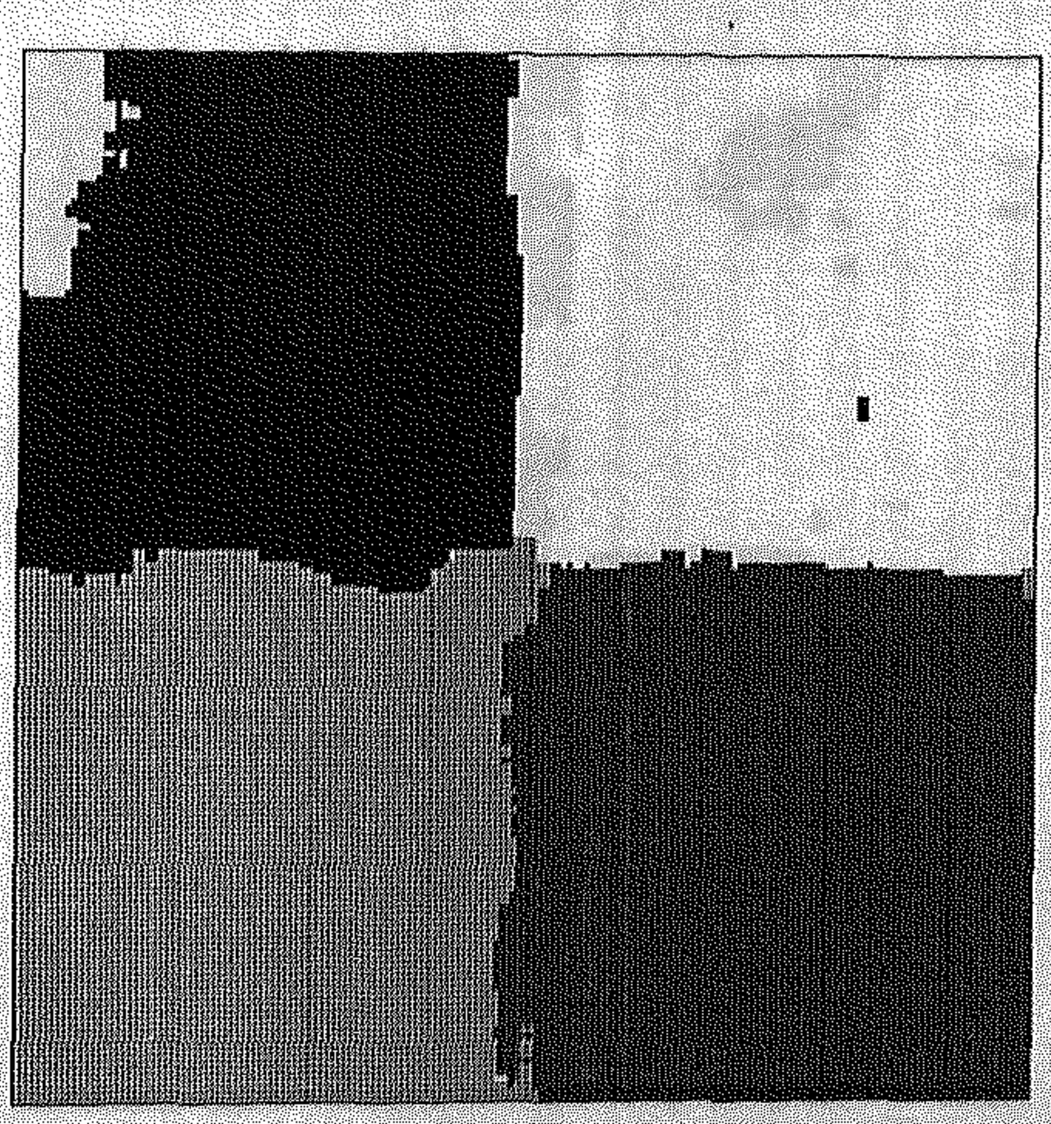


(c_2)

Figure 3.2c : (c_1) Four texture mosaic. (c_2) Segmentation mapping of (c_1).

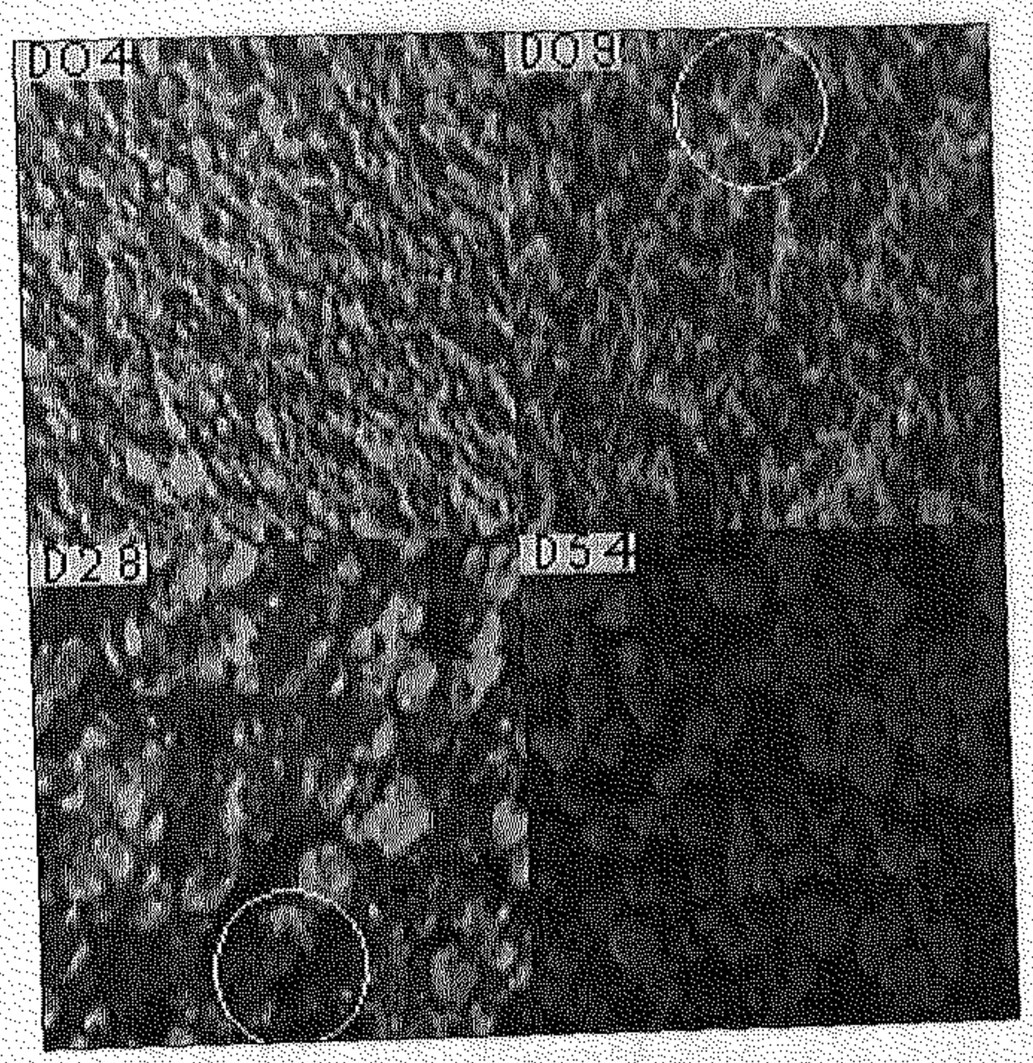


(d_1)

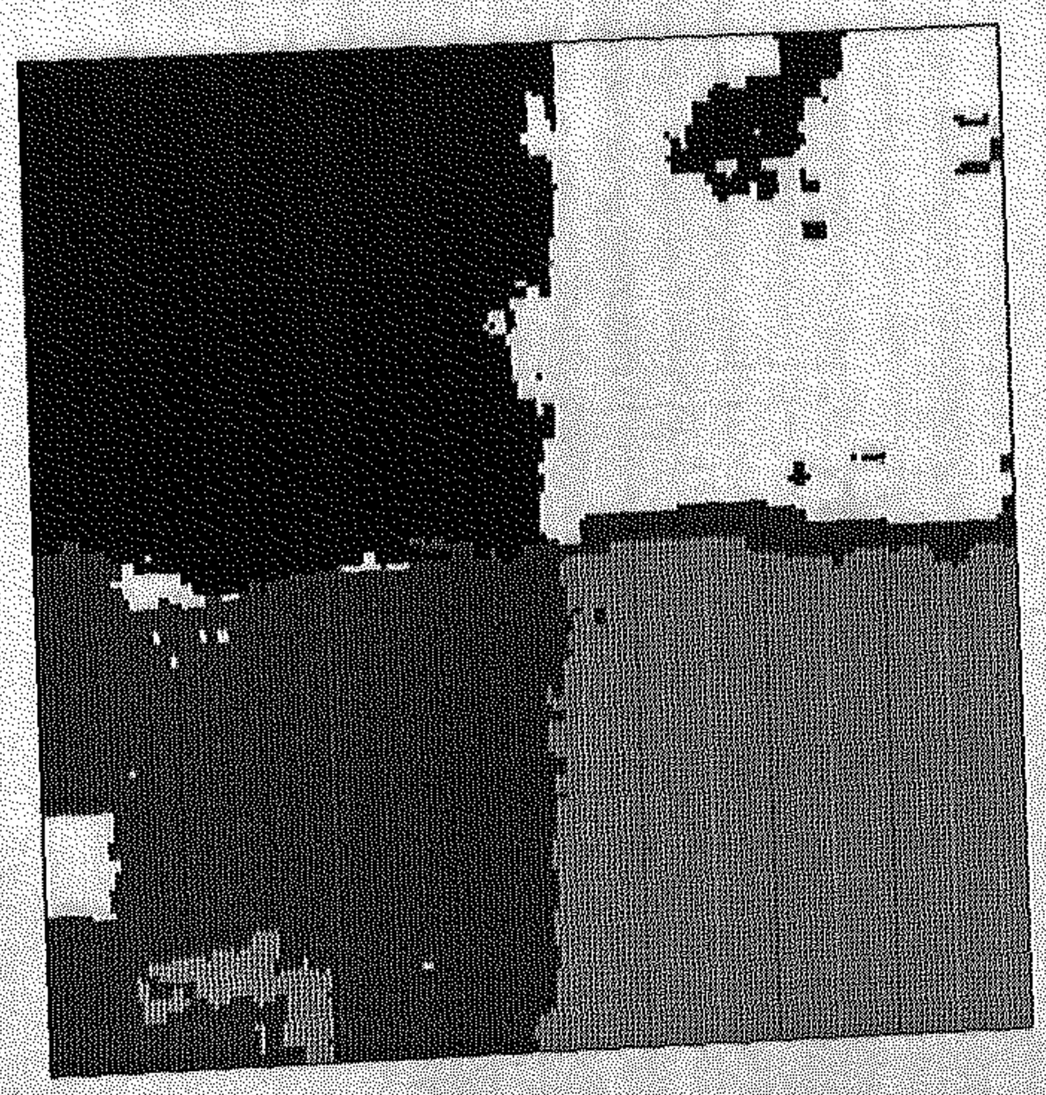


(d_2)

Figure 3.2d : (d_1) Four texture mosaic. (d_2) Segmentation mapping of (d_1).

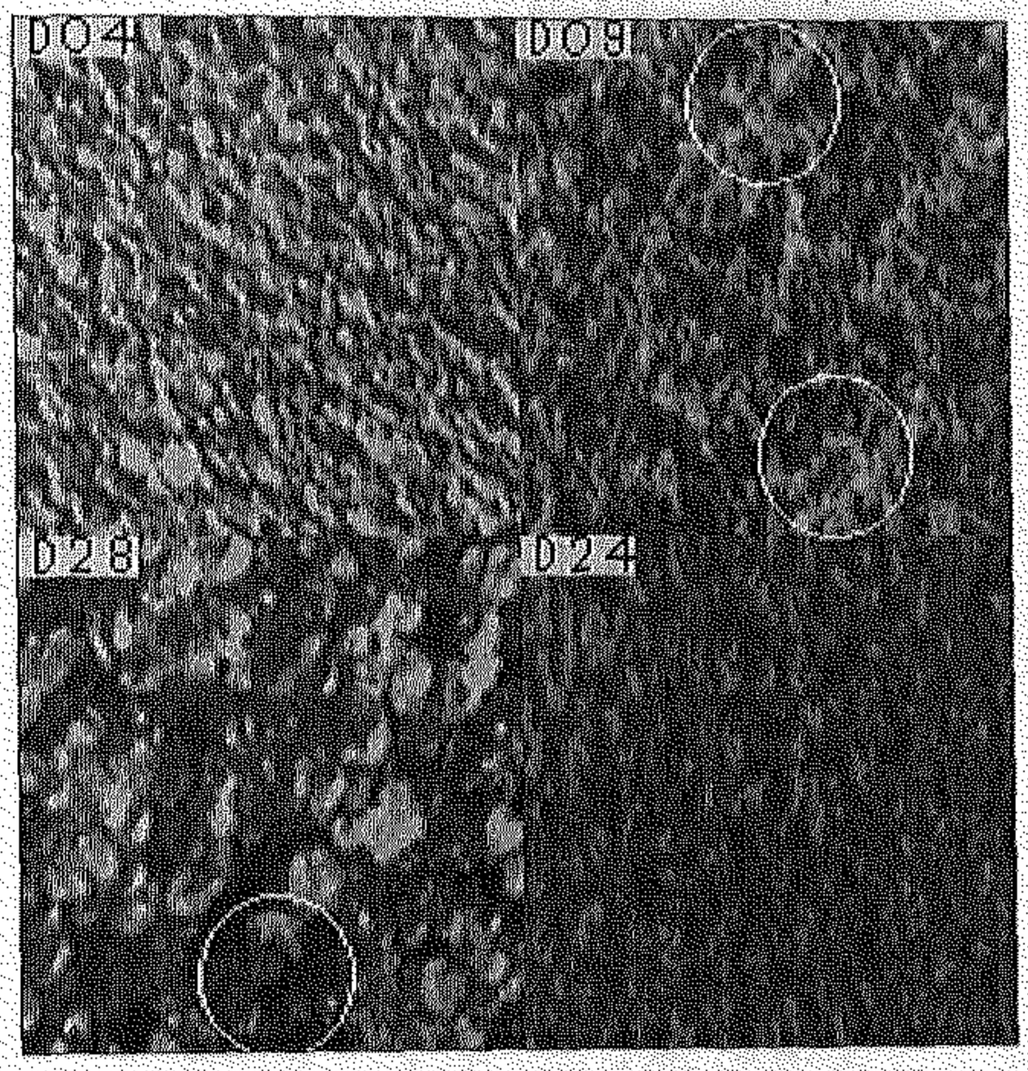


(e_1)



(e_2)

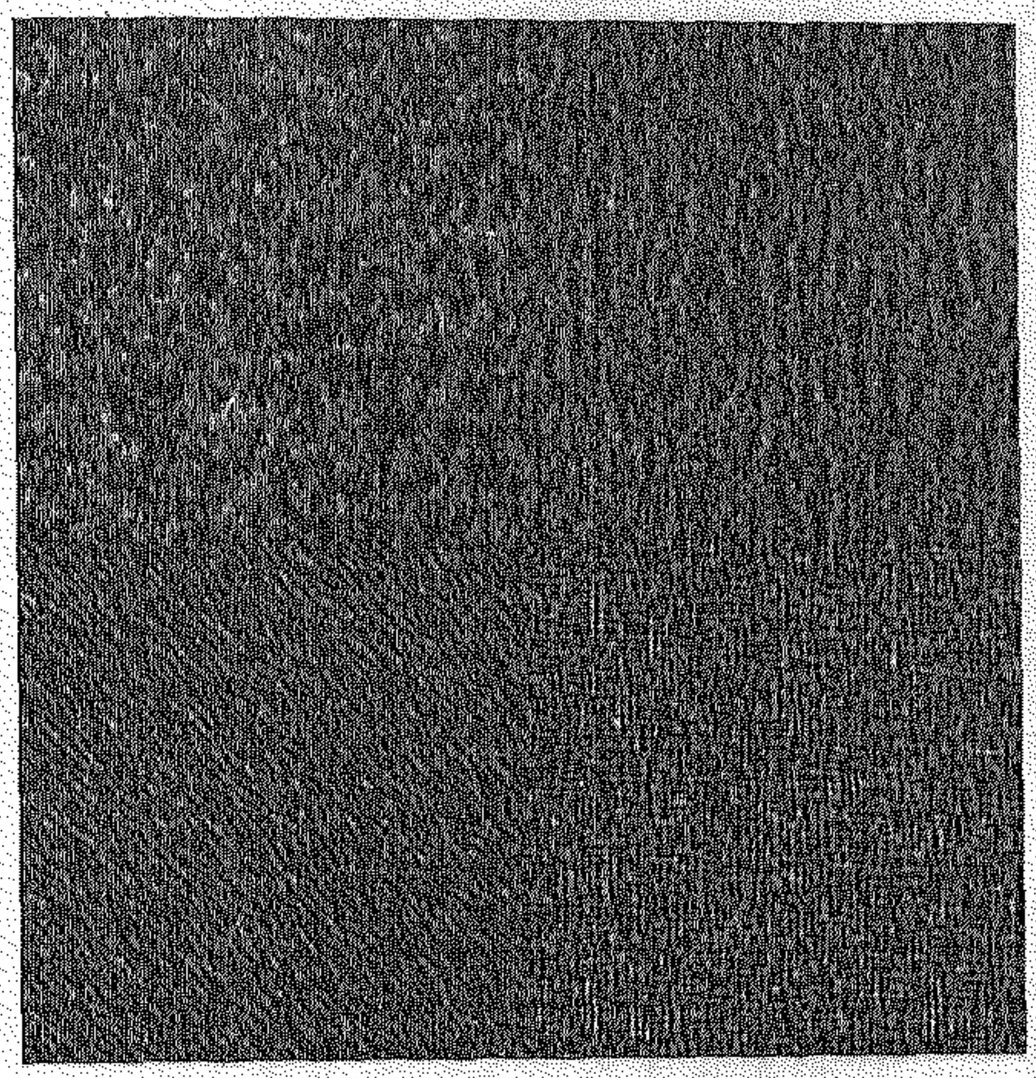
Figure 3.2e : (e_1) Four texture mosaic. (e_2) Segmentation mapping of (e_1).



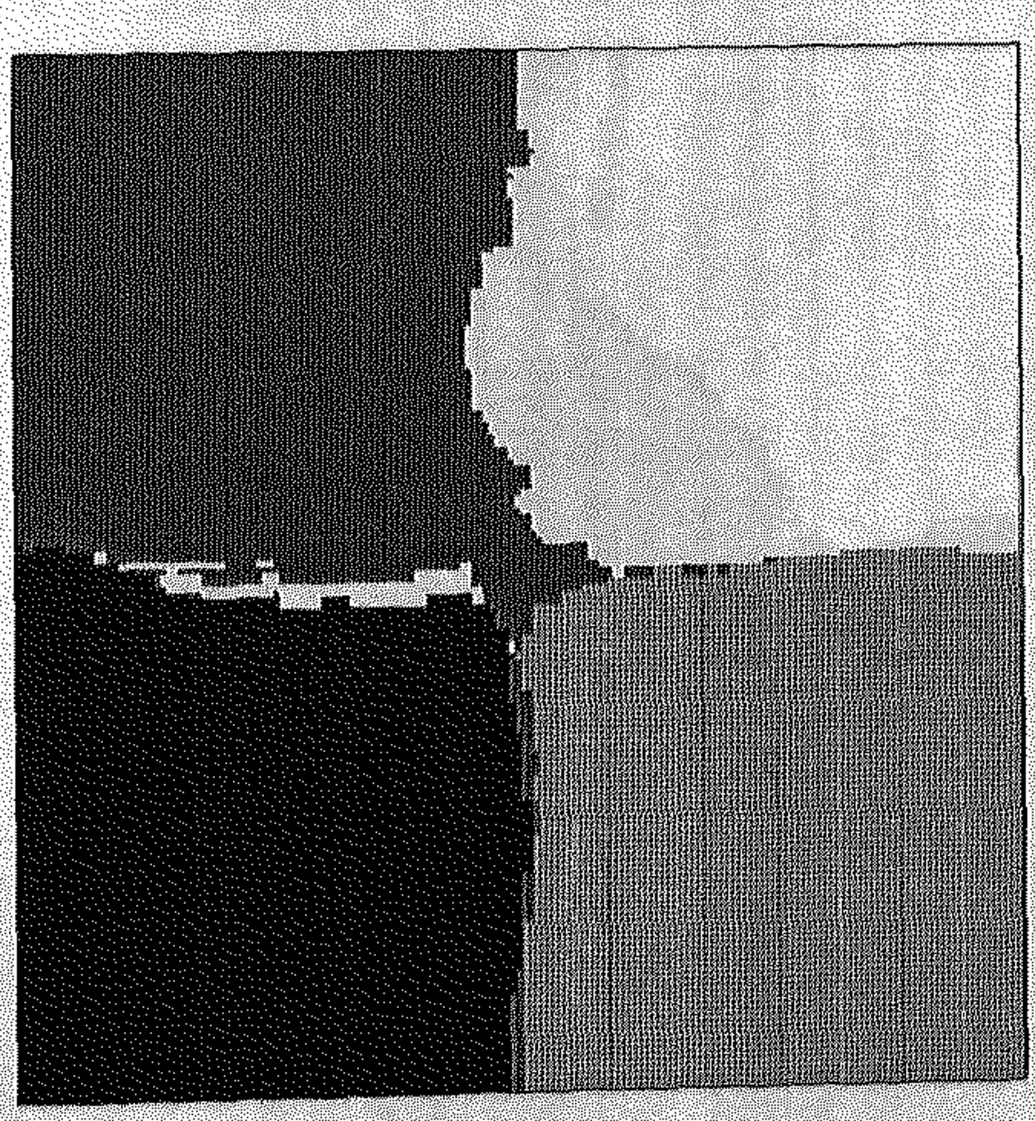
(f_1)

(f_2)

Figure 3.2f : (f_1) Four texture mosaic. (f_2) Segmentation mapping of (f_1).

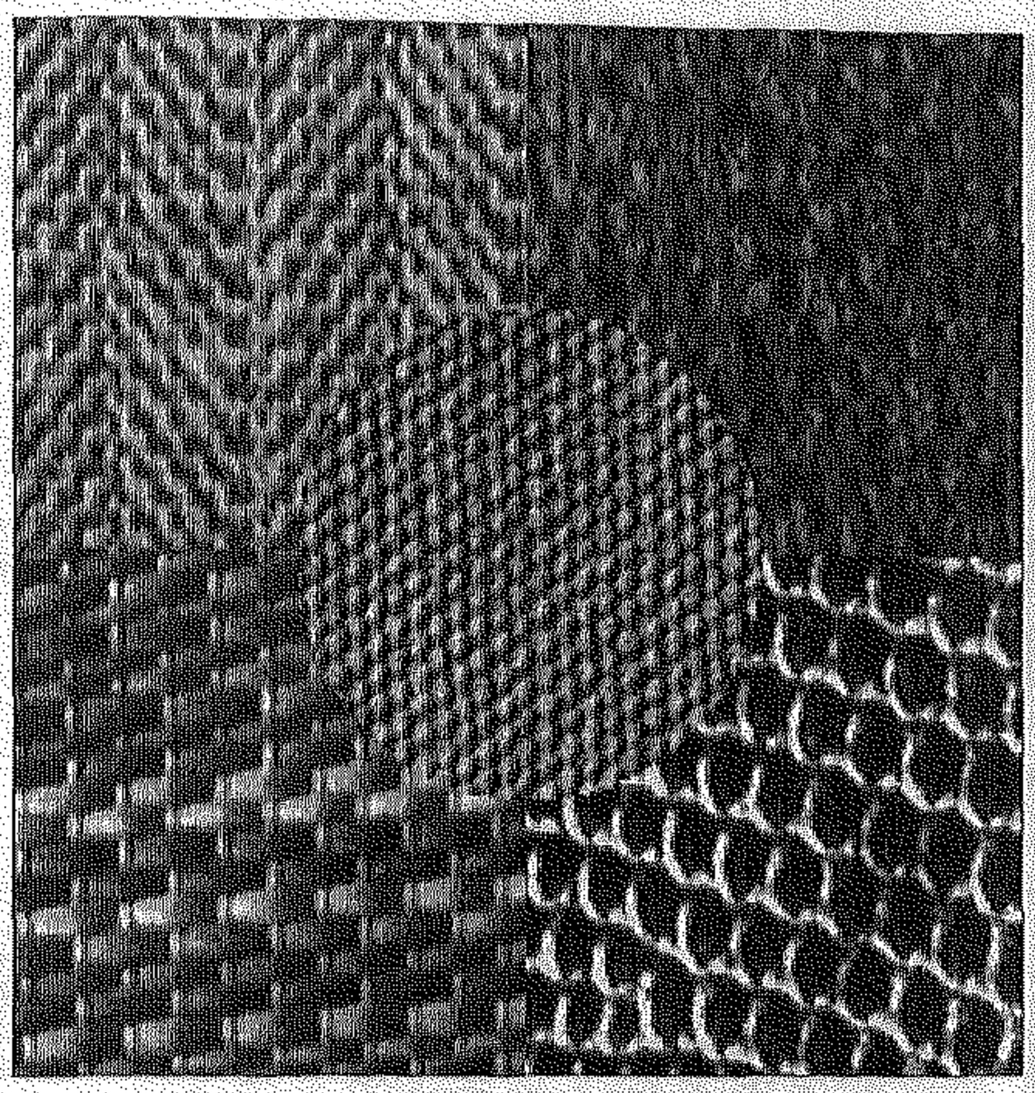


(a)



(b)

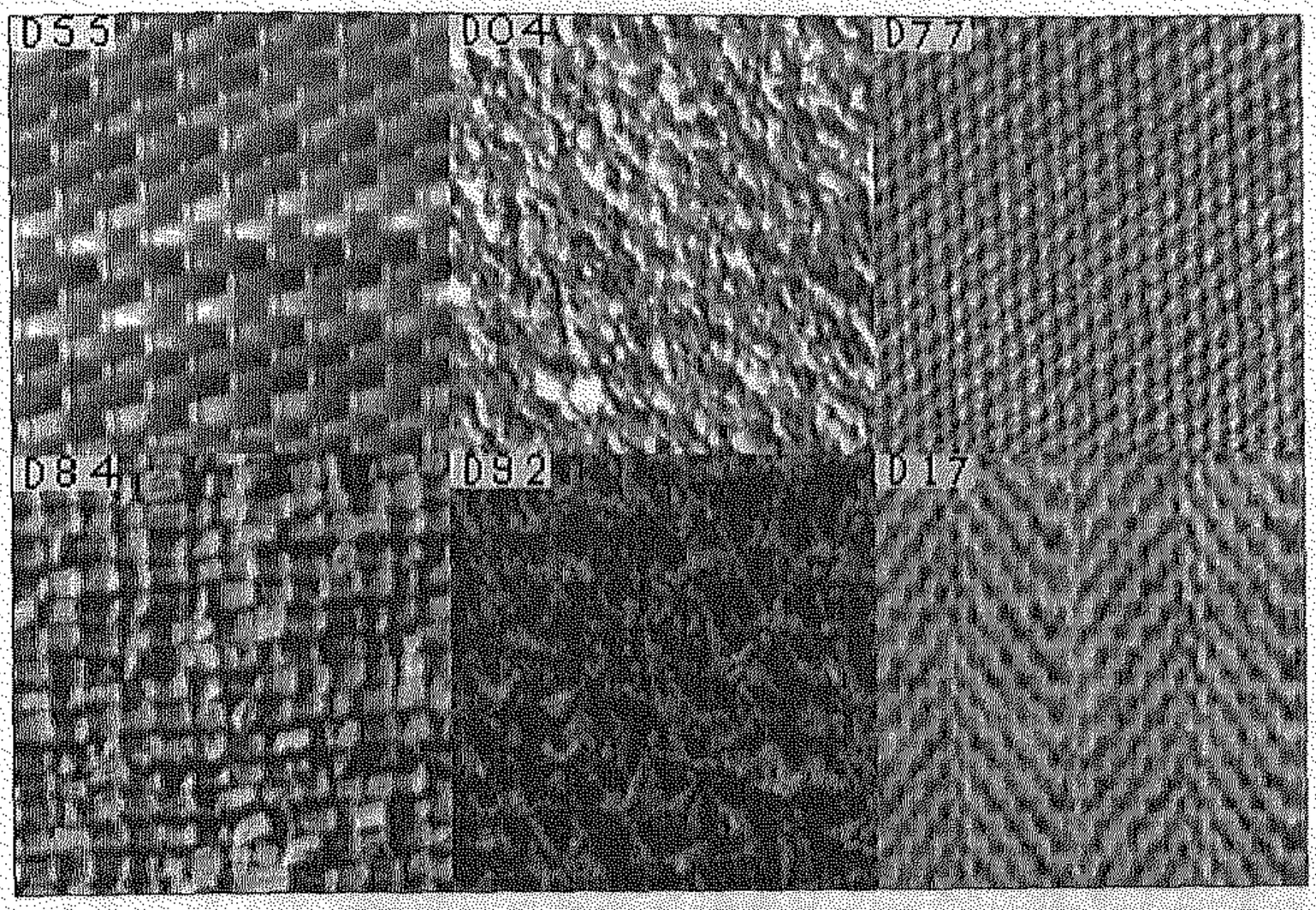
Figure 3.3 : (a) Four texture mosaic generated by Gaussian Markov Random Field model. (b) Segmentation mapping of (a).



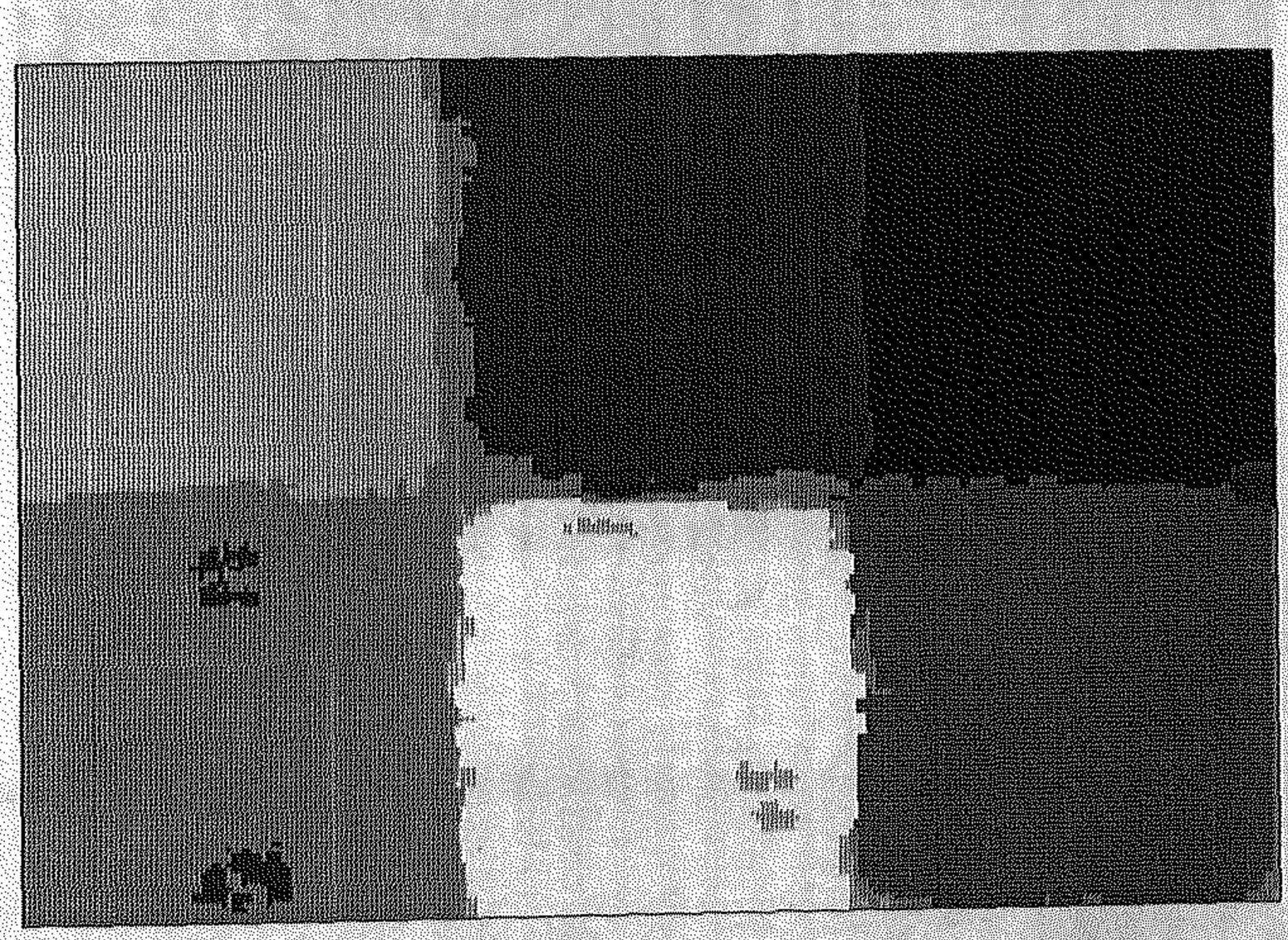
(a)

(b)

Figure 3.4 : (a) Mosaic of five natural textures. (b) Segmentation mapping of (a).

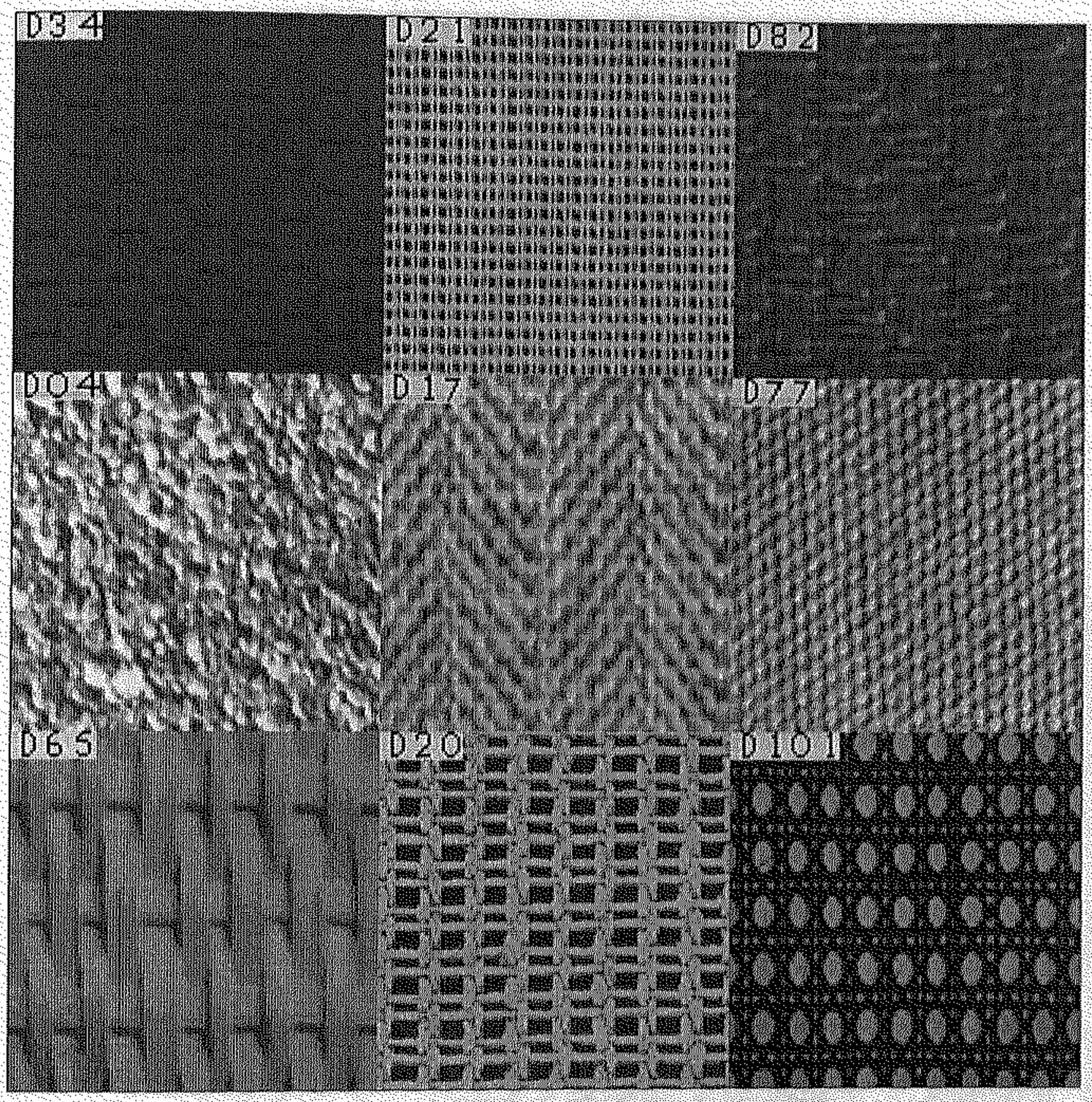


(a)



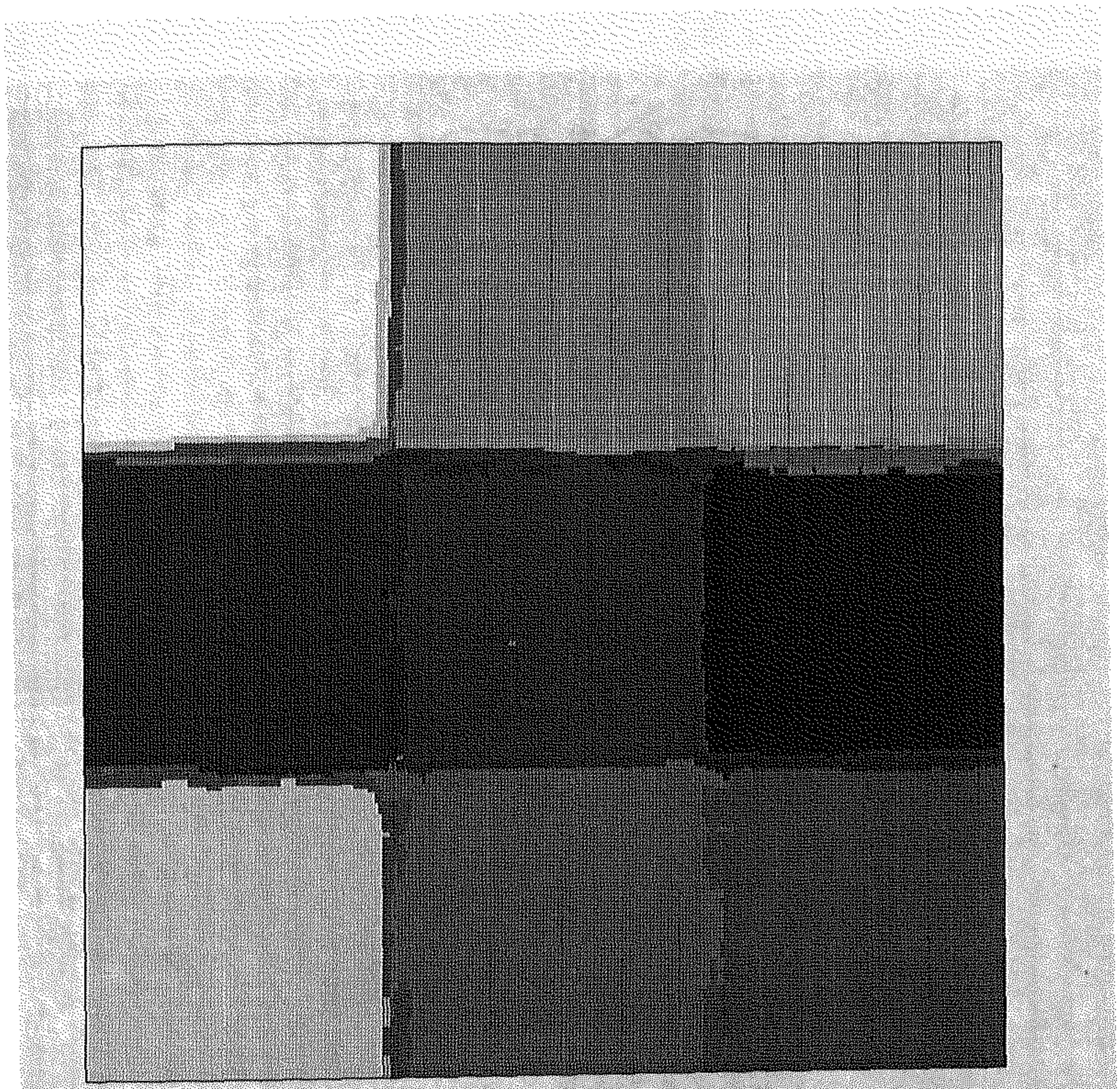
(b)

Figure 3.5 : (a) Mosaic of six natural textures. (b) Segmentation mapping of (a).



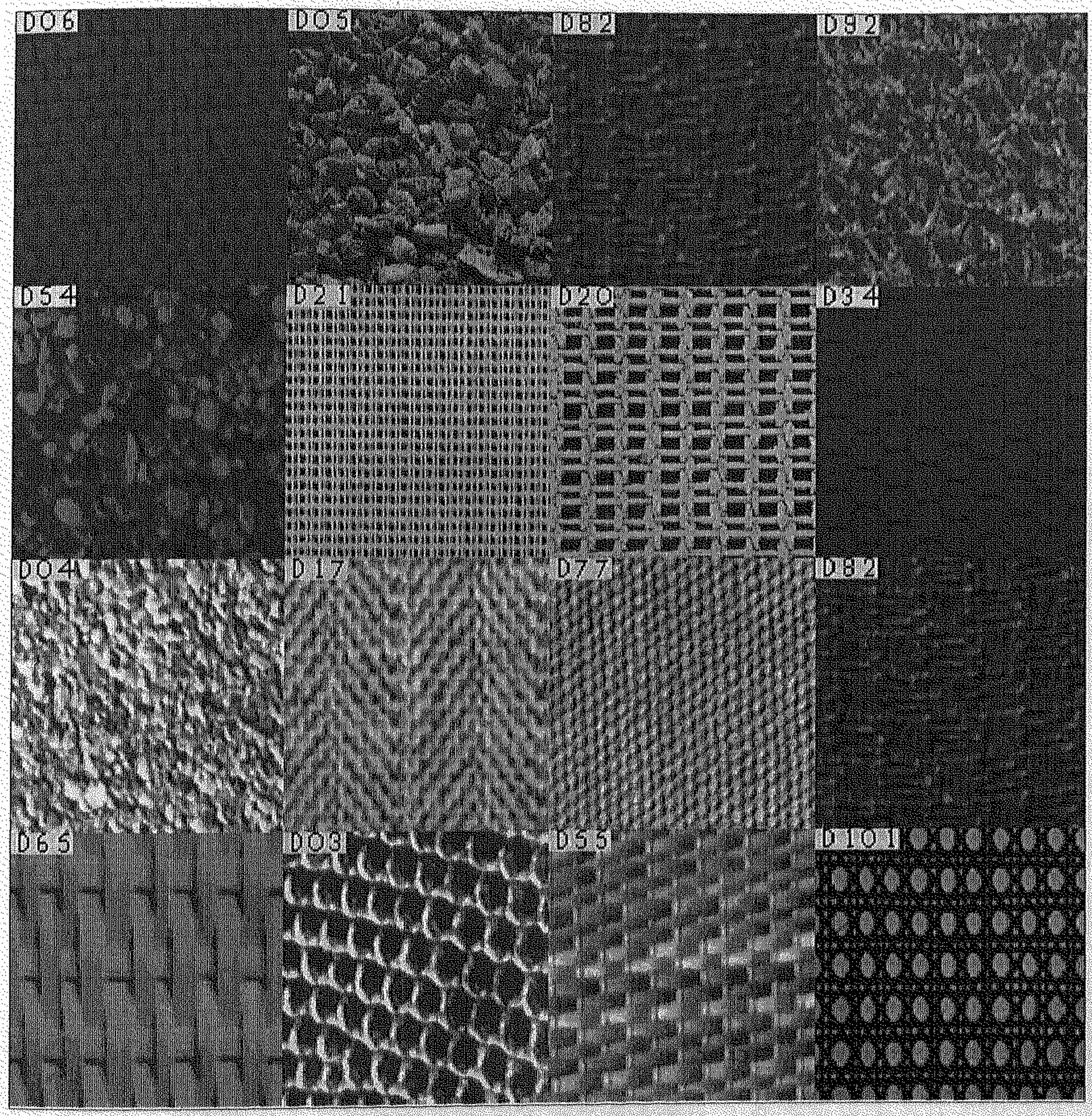
(a)

Figure 3.6 : (a) Mosaic of nine natural textures. (b) Segmentation mapping of (a).



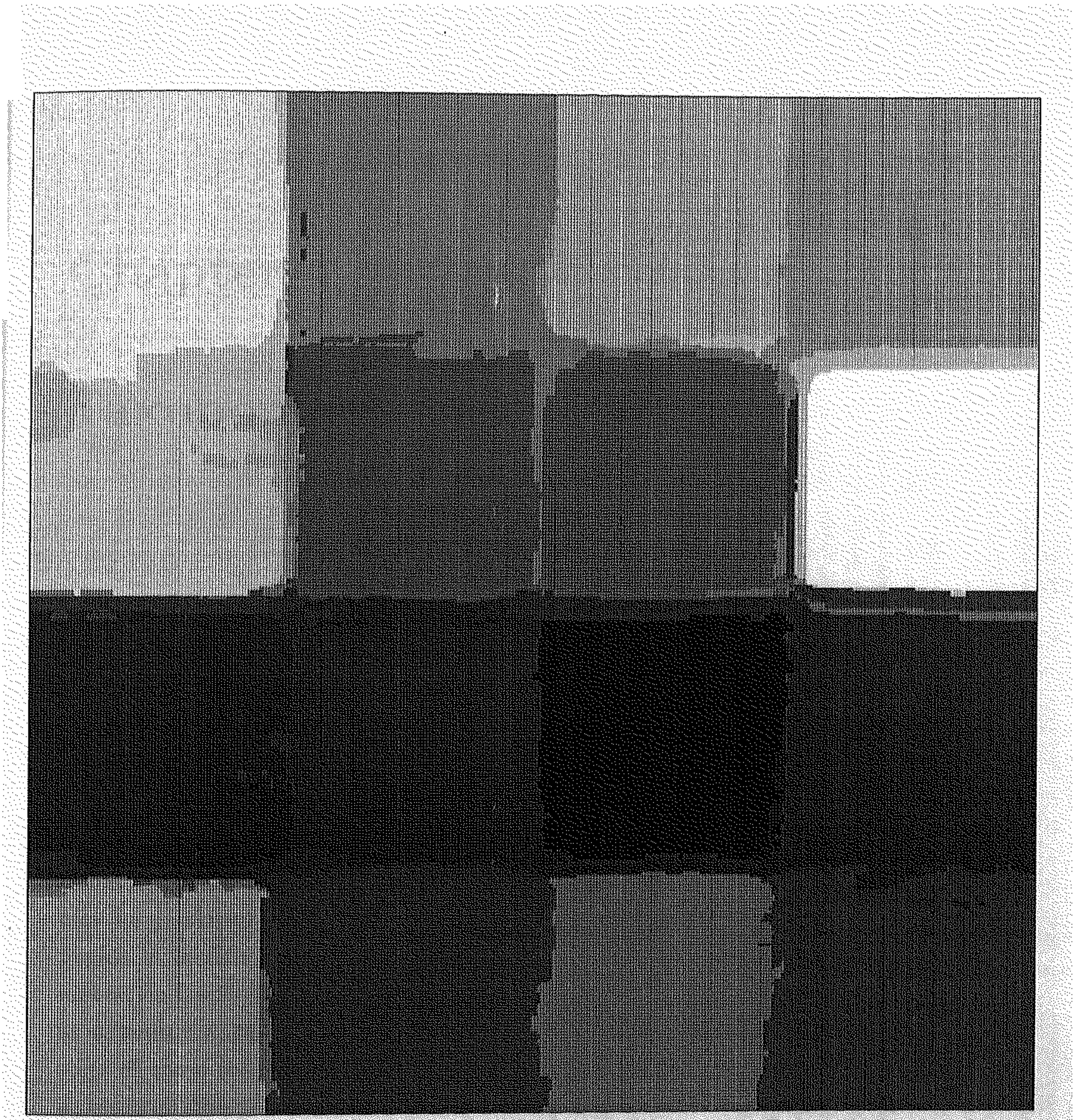
(b)

Figure 3.6 : *Contd.*



(a)

Figure 3.6 : (a) Mosaic of sixteen natural textures. (b) Segmentation mapping of (a).



(b)

Figure 3.7 : *Contd.*

Chapter 4

Taxonomical classification

4.1 Introduction

Textures can be broadly classified into three categories namely, strongly ordered, weakly ordered and disordered texture. Strongly ordered or structural textures are those which exhibit either a specific placement of some primitive elements or comprised of *distribution of a class of elements*. Weakly ordered or oriented textures are characterized by a dominant local orientation at each point of the texture field, which can vary smoothly from one place to other. Disordered or random textures are those that *neither show repetitiveness nor* orientation and may be described on the basis of their roughness. Most of the reported research papers in texture analysis involve texture classification [34,40,46,59,81] and segmentation [19,23,22,32,47,66,72,82]. These analyses are based *mainly on statistical* or structural approach. The goal of the statistical approach is to estimate parameters of some random process. Structural approaches try to describe a texture in terms of the primitive elements and the placement rules that define the geometric relationships between these elements [162,102]. Structural approaches are suitable mainly for describing the structural or strongly oriented textures, while statistical approaches are more suitable for describing the disordered or random textures. *Weakly ordered* or oriented textures cannot be modeled very well by

statistical or structural methods. More recently, attention has been paid to flow-like textures [84,138] and features based on flow-field and coherence measures are proposed [136] for describing oriented textures.

Thus, to choose an appropriate algorithm for further processing of a particular texture, it is useful to know apriori if it is a structural, random or oriented texture. Such taxonomical classification is the main concern of this chapter. In the approach proposed here, texture images are initially divided into two groups, namely directional and non-directional (random). Then the directional textures are further classified into two groups, *e.g.*, structural and oriented/flow-like textures. The approach of classification is based mainly on two features namely texture directionality measure and a measure based on generalized dimension. Here we describe the texture directionality measure in Section 4.2, while, the generalized dimension is already described is Section 2.4. The stepwise implementation of taxonomical classification procedure is described in Section 4.3. The classification results are also presented for a set of natural textures.

Another important problem is addressed here. When we look at a certain texture image, we feel that a minimum area of the image should be 'seen' to get a clear notion of the texture. For example, the texture of scattered pebbles need larger area for characterization than the texture of beach sand, if imaged at the same resolution. This is simply because the pebbles are much bigger than the sand particles. However, all textures are not made of simple primitives like pebbles or sand particles and other properties play prominent role in this context. We address the problem of finding automatically the acceptable size of a texture field that contains the full characterization of the texture. The technique is described in Section 4.4 where the taxonomical classification by the approaches of Section 4.3 is used. Experimental results on a set of natural texture images are also presented in Section 4.4.

4.2 Texture directionality measure [21]

Textures which have a dominant local orientation everywhere, may be called *oriented textures.* Analysis of orientation may be applied to practical problems such as detection of defects in wood and metal strips under bend and stress, regeneration of a missing portion in some texture field, etc. Also, continuous spatial orientation in texture is connected to the concepts of flow-field and coherence measures which are considered to be intrinsic properties of images.

4.2.1 Background

The reason of our ability to detect orientation has been understood from the discovery due to Hubel and Wiesel [68] that the mammalian visual cortex contains orientation selective cells. Physiological study of orientation selective mechanism [148] and psychological study of oriented pattern perception [54] are reported in the literature. Julesz and Bergen [78] introduced the notion of textons in texture perceptron, which can have properties such as color and orientation.

Several authors reported methods of detecting texture orientation. Zucker [176] reported an approach based on the combination of outputs of several linear operators. Kanatani [79] used integral geometry to determine surface orientation from the intersection of scan lines with the texture curves. Bajcsy [7,6] used Fourier power spectra to detect texture directionality.

Kass and Witkin [84] used the following method to find the dominant direction of a texture. At first, they used Laplacian of Gaussian to compute the gradient vector field. Consider the vector in the complex plane formed by combining G_x and G_y as $(G_x + iG_y)$. Let the vector have the polar representation $Re^{i\theta}$. The square of this vector is $R^2e^{2i\theta}$. Squaring gradient vectors at the point in opposite directions make them reinforce each other. The x and y components of J are denoted as

$$J_x(m,n) = G_x^2(m,n) - G_y^2(m,n)$$

$$J_y(m,n) = 2G_x(m,n)G_y(m,n)$$

respectively. Next, J_x and J_y are smoothed again using Gaussian filters in order to average the orientation estimated over a neighborhood. Let the smoothed components be J_x^* and J_y^*. Consider the angle $\bar{\theta}_{mn}$ defined by

$$\bar{\theta}_{mn} = \frac{1}{2}\tan^{-1}\left\{\frac{J_x^*(m,n)}{J_y^*(m,n)}\right\}$$

The dominant orientation at the point (m,n) is calculated as $\bar{\theta}_{mn} + \pi/2$.

Rao [136] computed G_x and G_y using gradient of Gaussian and represented the gradient vector in complex plane. Let the gradient vector at a point (m,n) be represented by $R_{mn}e^{i\theta_{mn}}$. Thus the estimate of the dominant orientation $\bar{\theta}$ in a $N \times N$ neighborhood of the image would be given by

$$\bar{\theta} = \frac{1}{2}\tan^{-1}\left\{\frac{\sum_{m=1}^{N}\sum_{n=1}^{N} R^2 \sin 2\theta_{mn}}{\sum_{m=1}^{N}\sum_{n=1}^{N} R^2 \cos 2\theta_{mn}}\right\}$$

The estimated orientation angle at $(m+N/2, n+N/2)$ is then $\bar{\theta} + \pi/2$, since the gradient vector is perpendicular to the direction of anisotropy.

4.2.2 Proposed method of detection and gradation of oriented texture

A different method of orientation detection and gradation based on evidence accumulation as in the Hough transform [65] is adopted here [21]. The directionality analysis is done in three steps namely edge detection, directional histogram construction, and peak detection.

Edge detection

The edge image is formed from the gradient vector of magnitude G and angle ϕ with the horizontal. Consider two masks [13]

$$h_x(i,j) = \frac{2i}{\sigma^2} exp[-(i^2+j^2)/\sigma^2] \tag{4.1}$$

$$h_y(i,j) = \frac{2j}{\sigma^2} exp[-(i^2+j^2)/\sigma^2] \tag{4.2}$$

where $-s < i,j < s$

Determination of mask size s is a critical issue. A big mask needs more computation while a small mask can lead to inaccuracy. An approximate solution to the problem is given by [13]

$$s \approx \sigma \sqrt{-\log(0.005) - 2\log\sigma} \tag{4.3}$$

It is to be noted that at the points on the line defined by $(0,j)$, h_x is zero while at the points on the line defined by $(i,0)$, h_y is zero.

These masks are convolved with the image I

$$G_x = (h_x * I) \quad , \quad G_y = (h_y * I)$$

to get

$$G = (G_x^2 + G_y^2) \quad \text{and} \quad \phi = \tan^{-1}[G_y/G_x] \tag{4.4}$$

The gradient vector (G,ϕ) is computed on a mask of size $s \times s$ at each pixel of the image using eqs.(4.1)–(4.4). The resulting image can be called edge image E.

Dominant direction histogram construction

For a $(2w+1) \times (2w+1)$ pixel window W, centering at (m,n) of the edge image E, an array of 180 accumulator bins are defined for directions at 1° interval. Let

$A_{\theta(m,n)}$ denote the accumulator bin for edge contribution along $\theta°$ direction.

The contribution of a pixel $(i,j) \in W$ in the accumulator $A_{\theta(m,n)}$ is $G(i,j)\cos^2(\theta - \phi_{ij})$ where ϕ_{ij} is the angle of the gradient at (i,j). The contributions from all pixels in the window can be accumulated as

$$A_{\theta(m,n)} = \sum_{i=m-w}^{m+w} \sum_{j=n-w}^{n+w} G(i,j)\cos^2(\theta - \phi_{ij}) \;\; ; \quad \theta = 0°, 1°, 2°, \ldots, 179° \qquad (4.5)$$

Define the angle image $\bar{\theta}(m,n)$ so that

$$A_{\bar{\theta}(m,n)} \geq A_{\theta(m,n)} \qquad \forall\theta \qquad (4.6)$$

Thus, $\bar{\theta}(m,n)$ denotes the dominant edge direction in the window around the pixel (m,n).

Define a quantity μ as

$$\mu(m,n) = \left[\frac{A_{\bar{\theta}(m,n)}.180}{\sum_{\theta(m,n)} A_{\theta(m,n)}}\right] - 1 \qquad (4.7)$$

Another array, called *Direction Histogram array H* is used. The array consists of 180 elements, where the k-th element H_k accumulates the number of pixels at which the dominant direction is k degrees. However, the plot of H_k against k often does not show a prominent peak even in strongly directional texture image. Hence, a modified histogram is generated by computing H_k' as follows.

1. Initialize $H_k' \leftarrow 0 \qquad \forall\ k$

2. For each k and $\forall\ (m,n)$ do

$$H_k' \leftarrow H_k' + \mu(m,n) \qquad (4.8)$$

Note that μ is always positive since the bracketed quantity on the right hand side of the eq (4.7) is always greater than 1. Here μ acts like an additive weight to the histogram array elements. In this way we have consistently found sharp peaks for directional textures, whose positions are detected by the following approach.

Peak detection

The histogram is smoothed by moving average filter to get rid of spurious peaks. The window size used to compute the moving average is 11. Although most of the spurious peaks are removed by the process, some unwanted peaks and valleys may still remain in the histogram. These peaks and valleys can be discarded by a simple thresholding approach. Let the average height of the histogram denote the threshold T_1, *i.e.*,

$$T_1 = \sum_k H'_k/180 \tag{4.9}$$

Then the following steps are taken:

Step(1): Choose the peaks above and valleys below the threshold (T_1). Discard any valley above and peak below the threshold T_1. If 'v' denotes a valley and 'p' denotes a peak, then the chosen extrema can be represented by the string of the type

vvvv.....pppp.....vvvv....

From each cluster of peaks choose the one with maximum height. Here, P_i denotes the i-th chosen peak and its height is denoted by h_i.

Step(2): A measure of cut-off ratio of P_i can be defined as

$$f_i = \frac{h_i - T_1}{h_i} \tag{4.10}$$

where T_1 is given by eq (4.9).

Dominant peaks of the histogram are considered as global orientation of the texture. It should be noted that the number of such orientations should not exceed two, since our capability of detecting three or more global orientation is poor. To suppress the display of irrelevant global peaks, a threshold on f_i is considered. Peaks P_i for which $f_i < 1/3$ are discarded. As a result, the number of peaks will not exceed 2 and at the same time irrelevant global peaks will not be accounted.

4.3 Taxonomical Classification

4.3.1 Procedure

For any pattern classification task we need *apriori* knowledge about the classes. In the proposed scheme we compute a threshold value during the training phase and use it during classification phase. For this purpose we considered 5 directional and 5 structural texture images as the training set S. At first, for each of these images the local dominant direction $\bar{\theta}(m,n)$ in a sub-image of window W_1 is calculated by using eq (4.6). Then the histogram array is computed using eq (4.8). Let the number of peaks in the histogram be N. The total peak strength $P(W_1)$ is calculated as

$$P(W_1) = \sum_{i=1}^{N} h_i(W_1) \tag{4.11}$$

where $h_i(W_1)$ is the height of the i-th peak using a window of size W_1. Note that the number of peaks $(N)$is not allowed to be more than 2, as argued earlier.

The minimum peak strength P_{min} is calculated as $P_{min} = \min_S P(W_1)$. The threshold is defined as

$$T = 2 \times P_{min} \log 3 \tag{4.12}$$

At the classification stage, the test image is taken and $P(W_1)$ is computed. The grade (G) of the test image is defined as

$$G = \tanh\left(\frac{P(W_1)}{T}\right)$$

where T is computed at the training phase using eq 4.12. If $(G < 0.5)$ for a test image then it is classified as a random texture. On the other hand, the following steps are taken if $G \geq 0.5$. The number of peaks N_1, corresponding to the window size W_1 is noted. Another window size W_2 is chosen so that $W_2 > W_1$ and the number of the peaks N_2 corresponding to W_2 is noted. If $N_1 = N_2 = 1$ then the texture

is a **uni-directional** one. Otherwise, the texture is either double-directional or structural texture. Double-directional and structural textures can have two dominant peaks. Usually peak strength of a double-directional texture increases with the window size. In other words, when $(W_2 > W_1)$, we have $(P(W_2) > P(W_1))$. The peaks are also well balanced for this type of texture, *i.e.*, the ratio of the peak heights is nearly 1. For this purpose, another measure called balance of the peaks (B), is defined as

$$B = \frac{\min(h_1(W_1), h_2(W_1))}{\max(h_1(W_1), h_2(W_1))}$$

An *incremental ratio* is also defined as $I = P(W_2)/P(W_1)$. If $I \times B > 1.0$ for a texture then the texture is said to be **double directional**. Otherwise, we classify the texture as a **structural** one.

For random textures another quantity called F_2 is computed. F_2 is derived from generalized dimension (D_q), given in equation (2.12). Here the partition function used is $\chi(q,r) = \sum_{i,j} [n'_r(i,j)]^q \simeq r^{\tau(q)}$ where $r = s/M$, $n'_r(i,j) = n_r(i,j)/N_r$; n_r and N_r are defined by eq (2.7) and (2.8), respectively. We define $F_2 = TD - D_2$, where D_2 is generalized dimension of order 2 and TD is the topological dimension, which is 2 in this case. The role of F_2 in distinguishability of the texture primitives (*texels*) is discussed in the results described below.

4.3.2 Results

For our experiment we considered 32 textures from Brodatz album [17] (See Fig. 4.1). At first, the dominant peaks and peak strengths are computed using 5×5 window. To guide the selection of the value of the threshold T, the total peak strength of all textures in the training set S are computed in the training phase. During the classification phase, the value of G of a test texture image is computed. If $G < 0.5$, the texture is declared as a **random** texture. It is seen that D04, D05, D09, D28, D29, D33, D57, D67, D75, D98 and D109 fall into this category. If $G \geq 0.5$ for a given texture then the dominant peaks are again computed using 9×9 window. The number of dominant peaks are noted for both the window sizes. D12, D15, D24, D37, D51, D65, D68, D76 and D93 have only

Table 4.1: Taxonomical Classification of natural textures

Plate	Texture	Class	Plate	Texture	Class
D03	Reptile skin	ST	D04	Pressed cork	RA
D05	Expanded mica	RA	D06	Woven Aluminum wire	ST
D09	Grass lawn	RA	D12	Bark of tree	UD
D15	Straw	UD	D16	Herringbone woven	BD
D17	Herringbone woven	BD	D20	French canvas	ST
D24	Pressed calf leather	UD	D28	Beach sand	RA
D29	Beach sand	RA	D33	Pressed cork	RA
D34	Netting	ST	D37	Water	UD
D51	Raffia with cotton	UD	D54	Beach pebbles	RA
D55	Straw matting	ST	D57	Handmade paper	RA
D65	Oriental Rattan	UD	D67	Plastic pellets	RA
D68	Wood grain	UD	D75	Coffee beans	RA
D76	Grass fiber cloth	UD	D77	Cotton canvas	ST
D82	Straw cloth	ST	D84	Raffia	ST
D92	Pigskin	RA	D93	Fur	UD
D98	Quartz	RA	D109	Handmade paper	RA

one dominant peak for both the window sizes and therefore, they are declared as uni-directional textures. Among the others, the value of $(I \times B) > 1.0$ for D16 and D17 and hence they are declared as double-directional textures. Both uni- and double-directional textures belong to **directional or weakly ordered** textures. Using our method D03, D06, D20, D34, D55, D77, D82 and D84 are classified as ***structural or strongly ordered*** textures. The classification is systematically presented in Table 4.1, where RA, ST, UD, and BD denote random, structural, uni-directional, and bi-directional textures, respectively.

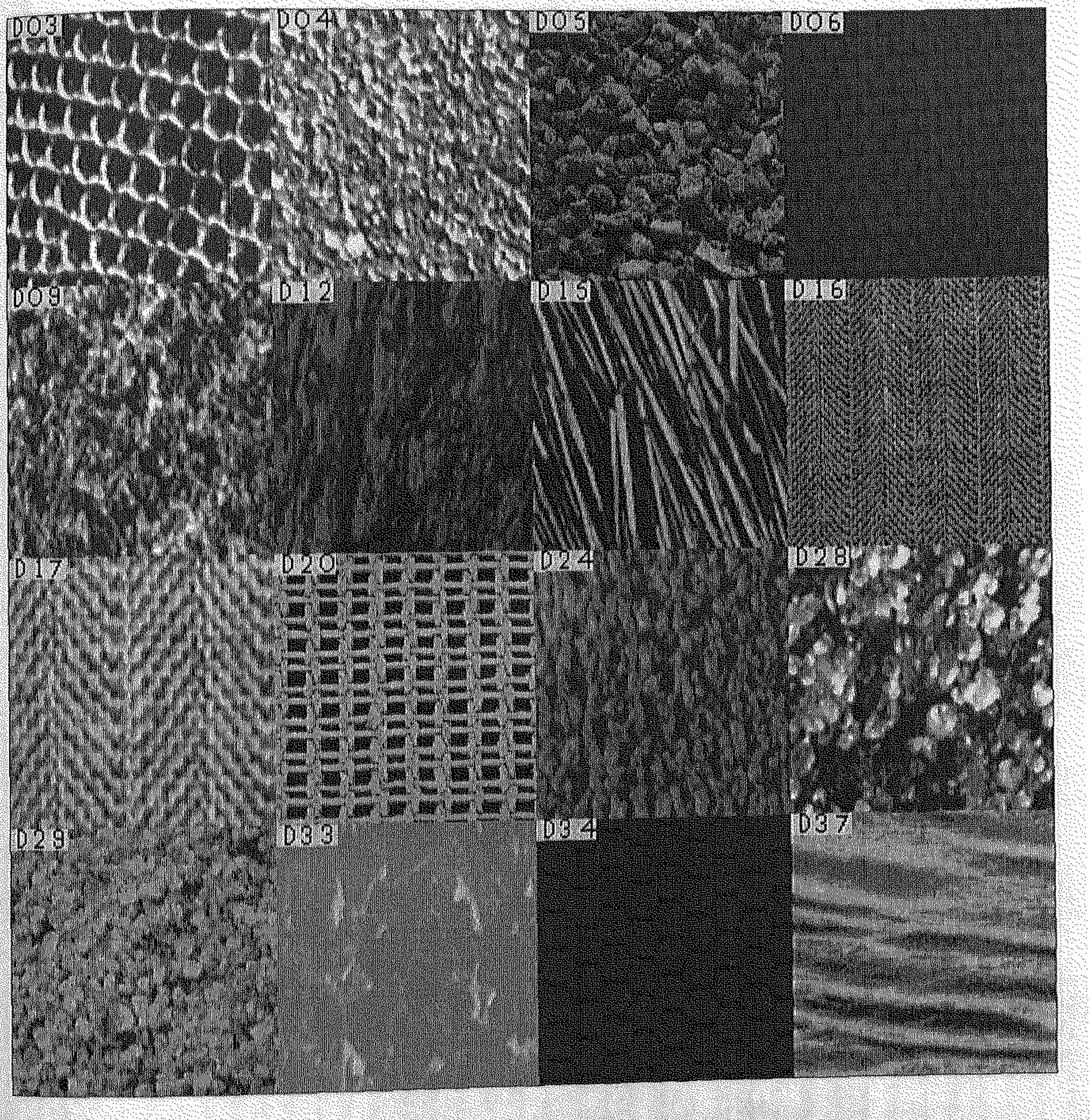


Figure 4.1 : Textures used for the experiment (from Brodatz album).

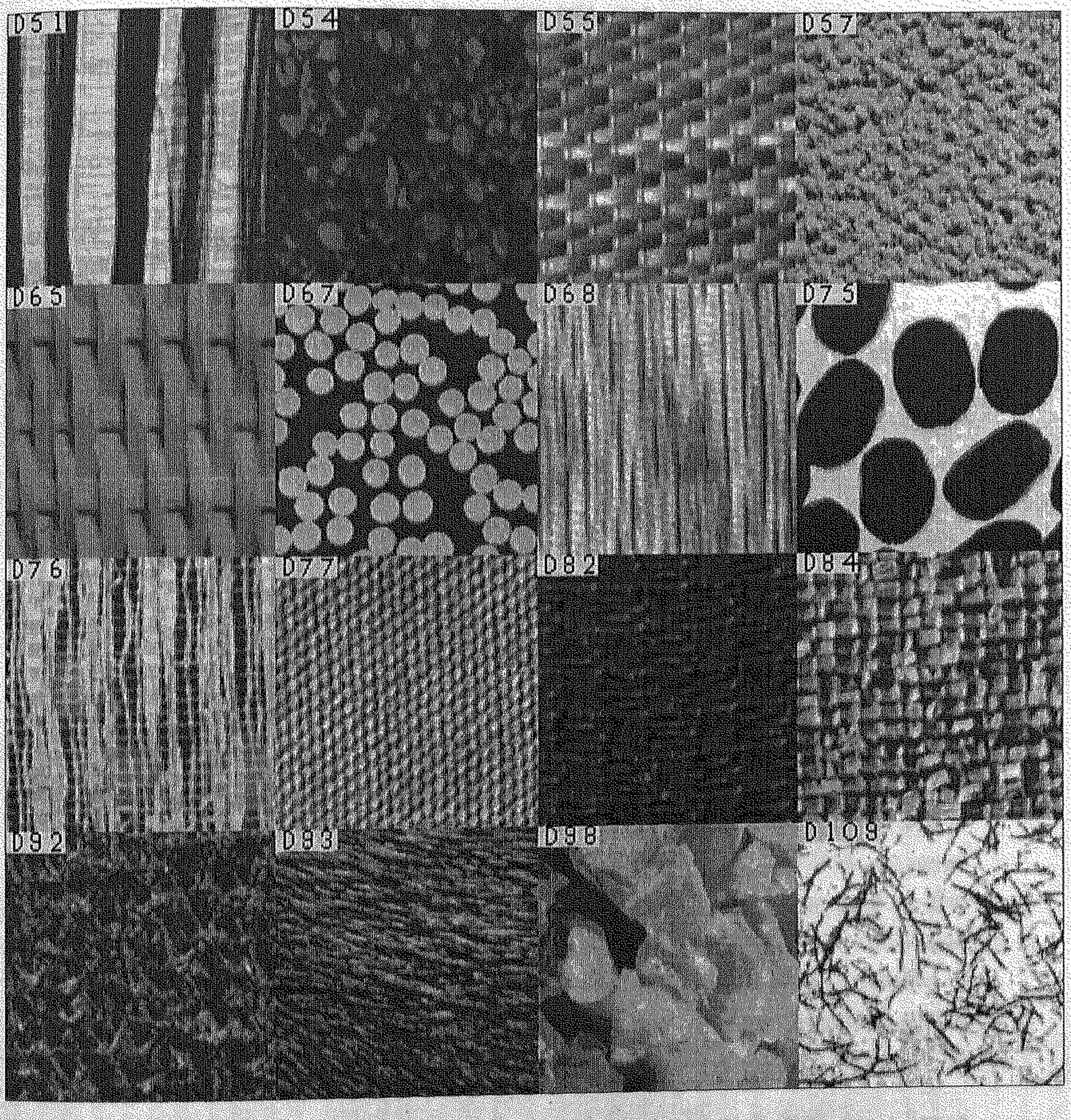


Figure 4.1 : *Contd.*

Table 4.2: Generalized dimension (F_2) of disordered textures

Plate	Texture	F_2
D04	Pressed cork	0.271
D05	Expanded mica	0.478
D09	Grass lawn	0.283
D28	Beach sand	0.661
D29	Beach sand	0.454
D33	Pressed cork	0.944
D54	Beach pebbles	0.532
D57	Handmade paper	0.341
D67	Plastic pellets	0.962
D75	Coffee beans	0.993
D92	Pigskin	0.361
D98	Quartz	0.736
D109	Handmade paper	0.801

On the other hand, F_2 is computed for a random texture. A higher value of F_2 indicates better distinguishability of the *texels*. For example, in Fig. 4.1 the texels of D05 are better distinguished than the texels of D04. From Table 4.2 it is seen that F_2 of D05 is larger than F_2 of D04. Similarly, texels of D75 are more prominent than the texels of D92. A further classification may be done on the basis of F_2. It is easier to segment the prominent texels, while their size and shape properties can be computed to provide micro level description of the texture. A detailed study of size and shape of the primitives is presented in Chapter 5, where the informations are used in the study of rock thin-section.

The classification results using our scheme are satisfactory for a large number of textures. Only one error is found in case of D65 which is apparently a structural texture but is classified as a directional texture. This misclassification

Table 4.3: Pooled ranking of textures on the basis of directionality and structure

Property	Ranking							
Directionality	D68	D51	D76	D20	D65	D15	D55	D16
	D17	D03	D37	D06	D34	D82	D77	D12
	D84	D93	D24	D109	D67	D04	D05	D09
	D92	D57	D75	D28	D54	D29	D98	D33
Structure	D34	D20	D55	D65	D16	D06	D77	D82
	D51	D17	D84	D03	D75	D05	D28	D37
	D67	D54	D68	D15	D109	D09	D76	D24
	D04	D57	D12	D92	D93	D98	D29	D33

is due to the shadow effect which causes a reduction of strength in the direction normal to the dominant direction.

Next, we compared our results with the results obtained from a human psychovisual test. For psychovisual experiments we again consider the same set of textures from Brodatz's album. A group of 20 persons from different educational background and field of interest were chosen for our experiment. The human subjects were asked to rank the textures on the basis of directionality as well as on the basis of structure. The ranking should be such that strongly directional textures would precede the weakly ones. Now, we combine the ranks provided by different subjects as follows. The sum of individual ranks are computed for each of the textures. The texture corresponding to minimum sum is given the highest rank (we call it *pooled rank*). The texture corresponding to the minimum sum among the remaining textures is given second ranking, and so on. The pooled ranking of the textures is shown in Table 4.3 where the upper left corner tops the ranking and the successive ranks are obtained by horizontal scanning. The subjects were also obliged to choose the set of bidirectional textures and rank them according to the balance of the directionality of textures. To compare individual subject's ranking (on the basis of directionality) with our ranking (by computer) we

calculate Spearman's co-efficient of rank correlation [156]. We obtained 0.96, 0.74 and 0.87, respectievly, as the maxiumum, minimum and average rank correlation of individual ranking and our ranking while the rank correlation between our ranking and pooled ranking is 0.85.

The subjects were also told to classify the textures into two groups, *i.e.*, directional and non-directional. The votes given by the subjects are combined by majority logic. Thus, majority of the subjects have chosen D03, D06, D12, D15, D16, D17, D20, D24, D34, D37, D51, D55, D65, D68, D76, D77, D82, D84 and D93 as directional textures which agree with computer ranking.

Also, most subjects decided that D03, D06, D16, D17, D20, D34, D55, D65, D77, D82, and D84 have two dominant directions. It is to be noted that the dominant directions of the bi-directional textures are balanced, whereas that of structural ones are ill balanced. However, some of the structural textures have well balanced dominant directions. The results obtained from the tests reveal that among these textures, D16, D17 and D20 have highly balanced dominant directions.

4.4 Detection of texture field representative size

4.4.1 Procedure

The second problem addressed here is to find the minimum area of the texture image field, which is representative of the texture. It is assumed that the representative texture field has a square shape and hence we need to find the sidelength of this square.

Our procedure is to start with an initial sidelength (ℓ_0) and iteratively modify the value till some criterion is satisfied. Edge repetition array (ERA) [162] is used to find ℓ_0 for the directional and the structural textures, whereas a measure (F_2) derived from generalized dimension of order 2 is used for the disordered

textures. A typical ERA is shown in Fig. 4.2.

The edge repetition array is basically a histogram of frequency of edge pairs of similar orientation versus the separation of such edge pairs. To compute the histogram all edge pairs for a particular edge direction are accumulated by scanning the angle image in a direction perpendicular to the edge direction. For a repetitive texture, this histogram will exhibit periodicity in the form of peaks at nearly regular intervals. Thus, peak interval and height carry useful information about the spatial organization of the texture.

For the structural textures ERA along 6 directions $(0^o, 30^o, 60^o, 90^o, 120^o, 150^o)$ and for the directional textures ERA along only one direction (one of the directions among $0^o, 30^o, 60^o, 90^o, 120^o, 150^o$ is chosen, the one which is nearest to the dominant direction of the given texture) are computed. Now, for the structural textures the following steps are followed. Suppose that the average of the dominant peak spacings of the i-th ERA is δP_i(See Fig. 4.2). Then ℓ_0 is chosen as $2 \times \max_i \delta P_i + 1$. In case of the directional textures, if the average spacing of the dominant peak of the ERA is δP, then the value of ℓ_0 is chosen as $2 \times \delta P + 1$. The value of ℓ_0 for the disordered textures are computed from F_2. We choose ℓ_0 as an odd integer so that there are equal number $(\ell_0 - 1)/2$ of pixels to the left and to the right of a candidate pixel.

The following steps are executed to obtain the final value of sidelength. A maximum allowable error is selected as ϵ.

Step(1): Choose N random points from the image.

Step(2): Around each point consider a subimage of size $\ell_0 \times \ell_0$ and estimate the Fractal dimension. Let the Fractal dimension of the i-th point be x_i.

Step(3): Compute $\sigma^2 = \frac{1}{N}\sum(x_i - \bar{x})^2$

Step(4): if $\sigma = \epsilon$, stop //sidelength $=\ell_0$ is returned//

Step(5): if $\sigma < \epsilon$ goto *Step*(11)

Step(6): Compute w as a function of ℓ_0 and σ //$w = \ell_0 + \ell'$ where $\ell' \propto \sigma - \epsilon$ //

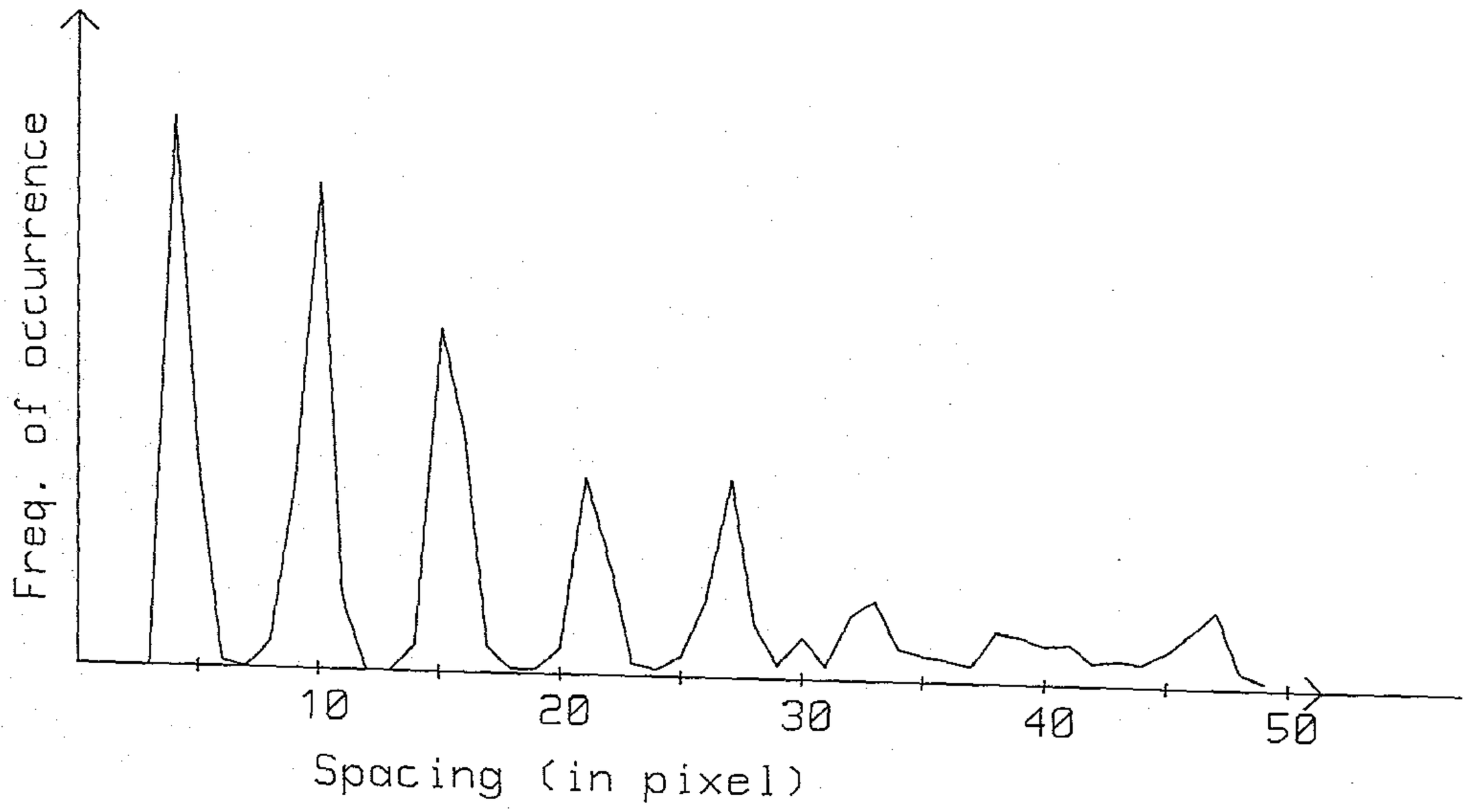


Figure 4.2: A typical ERA (for the texture D77)

Step(7): Choose N random points from the image.

Step(8): Around each point consider a subimage of size $w \times w$ and estimate the Fractal dimension. Let the Fractal dimension of the ith point be x_i.

Step(9): Compute $\sigma^2 = \frac{1}{N}\sum(x_i - \bar{x})^2$

Step(10): If $(\sigma > \varepsilon)$ $w = w + 2$; goto *Step*(7)
Otherwise, stop //sidelength $=w$ is returned//

Step(11): Compute w as a function of ℓ_0 and σ $//w = \ell_0 + \ell'$ where $\ell' \propto \sigma - \varepsilon$ //

Step(12): Choose N random points from the image.

Step(13): Around each point consider a subimage of size $w \times w$ and estimate the Fractal dimension. Let the Fractal dimension of the ith point be x_i.

Step(14): Compute $\sigma^2 = \frac{1}{N}\sum(x_i - \bar{x})^2$

Step(15): If $(\sigma < \varepsilon)$ $w = w - 2$; goto *Step*(12)
Otherwise, stop //sidelength $=w + 2$ is returned//

The above method is not satisfactory for a texture having two dominant directions (*e.g.* D16 and D17). The following modified steps should be taken in such a situation.

(i) The angle image is obtained using eq. (4.6).

(ii) For different spacing (d=2,...,100) co-occurrence matrices $P_{d\theta}(i,j)$ for all directions $(0^o, 45^o, 90^o, 135^o)$ are computed from the angle image. From each co-occurrence matrix the contrast (CON) is computed as

$$CON_{d\theta} = \sum_{i,j} P_{d\theta}(i,j) \times [\min\{|\, i-j\,|, 180 - |\, i-j\,|\}]^2$$

For each θ the contrast curve will be more or less periodic and the periodicity of each direction denotes the periodicity of the flow of the texture. Let V_θ be the periodicity in θ direction. Then $\max_\theta V_\theta$ is the size of the representative sub-image.

4.4.2 Results

To find the representative size of an image texture we considered the same set of images we used for the taxonomical classification. Among the selected textures D16 and D17 are double-directional and they are treated using the modified algorithm. For other textures ℓ_0 is chosen on the basis of their taxonomical classification. The maximum allowable error, ε in Section 4.4.1 is chosen as 0.015. Windows are chosen around 100 random positions for each texture image. Then the representative size of the texture is iteratively computed. The results of the representative size algorithm are shown in Table 4.4. The results match quite well with our visual notion about the size of the textures. For example, D75, the texture of coffee beans needs at least 4/5 beans to get the feel of the texture. Our algorithm outputs a window of size 69×69 pixels to represent D75 which approximately covers 4/5 beans. On the other hand, D77, the image of very fine grain and homogeneous texture should need very small window for representation. The output of our algorithm for D77 is only 13×13 pixels. D28 and D29 are images of the same texture (beach pebbles) with different magnification. By our algorithm, D28, the magnified version needs a window of size 61×61 pixels, whereas D29 needs 39×39 pixel window to represent the texture. D16 and D17 are herringbone weave with different magnifications. As they are double-directional textures the representative size of these textures are obtained from the periodicity of contrast curve ($CON_{d\theta}$) of the corresponding angle image. For D16 we get representative size 35×35 pixels while for D17, the texture with higher magnification our algorithm finds a representative size of 71×71 pixels. A pictorial representation is shown in Fig. 4.3.

4.5 Conclusion

A method of automatic classification of textures into three major symbolic groups is presented here. It is possible to have texture that do not clearly belong to one of the groups. Perhaps a fuzzy classification scheme can be used to take

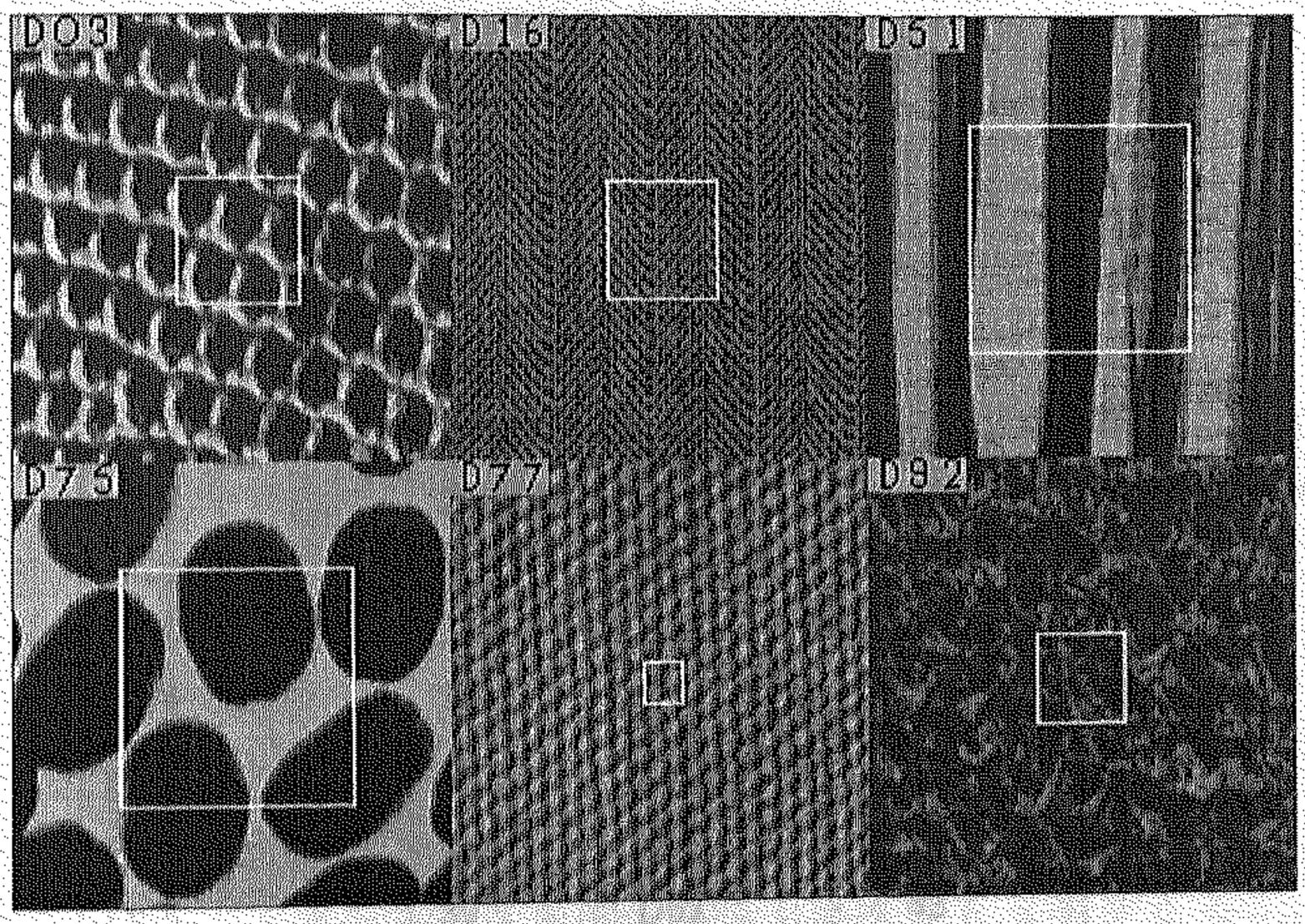


Figure 4.3 : Relative minimum representative sizes of a few textures (shown by a square with white border).

Table 4.4: Representative size of different texture fields

Plate	Rep. size(ℓ) in pixels	Plate	Rep. size(ℓ) in pixels
D03	37	D04	23
D05	37	D06	17
D09	27	D12	33
D15	57	D16	35
D17	71	D20	27
D24	25	D28	61
D29	39	D33	69
D34	25	D37	77
D51	67	D54	43
D55	25	D57	19
D65	39	D67	53
D68	49	D75	69
D76	55	D77	13
D82	25	D84	25
D92	27	D93	67
D98	63	D109	67

care of such a texture.

Once the primary classification is done in one of the three groups, more detailed symbolic description and classification may be attempted on the texture. For example, if a texture is strongly ordered, one can segment the texture primitives called texels and compute their size properties as well as their placement rules such as repetition frequency. The topic is well researched one, with several edge and region based methods being proposed to detect primitives [40,162,163,15]. Similarly, approaches based on formal grammar, nearest neighbor histogram, co-occurrence matrices and neighborhood graph [102,97,34,56] are proposed to detect the spatial structure of the primitives. Rao proposed that the textures could eventually be classified to one of the 17 wallpaper groups described in tiling theory. However, he did not propose a clear-cut algorithm for the purpose.

Secondary symbolic description of texture can depend on the application requirement. For example, in petrology it is necessary to obtain information of a grain (texel primitive) boundary as straight, curved, embayed, scaloped, lobate, dentate etc. [153]. Also, textures are named on the basis of size, shape and relative organization as seriate, prophyritic, poitilitic, ophitic, interstitial etc. [104]. Because of application dependency, we did not pursue the secondary symbolic description here.

Instead, we have utilized the taxonomical classification to find the relative size of the texture field necessary to describe the textures. Some researchers present segmentation results to show that their algorithms are powerful to detect a texture within a very small area. If this area is smaller than the minimum texture field size for the particular texture, then no conclusion about the power of the algorithm should be made.

Chapter 5

Texture based reservoir rock analysis

5.1 Introduction

Texture analysis is useful and important in many application oriented problems. In steel and similar processing industries, quality of surface is an important consideration. Surface quality can be judged by its texture. Farrokhnia [47] used multichannel Gabor filter texture features to grade the uniformity of metallic surface finish. Texture analysis has been employed in monitoring paper quality in paper *rolling mills* [31]. Conners *et al.* [35] used texture analysis techniques to detect and classify common surface defects in wood. Texture analysis is also used in inspecting semiconductor material surface [136].

Another important area of application is remote sensing where regions in satellite image are classified by *texture segmentation technique*. Recognition and segmentation of ingredients of rock sample are also done by texture analysis approaches. Changes in texture of a region or the skin of an object or living being signify important physical or biological phenomena.

In this chapter we are concerned with the analysis and gradation of reservoir rock using texture analysis. More particularly, we are interested in evaluating the potential of sandstones and limestones as petroleum reservoir rocks. We perform analysis on the images of specially prepared rock samples. The important part of the analysis is to estimate the porosity and pore types that are useful in estimating the degree of fluid (petroleum) flow.

The arrangement of this chapter is as follows. In Section 5.2 the basic features of sandstones and limestones as major reservoir rocks are discussed. In Section 5.3 sample preparation and image acquisition aspects are stated, while computer based estimation methods of the parameter are described in Section 5.4. Results are presented in Section 5.5.

5.2 Sandstone and limestone: the major reservoir rocks

The petroleum reservoir rock is a type of rock that stores (or can store) petroleum. The location of oil and gas pool within the reservoir is controlled by a number of complexly interrelated geological conditions. Generally speaking, any rock that contains connected pores may become a reservoir rock. As a matter of fact, however, nearly all the reservoir rocks are sedimentary rocks which have suffered little or no metamorphism, and most of them are sandstones, limestones and dolomites. Only in exceptional cases shales, silts and igneous rocks act as reservoir rocks. The pore spaces in the sediments are sometimes called the *reservoir space*. The original porosity of a sediment is affected by uniformity of grain size, shape of the grains, method of deposition and packing of the sediment, and compaction during and after deposition [115,39,131]. High porosity, however, is not enough for a sandstone to be a good reservoir rock. To permit the passage of oil and gas through the rock, the pores must be interconnected. In other words, the rock must be permeable [100].

5.2.1 Sandstone

Sandstone is the most important sedimentary rock type on the Earth's crust and they form roughly one-fourth of the total sedimentary rock volume. Sandstones are the sources of abrasives and act as reservoirs for natural gas, oil, and artesian water.

The framework of the normal sandstone can be described in terms of the geometric organization of the grains and their composition. Size, sorting, shape, roundness and surface texture of the framework grains and especially their packing and orientation are the most important geometric parameters. These characteristics are related to the specific hydraulic regimen that governed the deposition of the sand and to the size of the materials available to the depositing currents. These geometric parameters are described in section 5.4.

5.2.2 Limestone

In general, the term limestone is applied only to those rocks in which the carbonate fraction exceeds the non-carbonate constituents. Limestone is a common sedimentary rock. About 30% to 50% of the world's oil is produced from limestone or dolomite reservoirs.

The study of fossil content of an area is essential in the investigation of its stratigraphical history [64,48,53]. Fossils can be classified into genera, species and time groups, and can be used as a relative index of the age of the rocks in which they are found. The foraminifera are the most important group of microfossils for the petroleum geologist.

5.3 Thin section sample preparation and image acquisition

Thin-sections, as used in a geological sense, are very thin slices or fragments of rocks, fossils, or minerals mounted on glass slides for study under microscope [70]. The study of thin sections is useful in determining mineral content, texture, fabric, structure, microfossil content, void or pore space, and other aspects of a rock. Sectioning of carbonates and porous, weathered, or unconsolidated sediments or rocks require greater care and gentle handling. Initially, the sample is prepared for sectioning. A hard-rock sample should be chipped, or sawed, to obtain a thin fragment. Samples consisting of loose grains, as well as porous specimen, or fractured specimen should be impregnated. (Impregnation approach is described later on). The subsequent steps for preparing a thin-section is listed below.

1. Grind one side flat with coarse grit on a lap wheel, using water as a lubricant (kerosene or glycol for water soluble samples). Wash thoroughly to clean all coarse grit.

2. Grind on a lap with fine abrasive, or polish with fine abrasive on glass plate. Wash thoroughly to remove all abrasive and dry.

3. Mount the specimen, ground side down, on the glass slide with Lakeside 70, Canada balsam, or epoxide.

4. Grind the specimen with coarse abrasive to 1.0 mm or less thickness on a lap wheel or cut with thin-bladed saw. Grind the specimen with coarse abrasive to approximate final thickness, or to a wedge shape, on a glass plate. Wash thoroughly to remove all coarse grit.

5. Complete grinding by hand with sludge of fine abrasive on a glass plate to final thickness of 30μ or to a wedge from 30 to 500μ. Wash thoroughly and dry.

6. Stain, etch, or give other treatment as desired. Wash, clean the surface, and dry.

7. Apply a cover glass or other protective cover. Use Paramount, Epoxides, or Lakeside 70 permanent cover cement; glycerline for a temporary cover if subsequent staining, X-ray, luminescence, and so on, are desired. Clean the thin-section and attach a label or inscribe with a diamond pencil.

Impregnation or aggregation : Most of our specimens are porous and poorly cemented sandstone pieces, which generally are too weak to permit thin-sectioning without artificial bonding or impregnation of some kind. One method of impregnation is by using vacuum pump to draw out all interstitial air and allow a cold setting, highly fluid resin or epoxide to fill the pores or fractures and then harden. For successful impregnation, a sample must be dry. Equal parts of Araldite AY105 and Hardener 935P dissolved in three parts of toluene with subsequent heating for 2 hours at 90°C can be used for impregnation. There are many other binding material and hardener for impregnation. Ireland [70] developed a method of impregnation with Canada balsam that is inexpensive and quick.

Staining of carbonate rocks : For the carbonate rock one needs staining of thin-section before analysis. Techniques of staining is briefly discussed by Dickson [42].

Image acquisition : The microphotographs required for our study were obtained from two different sources.

1. We collected oil bearing sandstone samples of Assam (North-East Indian state). The samples are first impregnated using Canada Balsam and thin sections are prepared using the above method. Microphotographs of the samples are taken by a microscope and camera combination. Microphotographs were taken with smaller magnification (30× to 45×) for grain size and shape analysis. Higher magnification (80× to 160×) was used for porosity analysis.

The microphotographs of lower magnification are taken using polarized light, because for this experiment we need the quartz grains to be separated from each other. The mean gray value of an individual grain under polarized light depends on the angle between its c-axis and the direction of polarization. Most of the neighboring quartz grains do not have the same c-axis and this property helps us to extract individual quartz grain for shape and size analysis.

As we mentioned earlier, the sandstone is made up of framework and void. Void is slowly filled up by cement. The cement and void are not separable under polarized light with small magnification. We captured microphotographs with higher magnification (80× to 160×) under plane light to avoid the above shortcomings of porosity analysis.

2. Another set of microphotographs are taken from the album of Adams *et al.* [1]. All the samples of limestone reservoir rock are obtained from this source. Some of the sandstone samples are also collected from this album.

The microphotographs were gray level digitized using a flat-bed scanner under uniform illumination. The scanner resolution varied from 83 dpi to 112 dpi. All the images are of size 256 × 256 pixels and have 256 gray levels.

5.4 Analysis and measurement

Porosity and permeability are the most important parameters of sandstone reservoir rocks. On the other hand, bioclast separation and identification as well as estimation of porosity are the major issues of limestone reservoir rock. We adopted the texture segmentation technique to estimate porosity in sandstone and to separate bioclast in limestone samples. Permeability is estimated by using the formula due to Pettijohn [131]. To apply the formula we need to compute the grain size and sorting of the samples. Sphericity and roundness of the grains are two param-

eters which have direct or indirect effect on porosity and permeability of sandstone although no analytical relation exists to relate them.

5.4.1 Region Segmentation Technique

Texture segmentation approach is used to find porosity in sandstone samples and to separate the bioclasts in limestone samples. The segmentation is completed in three steps. For segmentation of sandstone samples we considered the image with higher magnification taken under plane light.

Feature selection: The features for the texture segmentation are based on fractal and multifractal dimension. Since the directionality plays only a minor role in the textures we deal with, we did not consider the features derived from horizontally and vertically smoothed image. Thus the considered features are.

(a) FD of the original image.

(b) FD of the high gray-valued image.

(c) FD of the low gray-valued image.

(d) Multifractal dimension of order 2. The partition function used for this purpose is

$$\chi(q,r) = \sum_{i,j} [n'_r(i,j)]^q \simeq r^{\tau(q)}$$

where r is described in Sec 2.3 and $n'_r(i,j) = n_r(i,j)/N_r$, where $n_r(i,j)$ and N_r are given by eq (2.7) and eq (2.8), respectively.

The method of estimation of the features are described in Section 3.3. All features are computed on overlapping windows of size 17×17.

Preprocessing : We mentioned earlier in Section 3.4 that if the features stated above are directly used for segmentation, considerable mis-classification can occur in the interior of a region and border between two regions. A feature domain smoothing can reduce the mis-classification to a large extent. We used EPNSQ [74] for feature domain smoothing. Window of size 7×7 pixels is chosen in EPNSQ approach.

Segmentation : The unsupervised segmentation technique described in Section 3.5 is adopted for our purpose. Here too the segmentation is done in two stages. In the first stage, a number of initial seeds are computed from the feature images. Each seed point generates a separate cluster. These clusters are then iteratively reduced to actual clusters by redistributing the data of smallest cluster among the other clusters. In the second stage, the K-means algorithm is adopted to obtain the desired clustering.

5.4.2 Porosity estimation

The porosity P of a rock is defined as the ratio of its pore volume V_p and the total volume V_b expressed in percent. In other words

$$P = \frac{V_p}{V_b} \times 100$$

5.4.3 Grain size, shape and permeability analysis

At this stage grain size, sorting, sphericity and roundness are computed. Of these parameters, the effects of size and uniformity have been studied experimentally. Researchers have found that in most cases the following relation holds good:

$$K = kd^2/\sigma \tag{5.1}$$

where K is the permeability, k is the proportionality constant, d is the average diameter (the Geometric mean) of the grain and σ is the standard deviation of the grain size in logarithmic scale.

For computation of the parameters of eq(5.1) separation of individual grains are done on the images of microphotographs with lower magnification captured under polarized light. To separate each grain, at first we used EPNSQ smoothing with 5×5 pixels window to smooth interior of the regions representing quartz particles. Then a simple histogram thresholding is done to separate quartz

particles from the background. The pixels having gray value above the threshold is assigned to 1 (representing quartz particles) and the rest are assigned to 0 (background). The connected components of 1-labeled pixels represent the quartz particles.

The grain size, sorting, sphericity and roundness measures are computed after the individual quartz particles are identified.

Grain size analysis : Grain size analysis determined by grain measurement, microscopic measurement in thin section, sieving, or sedimentation methods yield raw data on percent of grains in a sample that occur in each size class [110]. The mean grain-size of sandstone lies in the range of 1/16 mm to 2 mm. Krumbein [91] introduced the 'Phi'(ϕ) scale as log transformation of millimeters in order to simplify computation of statistical parameters (*i.e.*, if mean diameter of particle is x mm then it is expressed as $-\log_2 x$ in 'Phi' measurement). The mean grain-size and sorting have indirect effect on the permeability of a reservoir rock [131]. Coarser grained and well sorted sandstones are the most permeable among the reservoir rocks.

There are several numerical and graphical methods of finding the mean grain-size and sorting. The Mean (M) and sorting (S) are numerically calculated as

$$M = \frac{1}{n}\sum_{l=1}^{n} d_l \tag{5.2}$$

$$S = \frac{1}{n}\sqrt{\sum (d_l - M)^2} \tag{5.3}$$

where d_l is the diameter of the l th element in 'Phi' scale.

The diameter of a grain (in 'Phi') is calculated as

$$d_l = \log_2 m - \frac{1}{2}\log_2 A_l$$

where m is the magnification and A_l is the area of the l th grain. The mean grain size (M) can be computed from (5.2) and the uniformity (or sorting) of grains (S) can be computed using (5.3). Although the permeability depends on shape and

packing of the grains, an initial estimate can be found from eq (5.1) as

$$K = k2^{2M}/S \tag{5.4}$$

Sphericity and roundness: Analysis of grain morphology is generally restricted to form description of detrital particles [135]. Various qualitative attributes are used to describe grain shape, *e.g.*, round, disc-like, needle-like, fusiform etc. Such qualitative attributes vary from one observer to another. However, sphericity and roughness are the two parameters popularly used for shape analysis. The sphericity attempts to quantify the overall spherical shape of a grain while roundness describes the relative sharpness of grain corners and edges.

There exist various methods of computing sphericity. For measurement of grains in thin-section and grain projection, where only 2D data are available Riley [141] proposed a projection sphericity, ψ_r

$$\psi_r = \frac{d_i}{D_c}$$

where D_c = diameter of smallest circumscribing circle and d_i is the largest inscribing circle of the grain.

At first, we locate co-ordinates of the boundary points of the given grain. Let S be the set of boundary points.

Estimation of D_c : Consider a point $P_k(x_k, y_k)$ in the object and find its distance from the boundary points. Let D_k be the maximum of these distances. Find D_k for all points within the object. The point for which D_k is minimum is chosen as the center of the minimum circumscribing circle of the object. In other words,

$$D_c^2 = \min_k \max_i \left[(x_i - x_k)^2 + (y_i - y_k)^2\right]$$

Estimation of d_i : This problem is dual to the previous problem. Consider a point $P_k(x_k, y_k)$ in the object and find its distance from the boundary points. Let D_k be the minimum of these distances. Find D_k for all points within the object.

The point for which D_k is maximum is chosen as the center of the maximum inscribing circle of the object. In other words,

$$d_i^2 = \max_{k, P_k \in \text{Hull(S)}} \left[\min_i (x_i - x_k)^2 + (y_i - y_k)^2 \right]$$

The sphericity of the grain is calculated as $\psi_r = d_i / D_c$.

Roundness

Grain roundness is defined as the relative sharpness of grain corners, or more simply as description of grain surface curvature. Actually it is a measure of degree of wriggleness of an object boundary. The most widely used measure of roundness P_r due to Wadell [166] is

$$P_r = \frac{\sum (r_m / R)}{N} \tag{5.5}$$

where r_m is the radius of curvature of individual corners, R is the radius of maximum inscribing circle and N is the number of corners including the corners having zero radius of curvature.

Here we detect the corners and estimate the radii of curvature of the corners. The radii of curvature can be estimated by the method described by Pavlidis [124]. Let $P_i, i = 1, \ldots, m$ denote the sequence of pels in a curve. Let $A_1, A_2, \ldots, A_n$ denote the n-pel neighborhood of a particular pel P_i in the original curve ($P_i = A_{\lfloor n/2 \rfloor - 1}$). The radius of curvature at P_i is estimated as follows :

Denote the points A_1, P_i and A_n as X, Y and Z, respectively. The angle between XY and YZ are estimated as $\phi = \tan^{-1}\left[\frac{m_1 - m_2}{1 + m_1 * m_2}\right]$, where m_1 and m_2 are the slope of the lines XY and YZ, respectively (Corners of angle $\pi/2$ to be checked beforehand). The corners are detected as the points having $\phi < \pi$ and the radius of curvature of a corner is calculated as $r = \frac{|XY|}{\cos(\phi/2)}$. The roundness P_r is estimated using the eq (5.5) after radii of curvature of all the corners are computed.

5.5 Results and Discussion

Sandstone : It has been already mentioned that the two major criteria for an oil bearing sandstone is its porosity and permeability. They are examined in two stages. *In the first stage* we estimate the porosity of a given sample of sandstone using the texture segmentation technique described in Chapter 3. We discard the samples having low porosity ($< 10\%$) for the second stage. In stage two we estimate the permeability *using eq* (5.4). *For Grain size and shape analysis*, we normally consider a portion of the image. The smallest size of such a portion is computed by the method discussed in Section 4.4.

Among several samples examined, we present here the results for those having some oil bearing potential. *Some samples have no pores and hence cannot deposite oil.* Some porous samples are shown in Fig 5.1(a), 5.2(a), 5.3(a) and 5.4(a). The corresponding segmentation mappings are shown is Fig 5.1(b), 5.2(b), 5.3(b) and 5.4(b). The sample corresponding to Fig 5.1(a) *has the lowest porosity* and Fig 5.4(a) has the highest porosity. The qauntitative porosity measure is shown in Table 5.1. For grain size analysis we considered the same set of samples but images are captured at lower magnification. Images corresponding to Fig 5.1(c), 5.2(c) and 5.3(c) are lower magnified version of Fig 5.1(a), 5.2(a), and 5.3(a), respectively. Image corrsponding to Fig 5.4(a) is considered for grain size analysis also. The results of grain size analysis is shown in Table 5.2

Table 5.1: Porosity of sandstone samples.

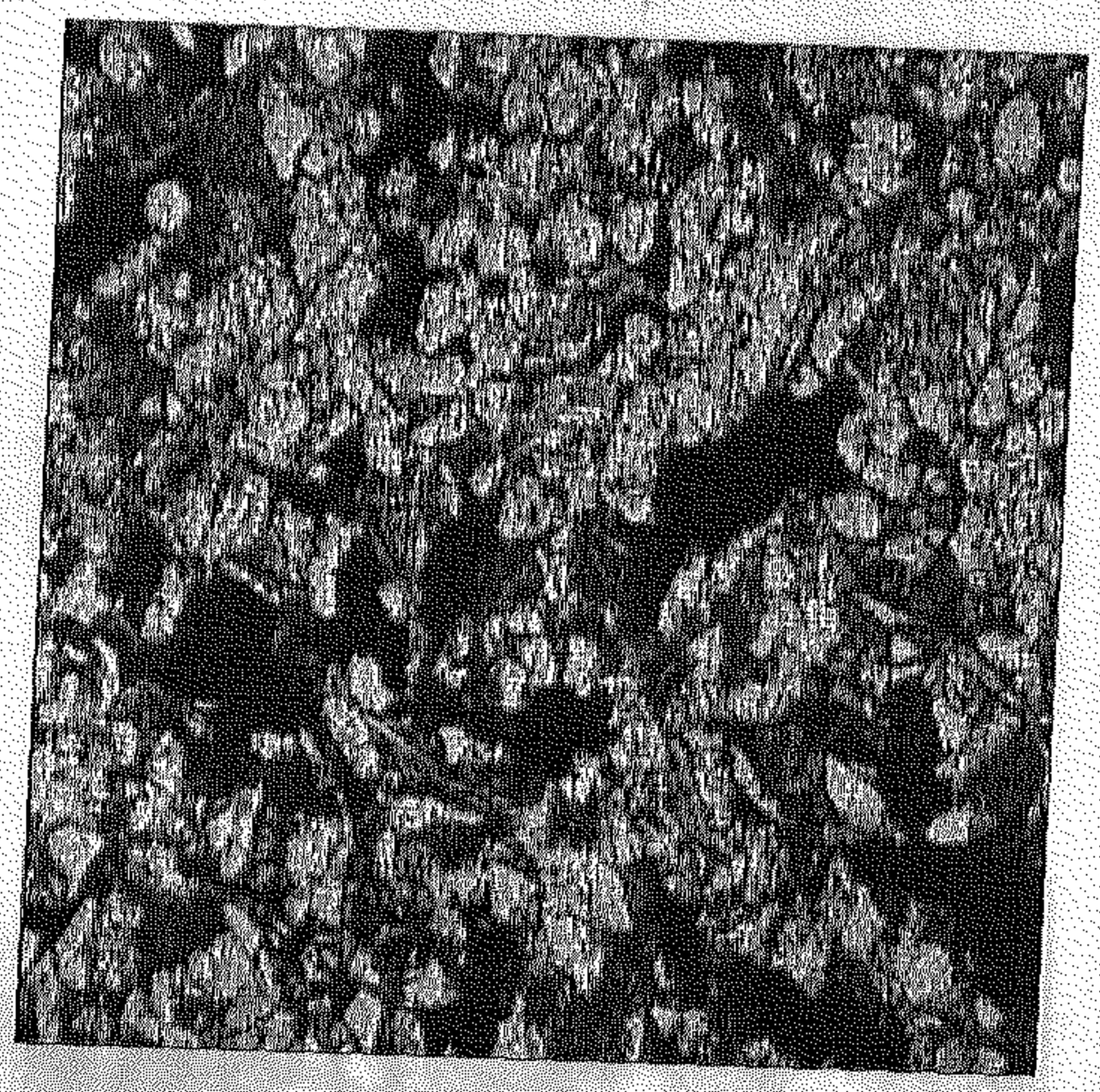
Image	Porosity in %
Fig 5.1	7.91
Fig 5.2	25.19
Fig 5.3	20.56
Fig 5.4	37.00

Table 5.2: Results of grain size and shape analysis.

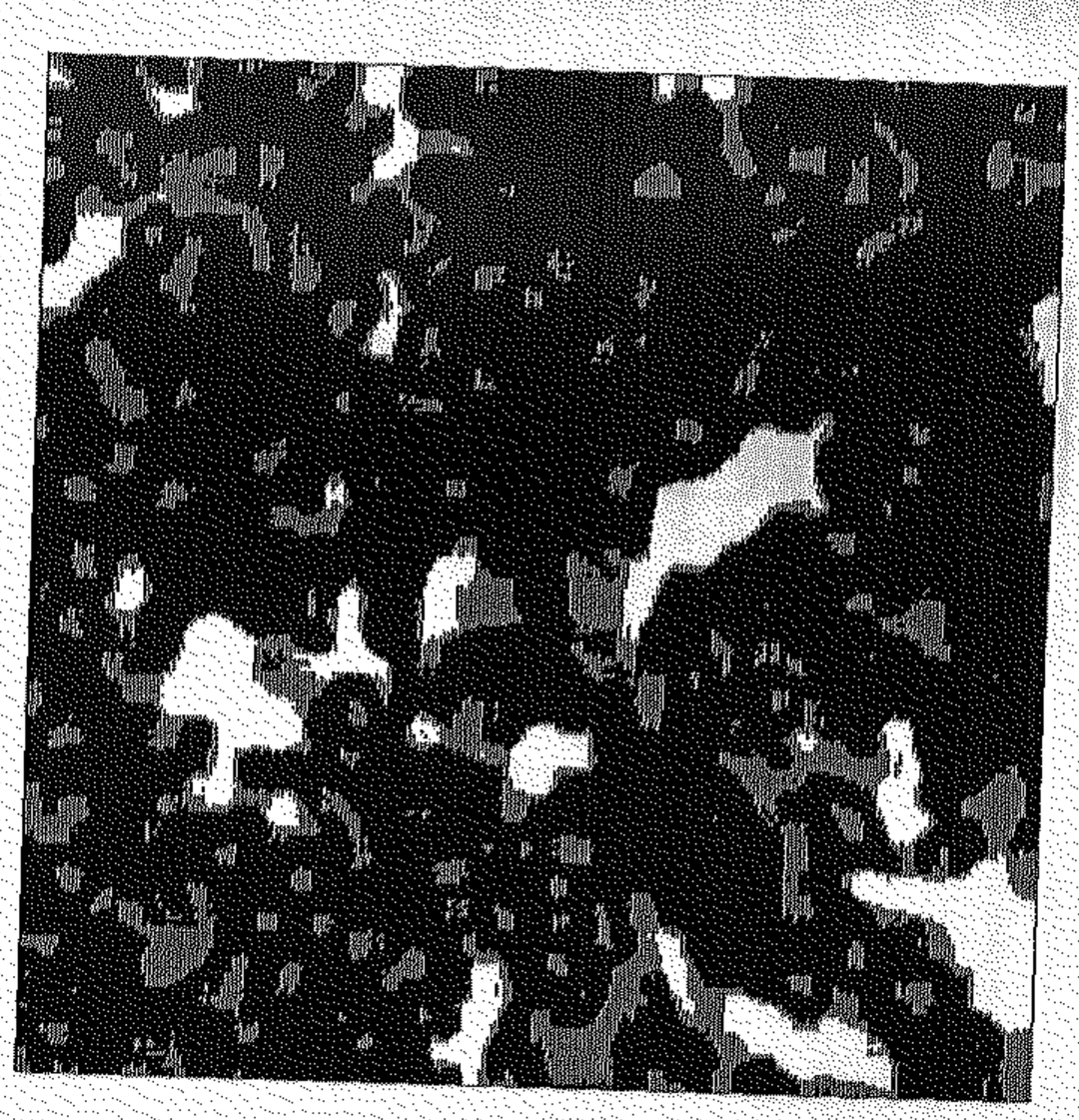
Image	Mean	Sort	Sphericity	Roundness
Fig 5.1	6.01	1.77	0.52	0.60
Fig 5.2	6.41	2.06	0.40	0.63
Fig 5.3	7.10	1.83	0.46	0.61
Fig 5.4	7.92	1.13	0.52	0.50

Limestone : For analysis of limestone we considered the images from the album by Adams *et al.* [1]. The analysis of limestone also consists of two stages. In the first stage we segment the images using the method discussed in Chapter 3. Here we took only the first three features. In this stage we separate bioclasts from the background of a given sample. The images are shown in Fig. 5.5(a) – Fig. 5.10(a). The segmentation mapping of these images are shown in Fig. 5.5(b) – Fig. 5.10(b), respectively.

In the second stage we identify the bioclast present in the sample. We are mostly interested in the presence of foraminifera. Foraminifera shows some preferred orientation in its structure, whereas other bioclasts such as bivalve, algae or brachiopod do not have any preferred orientation or line-like structure in them. Identification of foraminifera is done by orientation gradation technique described in Section 4.2. The samples corresponding to Fig. 5.7(a), Fig. 5.9(a), and Fig. 5.10(a) are identified as samples of foraminifera and they need further study for approximating their depositional environments.

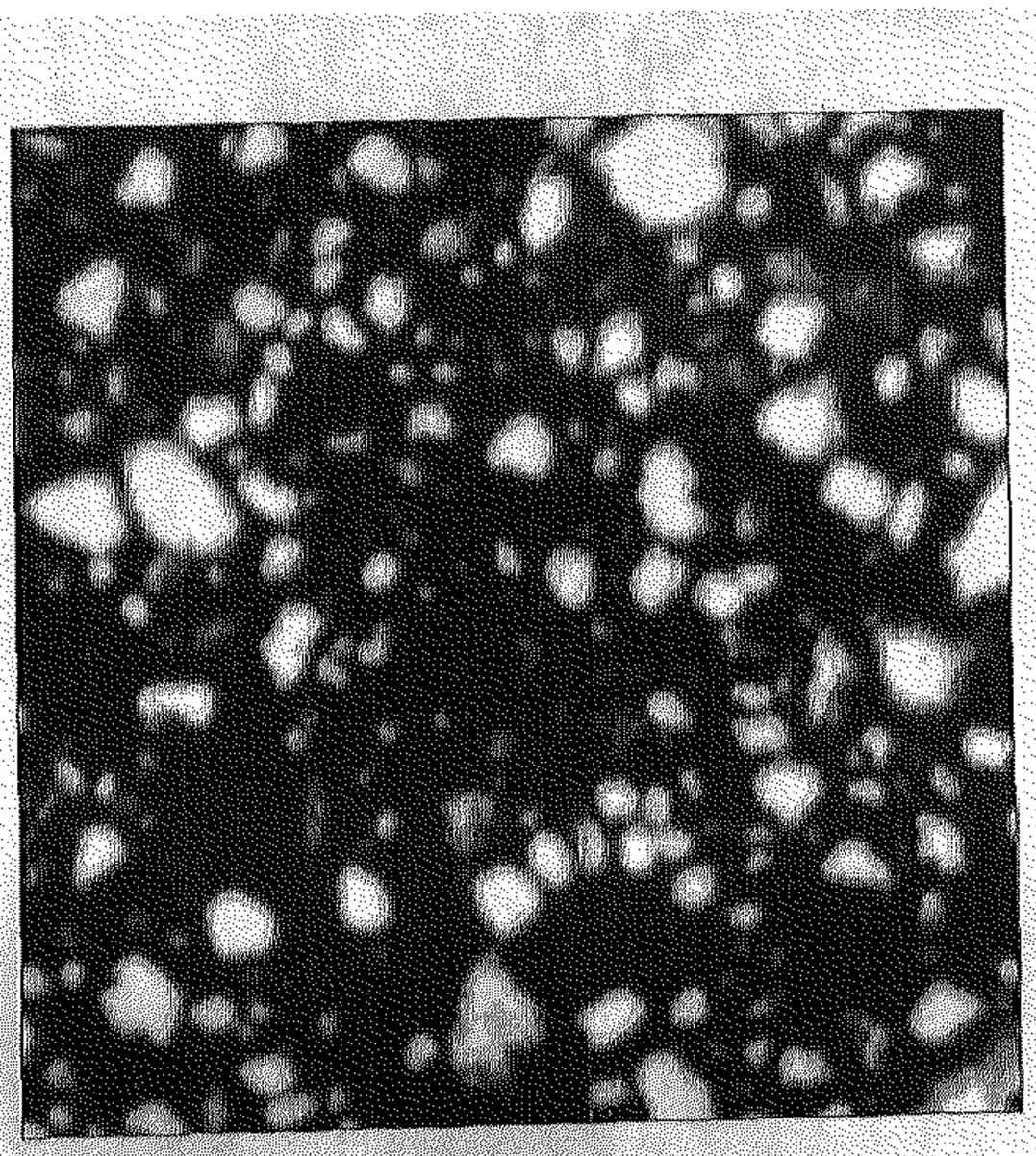


(a)



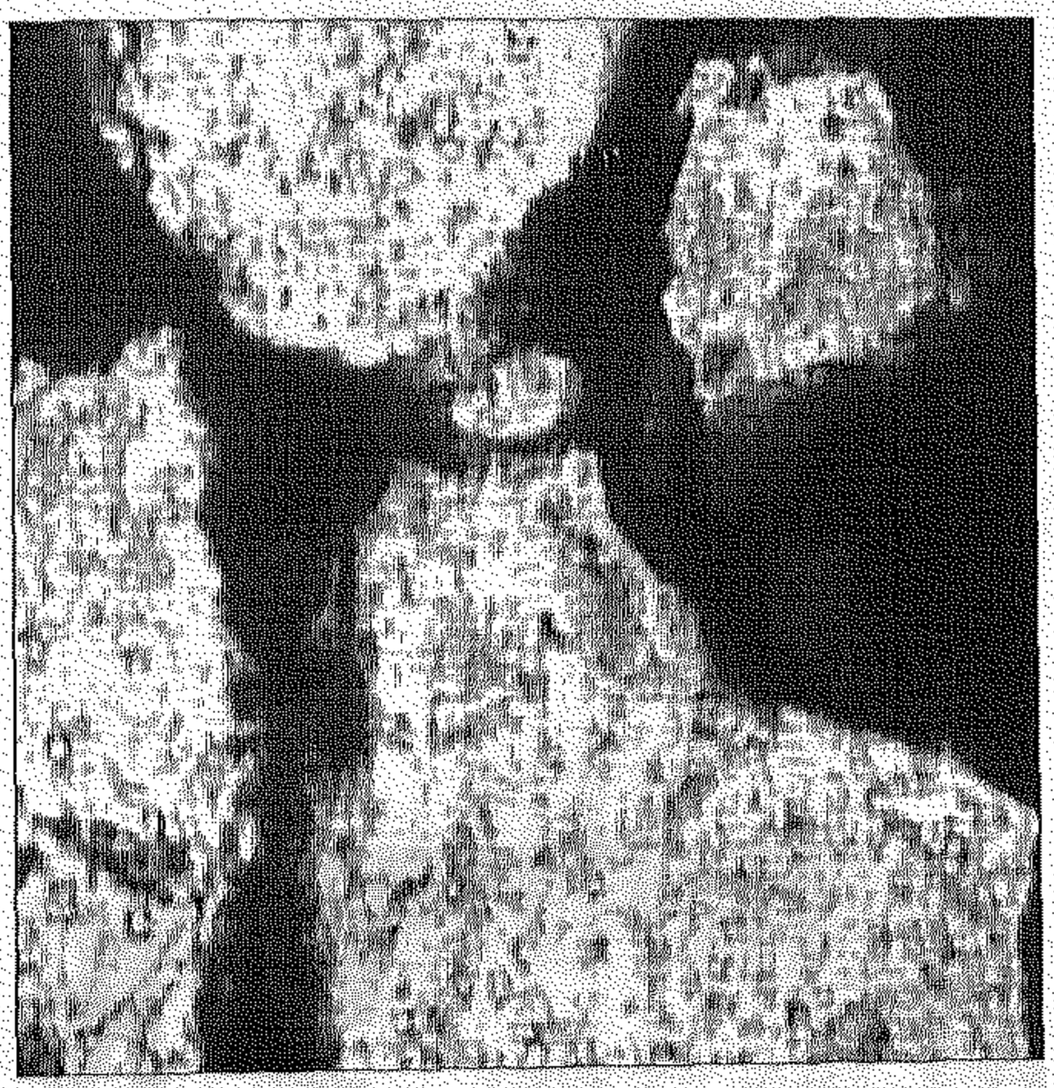
(b)

Figure 5.1 (a) Fine grained sandstone micrograph image (at 80×).
(b) Segmentation mapping of (a). (c) Same micrograph image at 32×.

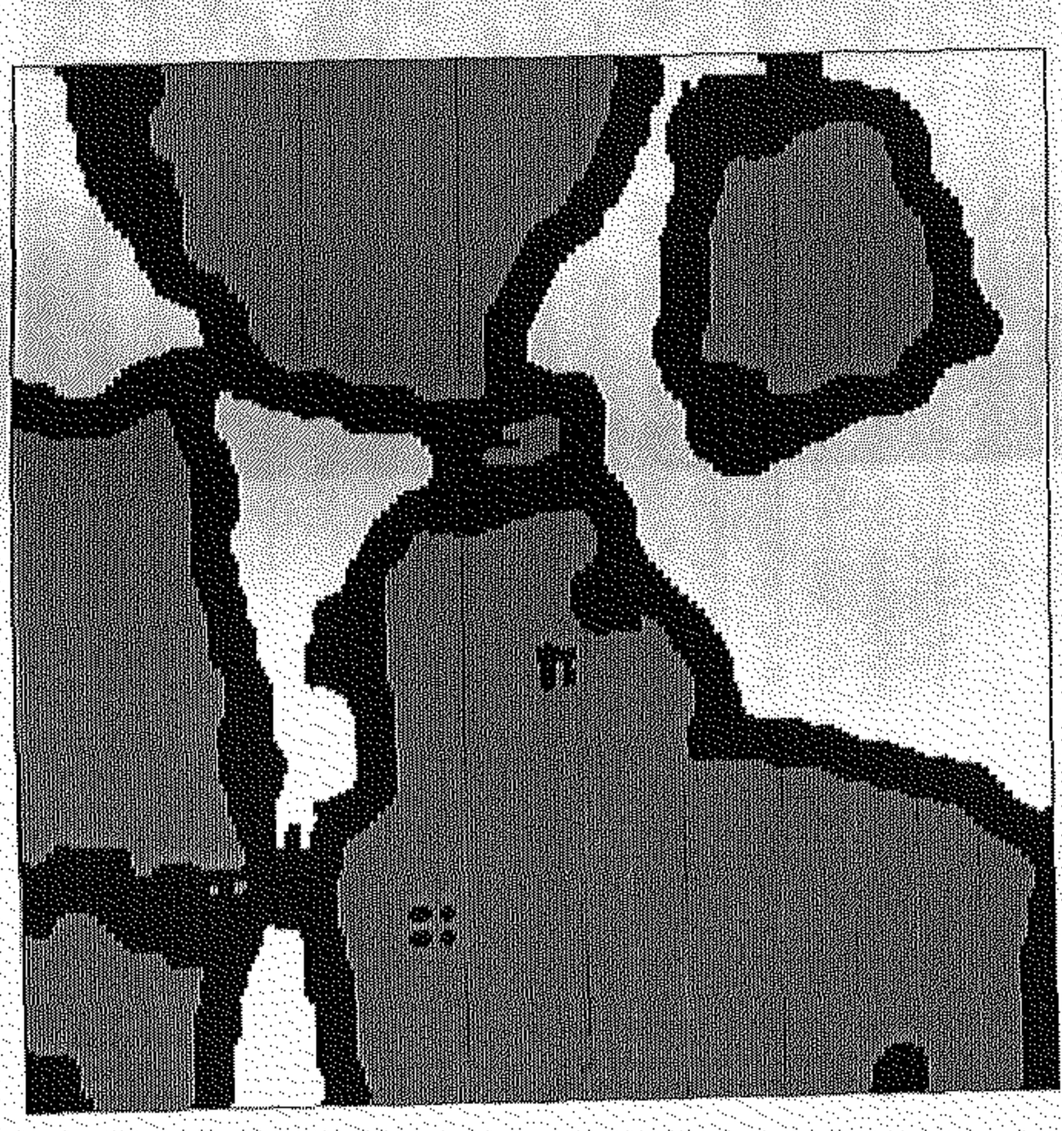


(c)

Figure 5.1 : *Contd.*

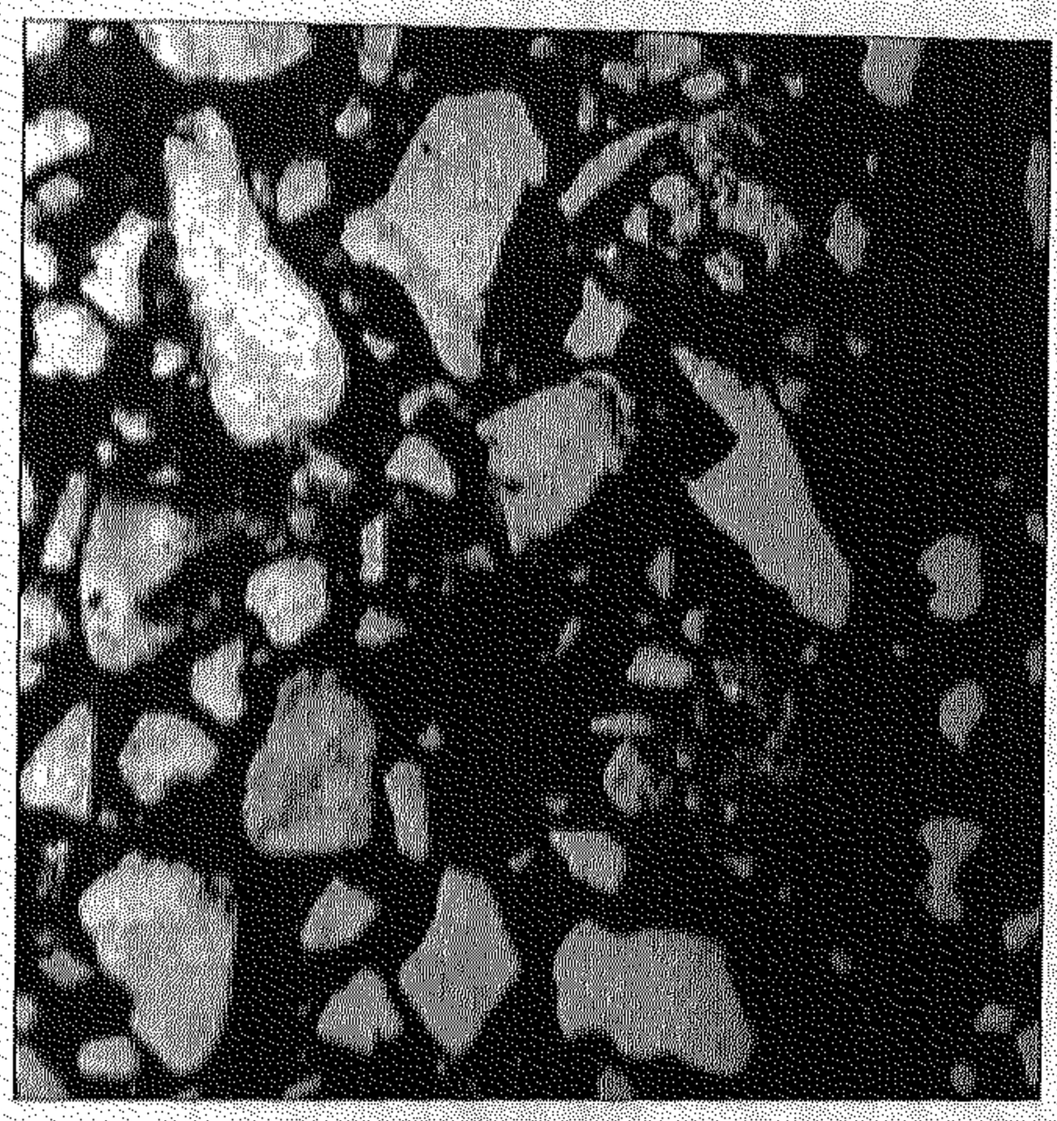


(a)



(b)

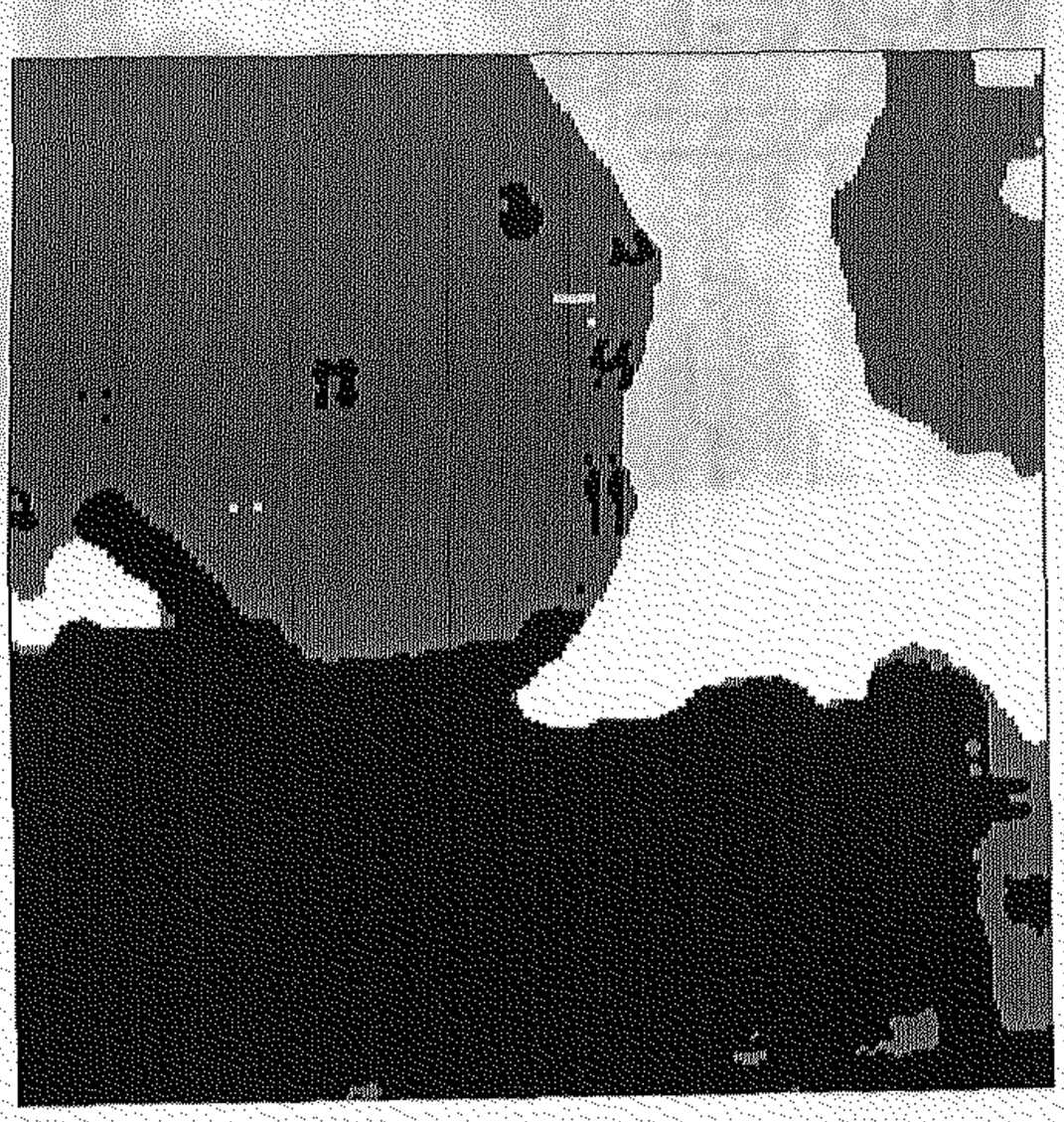
Figure 5.2 (a) Fine grained sandstone micrograph image (at 160×).
(b) Segmentation mapping of (a). (c) Same micrograph image at 32×.



(c)

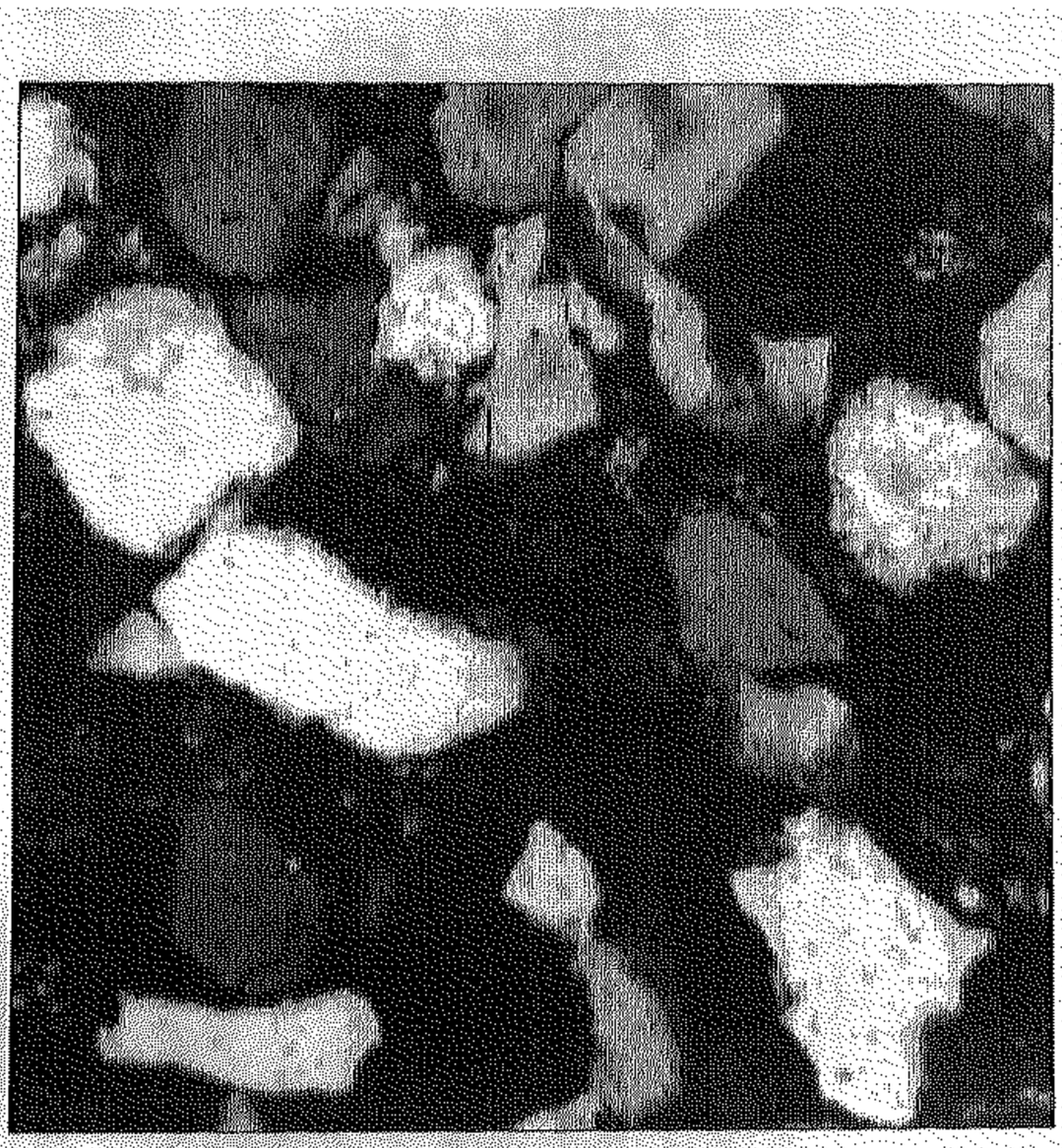
Figure 5.2 : *Contd.*

(a)



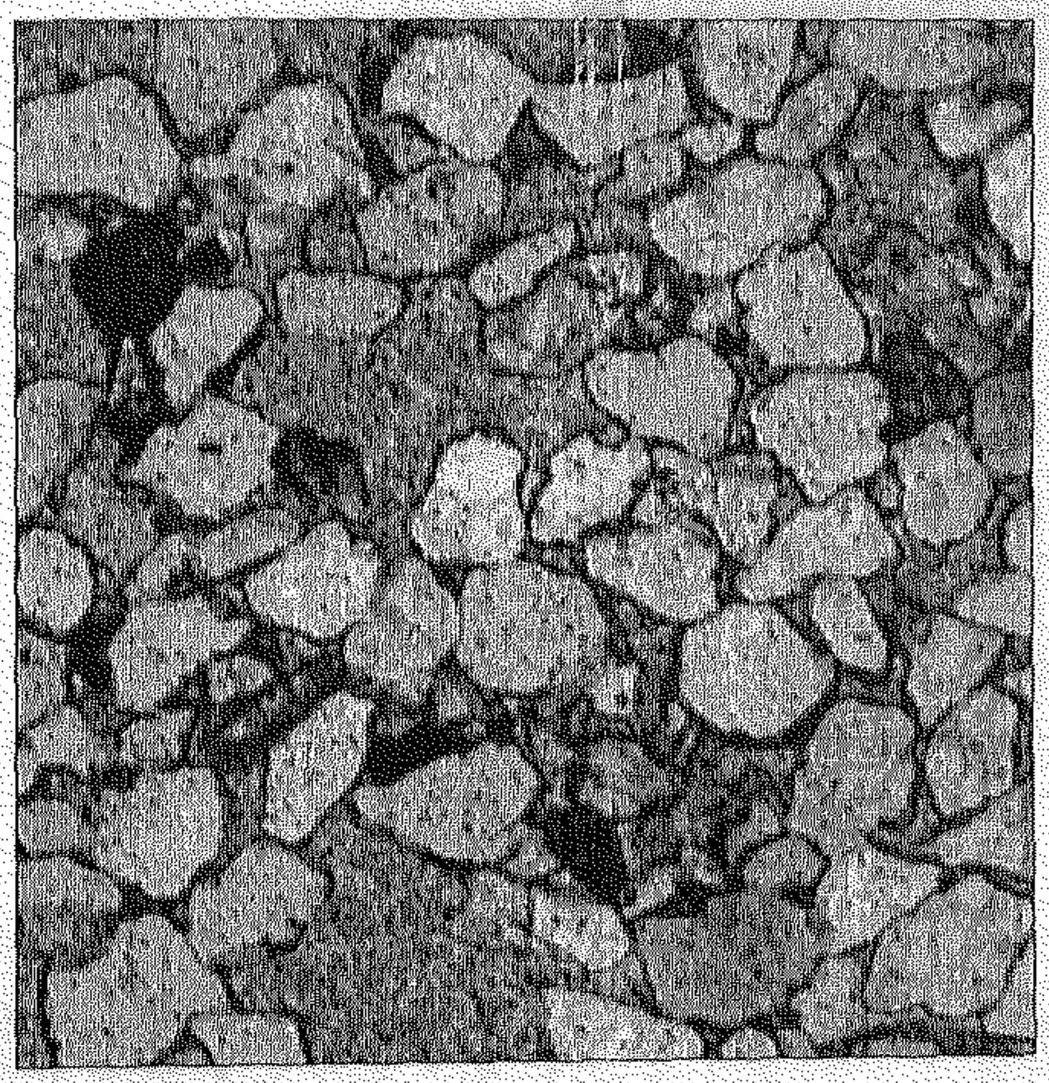
(b)

Figure 5.3 (a) Medium grained sandstone micrograph image (at 160×). (b) Segmentation mapping of (a). (c) Same micrograph image at 40×.



(c)

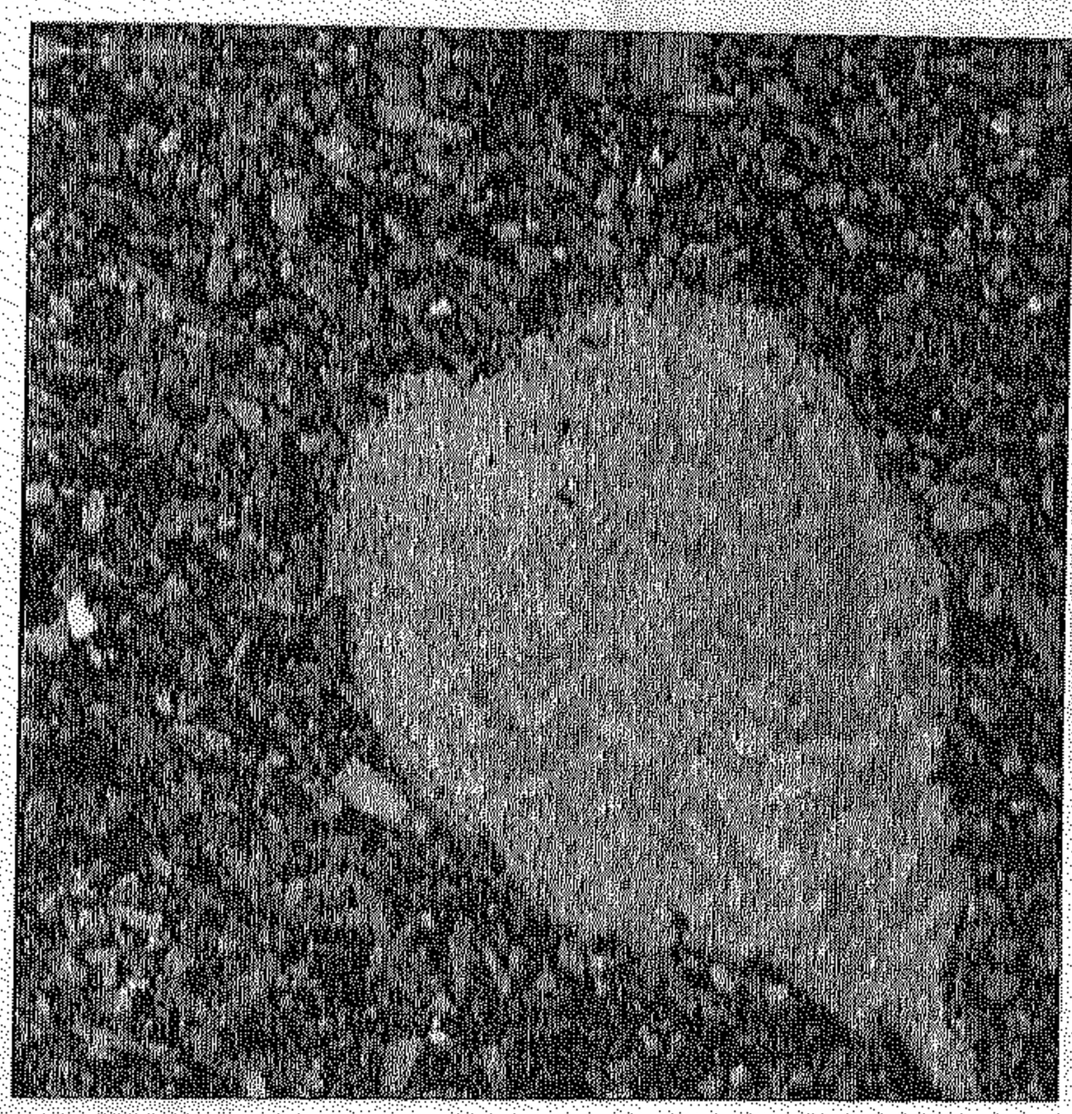
Figure 5.3 : *Contd.*



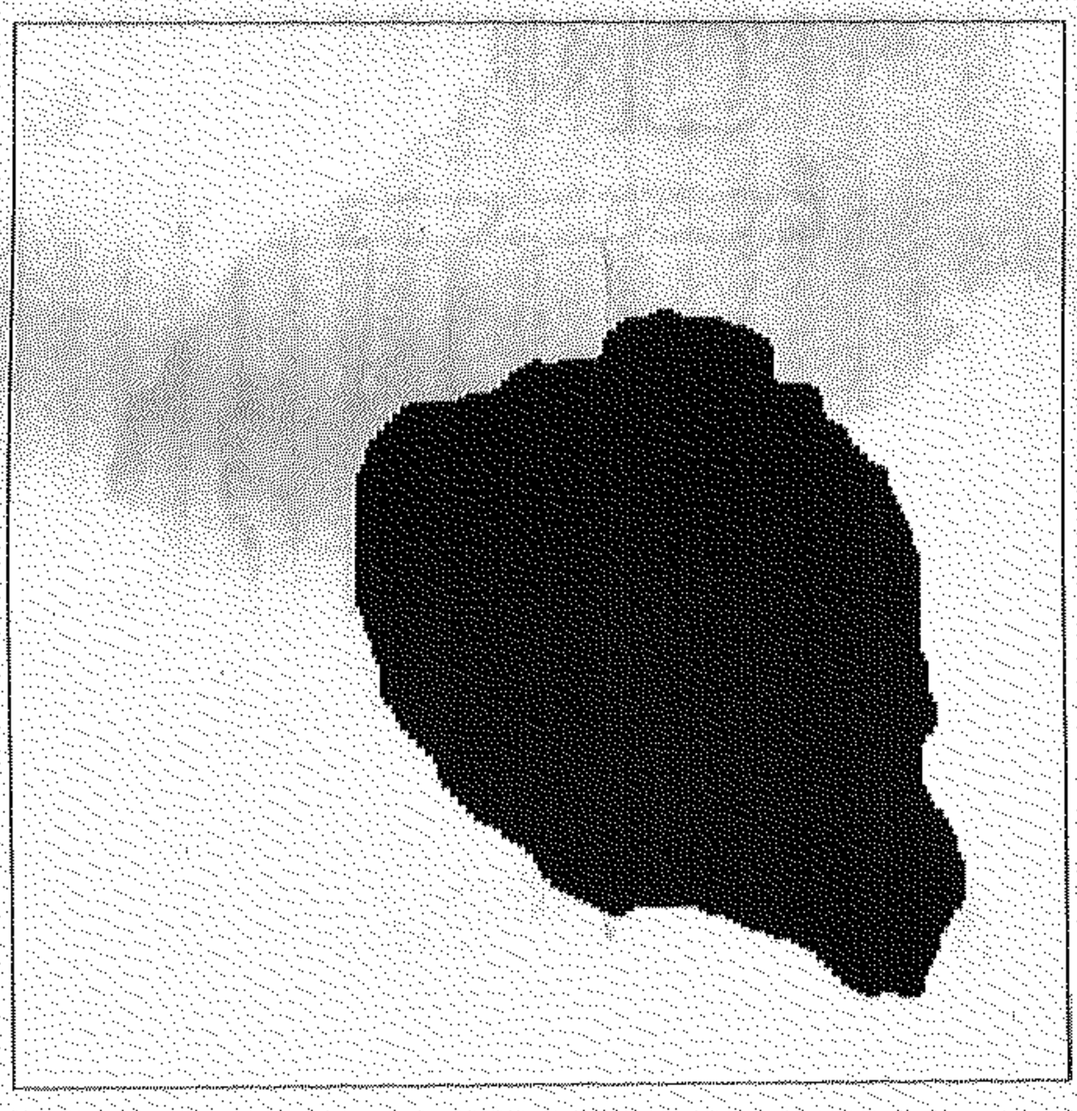
(a)

(b)

Figure 5.4 (a) Coarse grained sandstone micrograph image (at 43×).
(b) Segmentation mapping of (a).

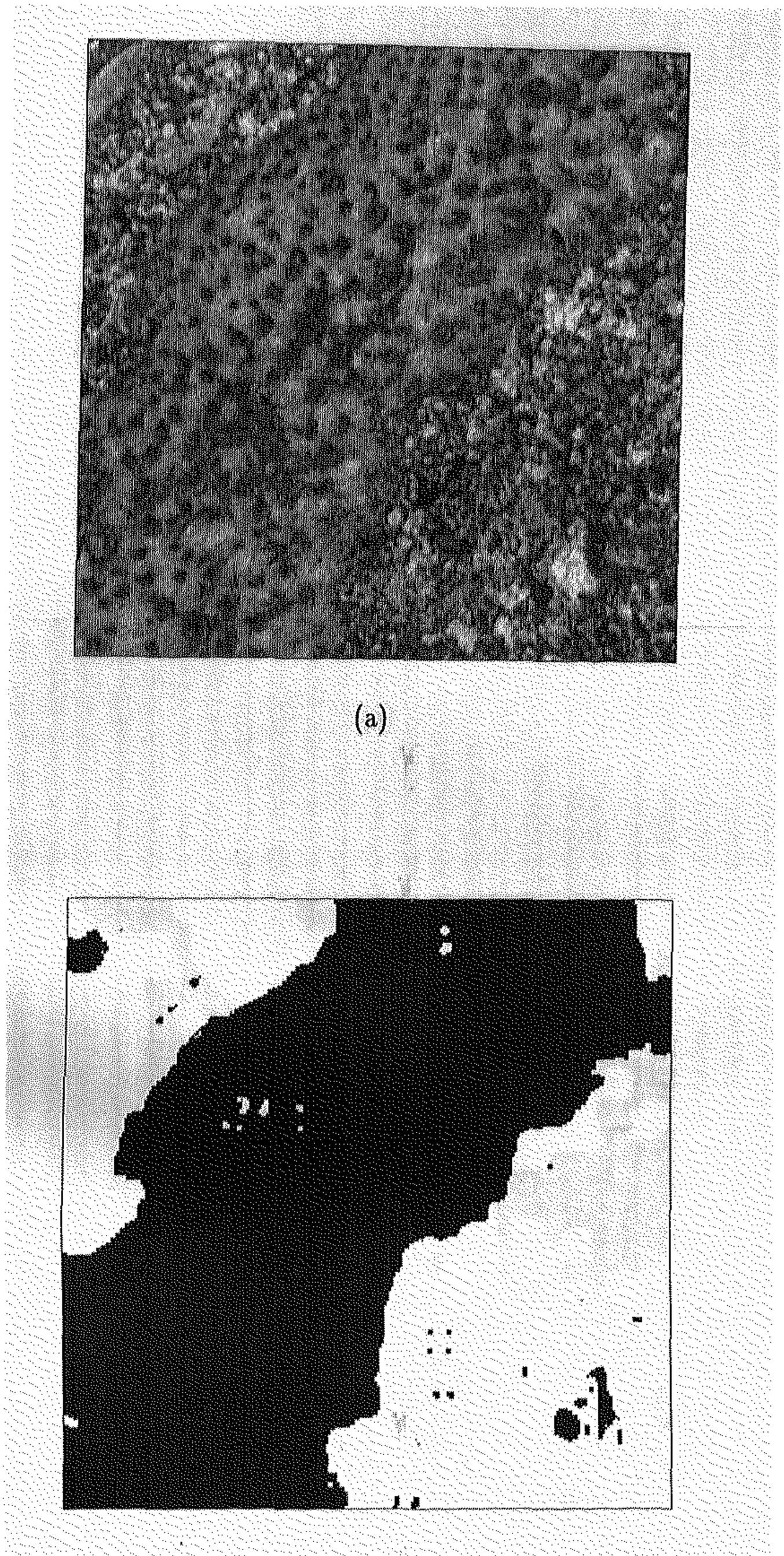


(a)



(b)

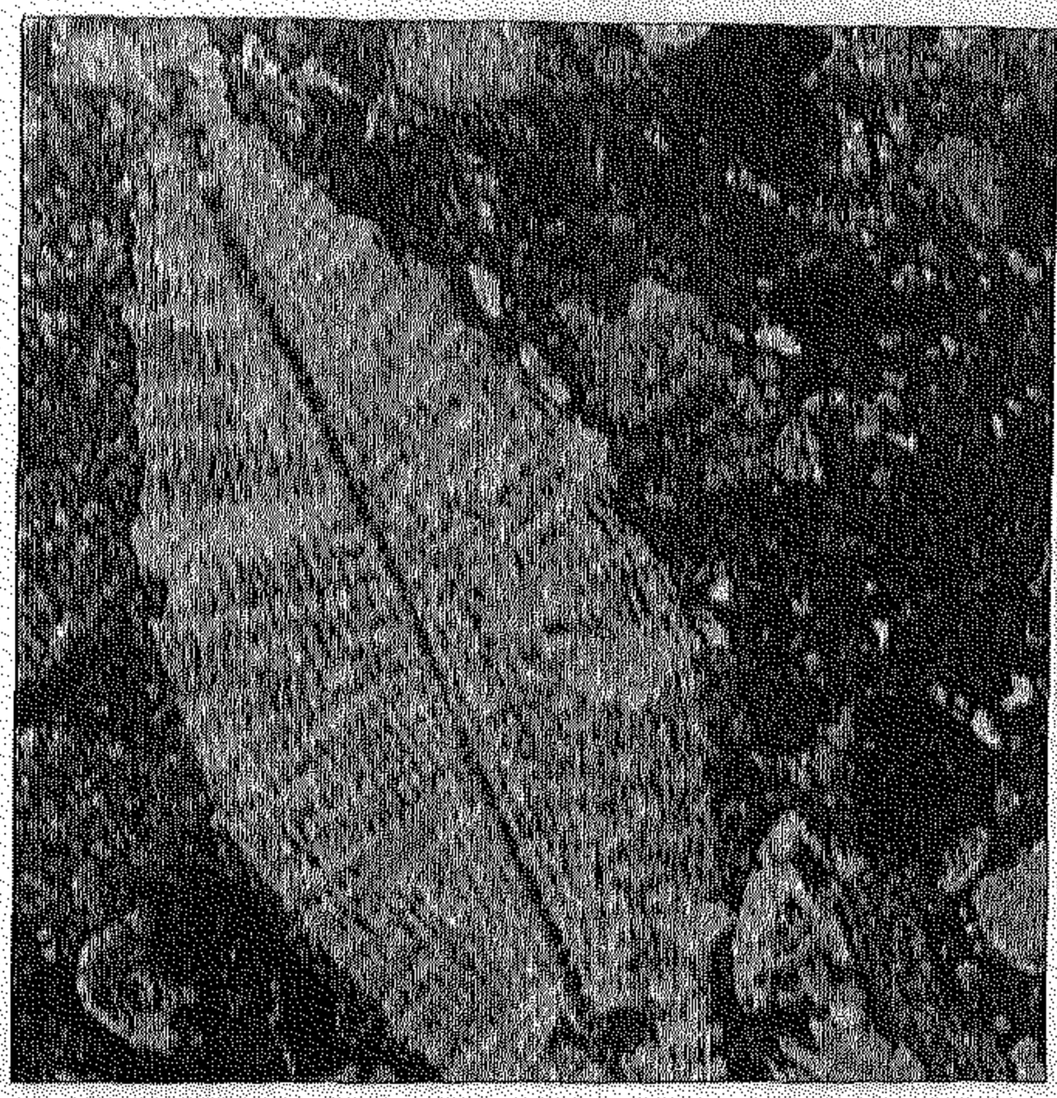
Figure 5.5: (a) Limestone micrograph image containing bivalve (at 14×). (b) Segmentation mapping of (a).



(a)

(b)

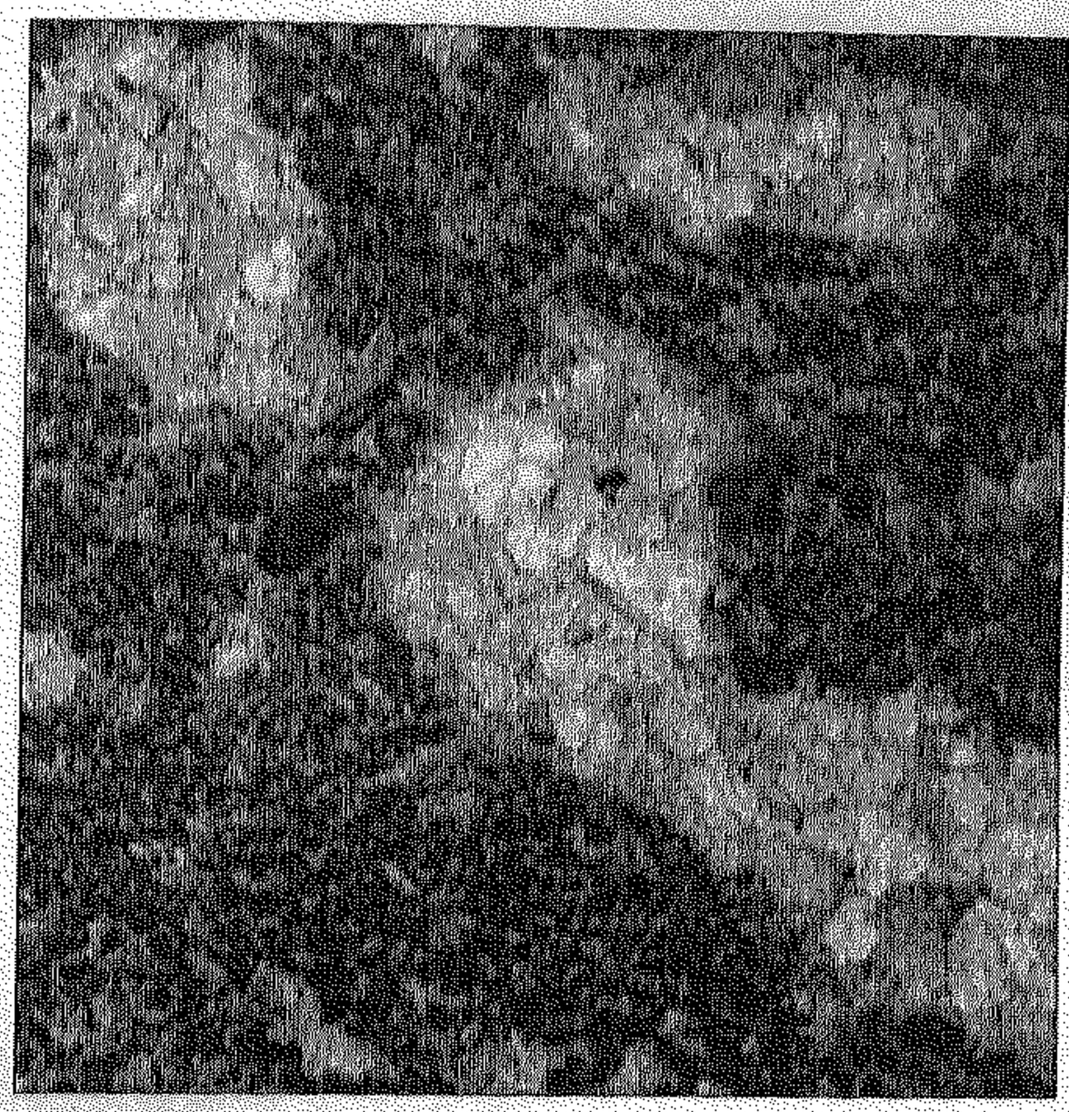
ure 5.6: (a) Limestone micrograph image containing chiopod (at 32×). (b) Segmentation mapping of (a).



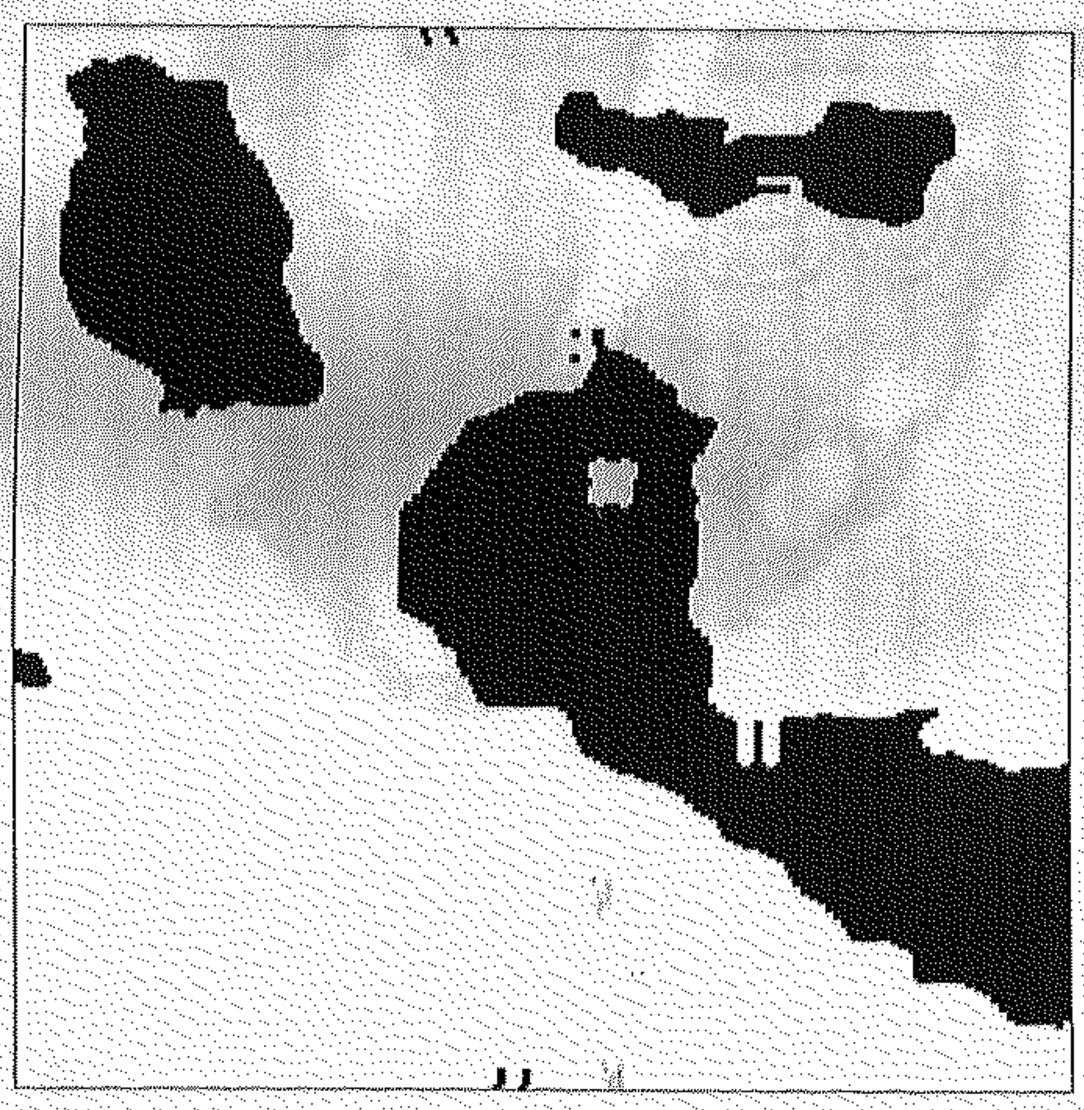
(a)

(b)

Figure 5.7: (a) Limestone micrograph image containing foraminifera (at 32×). (b) Segmentation mapping of (a).



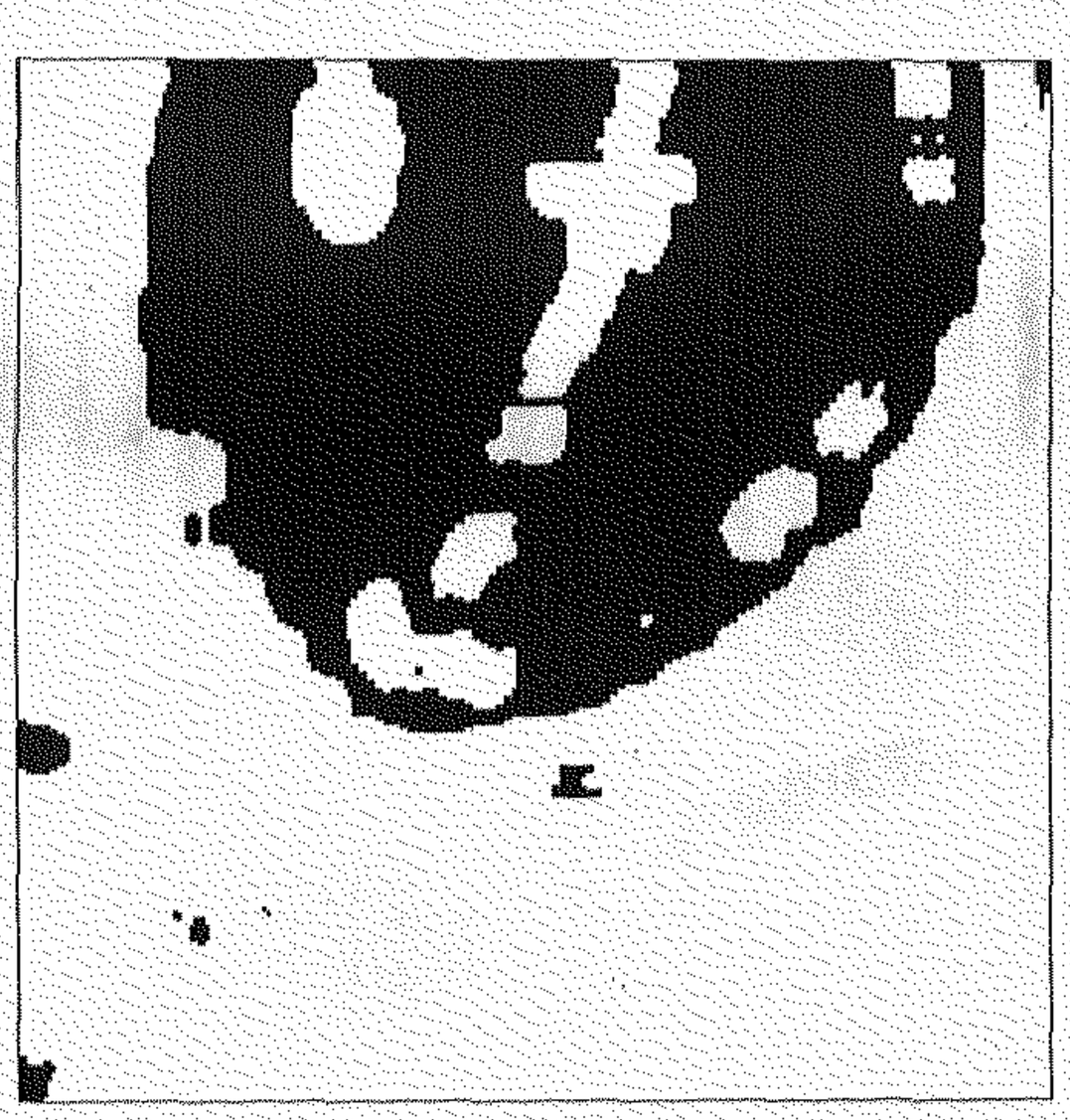
(a)



(b)

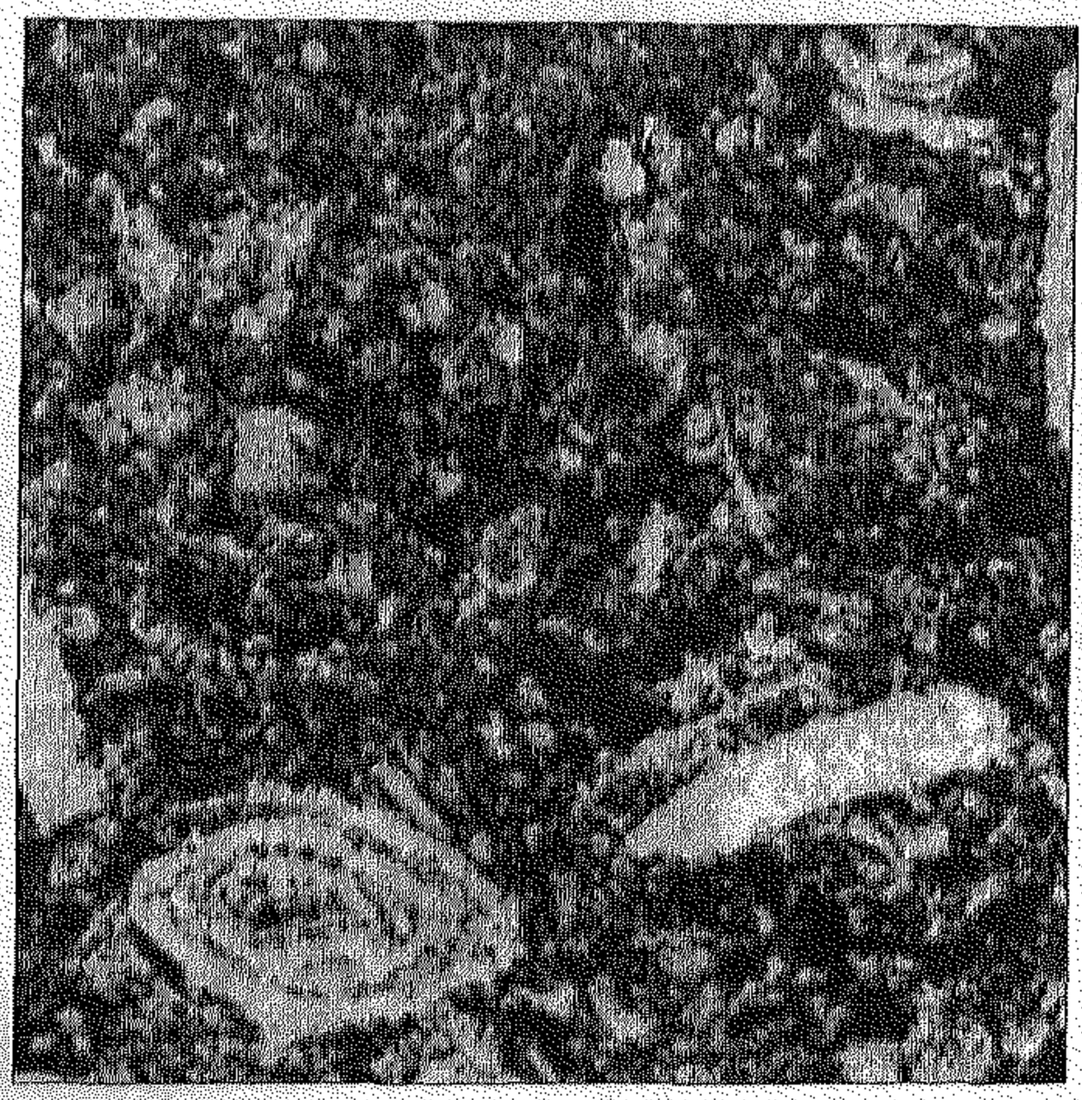
Figure 5.8: (a) Limestone micrograph image containing algal stromatolites (at 14×). (b) Segmentation mapping of (a).

(a)

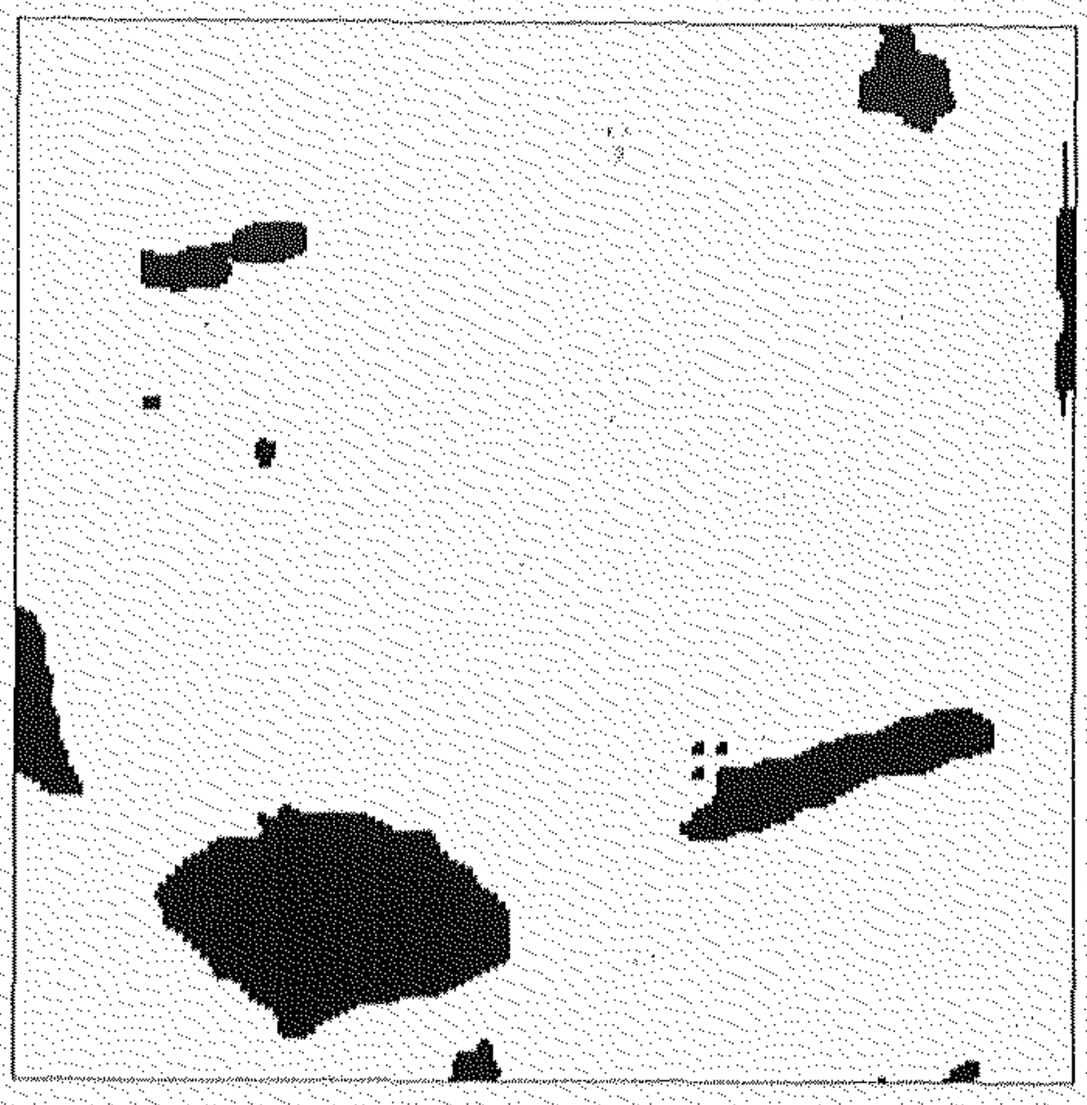


(b)

Figure 5.9: (a) Limestone micrograph image containing foraminifera (at 54×). (b) Segmentation mapping of (a).



(a)



(b)

Figure 5.10: (a) Limestone micrograph image containing foraminifera (at 54×). (b) Segmentation mapping of (a).

Chapter 6

Conclusion

6.1 Contribution of the present thesis

Texture analysis is an important element of vision, and occurs in a variety of natural and man-made objects. There are two major aspects of texture analysis: (a) To segment a given scene into different textural regions and (b) to provide each textural region some symbolic descriptions. In this thesis we looked at both aspects of the problem. Here textural segmentation is done by using fractal features. There is no unified approach for symbolic description of texture. We presented an algorithm to classify the textures into different symbolic category. The contribution of the present thesis is summarized below.

1. **Estimation of fractal dimension:**
   Several authors proposed techniques of estimating fractal dimension. Some of the techniques are accurate and cover the full dynamic range of Fractal Dimensions(FD), but are computationally expensive, while others are computationally attractive but do not cover the full dynamic range. To overcome these shortcomings the differential box counting (DBC) method of estimation of FD has been proposed, which is computationally attractive and also covers the full dynamic range of FD.

2. **Estimation of Generalized (Fractal) dimension :**
The properties of self-similarity of a set of points can be characterized by fractal dimension. For non-isotropic or inhomogeneous scaling properties, the sets require a more general treatment and one has to generalize the analysis using the concept of multifractal which implies a continuous spectrum of exponents for the systems characterization. Generalized dimension (GD) is a feature to evaluate multifractals. A method of estimating generalized dimension of texture is presented, which can be used in image analysis, in general, and texture segmentation, in particular.

3. **Fractal dimension based feature detection :**
A balanced set of features based on fractal dimension and multifractals are proposed for texture segmentation. These features are intended to capture roughness, coarseness and directionality of the textures.

4. **Modification of clustering approach :**
One of the major drawbacks of many clustering methods is the selection of seed points. An interesting algorithm is presented to take care of this problem.

5. **Orientation detection in texture field :**
A simple orientation detection and gradation technique has been proposed which is based on evidence accumulation as in Hough Transform.

6. **Broad taxonomical grouping of textures :**
Using directionality and the FD based features, a technique for broad taxonomical grouping as *random*, *oriented*, and *structural* textures has been proposed.

7. **Automatic representative size detection :**
An algorithm has been proposed to find the smallest area to be observed to understand a texture image. The significance of texture is lost if an area smaller than this representative size is observed.

8. **Reservoir rock gradation study :**
   An initial gradation of reservoir rock is done. Both the sandstone and limestone samples (microphotographs) are considered for this purpose.

6.2 Scope of Future Work

1. It has been noted that different methods of estimation of FD yields different results. Even with the same method, different scale range (box size) chosen for estimation of FD may lead to different FD values. It may be interesting to find an analytical range of box size. Some studies are made in this respect [28] but further study is necessary.

2. Estimated FD does not reflect the extent of fractalness of a fractal image. Future work can be done on devising an easy and efficient method for computing a degree of fractalness of an image.

3. For synthetic fractal image with a supplied FD value, the estimated FD often does not tally with the supplied FD. It may be interesting to find the reasons and modify the FD estimation technique, if necessary.

4. Most of the works with FD features relate to image segmentation by region based methods. Problems like texture edge detection may be tried with FD features.

5. A better and faster estimation method of generalized dimension (multifractal) could be suggested. The generalized dimension of different orders of the original and transformed image can be studied for various image processing problems.

6. In case of multispectral texture images, *e. g.*, in satellite imagery, an optimal method can be worked out to combine the *FDs* of different spectral bands for texture segmentation.

7. We have made a preliminary study on gradation of petroleum bearing (reservoir) rocks. It can be integrated into a more powerful knowledge based system of quality grading of reservoir rocks.

Bibliography

[1] A. E. Adams, W. S. Mackenzie, and C. Guilford. *Atlas of Sedimentary Rocks under the Microscope.* Longman, London, 1988.

[2] N. Ahuja. *Mosaic models for image analysis and synthesis.* PhD thesis, Univ. Maryland, College Park, 1979.

[3] N. Ahuja and A. Rosenfeld. Mosaic models for textures. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-3:1–11, 1981.

[4] J. Alloimonos and J. Swain. Shape from texture. In *Proc. 9th International Joint conf. on Artificial Intelligence*, pages 926–931, 1985.

[5] M. Amadasun and R. King. Textural features corresponding to textural properties. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-19:1264–1274, 1989.

[6] R. Bajcsy. Computer description of textured surface. In *Proc. 3rd International Joint conf. on Artificial Intelligence*, pages 572–579, 1973.

[7] R. Bajcsy. *Computer identification of textured visual scenes.* Technical Report, Stanford Univ., Palo Alto, CA, 1972.

[8] R. Bajcsy. Computer identification of visual surfaces. *Computer Graphics and Image Processing*, 2:118–130, 1973.

[9] R. Bajcsy and L. Lieberman. Texture gradient as a depth cue. *Computer Graphics and Image Processing*, 5:52–67, 1976.

[10] M. F. Barnsley. *Fractal everywhere.* Academic Press, San Diego, CA, 1988.

[11] M. F. Barnsley and A. D. Sloan. Methods and apparatus for image compression by iterated function systems. 1990. US Patent 4,941,193.

[12] J. Beck. Textural segmentation, second order statistics, and textural elements. *Biol. Cybern.*, 48:125–130, 1983.

[13] F. Bergholm. *Detection of edges and busy regions using Canny's edge detection.* Technical Report, Royal Institute of Technology, Stockholm, Sweden, 1988.

[14] S. Blacher, F. Brouers, and G. Ananthakrishna. Multifractal analysis of real heterogeneous materials. In *Proc. 11th ACTA Stereol, Irvine, CA*, pages 349–354, 1992.

[15] D. Blostein and N. Ahuja. Representation and three-dimensional interpretation of image texture: an integrated technique. In *Proc. First Int. Conf. on Computer Vision*, pages 444–450, 1987.

[16] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-12:55–73, 1990.

[17] P. Brodatz. *Texture: A Photographic Album for Artists and Designers.* Dover, New York, 1966.

[18] M. J. Carlotto. Texture classification based on hypothesis testing approach. In *Proc. 7th Int. Conf. Pattern Recognition*, pages 93–96, 1984.

[19] S. Chatterjee and R. Chellappa. Maximum likelihood texture segmentation using Gaussian Markov random field models. In *Proc. IEEE Conf. Computer Vision, Graph, Pattern Recog.*, 1985.

[20] B. B. Chaudhuri, B. Chanda, and D. Dutta Majumder. Some algorithms for image enhancement incorporating human visual response. *Pattern Recognition*, 17:423–428, 1984.

[21] B. B. Chaudhuri, P. Kundu, and N. Sarkar. Detection and gradation of oriented textures. *Pattern Recognition Letters*, 14:147–153, 1993.

[22] B. B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-17:, 1995.

[23] B. B. Chaudhuri, N. Sarkar, and P. Kundu. An improved fractal geometry based texture segmentation technique. *Proc. IEE-partE*, 140:233–241, 1993.

[24] R. Chellappa. Two-dimensional discrete Gaussian Markov random field models for image processing. In *Progress in Pattern Recognition*, Elsevier Science, North Holland, 1985.

[25] R. Chellappa, S. Chatterjee, and R. Bagdazian. Texture synthesis and compression using Gaussian random field model. *IEEE Trans. on System, Man, and Cybernetics*, SMC-15:298–303, 1985.

[26] C. C. Chen, J. S. Daponte, and M. D. Fox. Fractal feature analysis and classification in medical imaging. *IEEE Trans. on Medical Imaging*, 8:133–142, 1989.

[27] J. Chen and A. K. Jain. A structural approach to identify defects in textured images. In *Proc. of IEEE International Conf. on System, Man, and Cybernetics*, pages 29–32, 1988.

[28] S. S. Chen, J. M. Keller, and R. M. Crownover. On the calculation of fractal features from images. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-15:1087–90, 1993.

[29] D. Chetverikov. Detecting defects in texture. In *Proc. of the 9th International Conf. on Pattern Recog.*, pages 61–63, 1988.

[30] D. Chetverikov. Generating contrast curves for texture regularity analysis. *Pattern Recognition Letters*, 12:437–444, 1992.

[31] P. Cielo. *Optical techniques for industrial inspection.* Academic Press, San Diego, CA, 1988.

[32] M. Clark, A. C. Bovik, and W. S. Geisler. Texture segmentation using Gabor modulation/demodulation. *Pattern Recognition Letters*, 6:261–267, 1987.

[33] P. H. Coleman and L. Pietronero. The fractal structure of the universe. *Physics Report*, 213:311–391, 1992.

[34] R. W. Conners and C. A. Harlow. Towards a structural textural analyzer based on statistical methods. *Computer Vision Graphics and Image Processing*, 12:224–256, 1980.

[35] R. W. Conners, C. W. Mcmillin, K. Lin, and R. E. Vasquez-Espinosa. Identifying and locating surface defects in wood: part of an automated lumber processing system. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-5:573–583, 1983.

[36] R. W. Conners, M. M. Trivedi, and C. A. Harlow. Segmentation of a high resolution urban scene using texture operators. *Computer Vision, Graphics, and Image Processing*, 25:273–310, 1984.

[37] R. Creutzberg and E. Ivanov. Computing fractal dimension of image segments. In *Proc. III International conf. on Computer Analysis and of Images and Patterns CAIP '89*, 1989.

[38] G. C. Cross and A. K. Jain. Markov random field texture models. *IEEE Trans Pattern Anal. Machine Intell.*, PAMI-5:25–39, 1983.

[39] B. F. Curtis. Measurement of porosity and permeability. In R. E. Carver, editor, *Procedures in Sedimentary Petrology*, Wiley-Interscience, New York, 1971.

[40] L. S. Davis, S. John, and J. K. Aggrawal. Texture analysis using generalised co-occurrence matrices. *IEEE Trans Pattern Anal. Machine Intell.*, PAMI-3:251–259, 1979.

[41] P. Dewaele, P. Van Gool, and A. Oosterlinck. Texture inspection with self-adaptive convolution filters. In *Proc. of the 9th International Conf. on Pattern Recog.*, pages 56–60, 1988.

[42] J. A. D. Dickson. A modified staining technique for carbonates in thin-section. *Nature*, 205:587, 1965.

[43] V. Di Gesu and M. Tripiciano. Comparision of morphological operators for grey tones images. In *Proc. 5th Int. Conf. Image Analysis and Processing*, pages 59–64, World Scientific, Singapore, 1989.

[44] J. M. H. du Buf, M. Kardan, and M. Spann. Texture feature performance for image segmentation. *Pattern Recognition*, 23:291–309, 1990.

[45] M. P. Dubuisson and R. C. Dubes. Efficacy of fractal features in segmenting images of natural textures. *Pattern Recognition Letters*, 15:419–431, 1994.

[46] C. Dyer, T. Hong, and A. Rosenfeld. Texture classification using gray level co-occurrence based on edge maxima. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-10:158–163, 1980.

[47] F. Farrokhnia. *Multi-channel filtering techniques for texture segmentation and surface quality inspection.* PhD thesis, Michigan State Univ., Michigan, 1990.

[48] A. F. Fox. *The world of oil.* Pergamon Press, London, 1964.

[49] D. Gabor. Theory of communication. *J. Inst. Elect. Eng.*, 93:429–457, 1946.

[50] M. M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4:172–179, 1975.

[51] J. J. Gangepain and C. Roques-Carmes. Fractal approach to two dimensional and three dimensional surface roughness. *Wear*, 109:119–126, 1986.

[52] J. J. Gibson. *The perception of visual world.* John Wiley, New York, 1982.

[53] M. F. Glaessner. *Principle of Micropalaeontology.* Hafman Publisher, London, 1963.

[54] L. Glass. Moire effect from random dots. *Nature*, 243:578–580, 1969.

[55] P. Grassberger. Generalized dimension and strange attractors. *Phy. Lett. A*, 97:227–230, 1983.

[56] L. G. C. Hamey. *Computer Perception of Repetitive texture.* PhD thesis, Carnegie Mellon University, Pittsburgh, 1988.

[57] R. M. Haralick. Statistical and structural approaches to texture. *Proc. IEEE*, 67:786–804, 1979.

[58] R. M. Haralick, J. C. Mohammed, and S. W. Zucker. Compatibilities and the fixed points of arithmetic relaxation processes. *Computer Graphics and Image Processing*, 13:242–256, 1980.

[59] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-3:610–621, 1973.

[60] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29:100–132, 1985.

[61] H. Harmes, U. Gunzer, and H. M. Aus. Combined local color and texture analysis of stained cells. *Computer Vision, Graphics, and Image Processing*, 33:364–376, 1986.

[62] K. C. Hayes, A. N. Shah, and A. Rosenfeld. Texture coarseness : further experiment. *IEEE Trans. on System, Man, and Cybernetics*, SMC-4:467–472, 1974.

[63] D. C. He, L. Wang, and J. Guibert. Texture feature extraction. *Pattern Recognition Letters*, 6:269–273, 1987.

[64] G. D. Hobson and E. N. Tiratsoo. *Introduction to Petroleum Geology.* Gulf Publishing Co., Houston, 1981.

[65] P. V. C. Hough. A method and means for recognizing complex patterns. 1962. US Patent 3,069,654.

[66] J. Y. Hsiao and A. A. Sawchuk. Supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 11:1279–1292, 1989.

[67] J. Y. Hsiao and A. A. Sawchuk. Unsupervised textured image segmentation using feature smoothing and probabilistic relaxation techniques. *Computer Vision, Graphics, and Image Processing*, 48:1–21, 1989.

[68] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in cats visual cortex. *J. Phyiol.*, 166:106–154, 1962.

[69] K. Ikeuchi. Shape from regular pattern. In *Proc. IEEE 5th Int. Conf. Pattern Recognition*, pages 1032–1039, 1980.

[70] H. A. Ireland. Preparation of thin section. In R. E. Carver, editor, *Procedures in Sedimentary Petrology*, Wiley-Interscience, New York, 1971.

[71] A. L. Jacquin. *A fractal theory of iterated Markov operators, with applications to Digital Image coding.* PhD thesis, Department of Mathematics, Georgia Inst. of Tech., 1989.

[72] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.

[73] A. K. Jain, F. Farrokhnia, and D. H. Alman. Texture analysis of automotive finishes. In *Proc. of SME Machine Vision Applications Conf.*, pages 1–16, 1990.

[74] J .J. Jiang and A. A. Sawchuk. Noise updating repeated weiner filter and other adaptive noise smoothing filters using local image statistics. *Appl. Opt*, 25:2326–2337, 1986.

[75] B. Julesz. Texton gradients: the texton theory revisited. *Biol. Cybern.*, 54:245–251, 1986.

[76] B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981.

[77] B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, 8:84–92, 1962.

[78] B. Julesz and J. R. Bergen. Textons: the fundamental elements in preattentive vision and perception of textures. *Bell System Technical Journal*, 62:1619–1645, 1983.

[79] K. I. Kanatani. Detection of surface orientation and motion texture by a stereological technique. *Artificial Intelligence*, 23:213–237, 1984.

[80] S. Kanmani, C. B. Rao, D. K. Bhattacharjee, and B. Raj. Multifractal analysis of stress corrosion cracks. In *Proc. 11th ACTA Stereol, Irvine, CA*, pages 349–354, 1992.

[81] R. L. Kashyap, , and A. Khotanzad. Rotation invariant texture classification using circular random field models. In *Proc. IEEE Comput. Soc. Conf. on Computer Vision Pattern Recognition*, pages 194–200, 1983.

[82] R. L. Kashyap, , and A. Khotanzad. A stochastic model based technique for texture segmentation. In *Proc. IEEE 7th Int. Conf. Pattern Recognition*, pages 1202–1205, 1984.

[83] R. L. Kashyap. Analysis and synthesis of image patterns by spatial interaction models. In L. N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, Elesvier, North-Holland, 1981.

[84] M. Kass and A. Witkin. Analyzing oriented patterns. *Computer Vision, Graphics, and Image Processing*, 37:362–385, 1987.

[85] B. H. Kaye. Fractal dimension and signature waveform characterization of fine particle shape. *American Laboratory*, 55–63, 1986.

[86] J. Keller, R. Crownover, and S. Chen. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and Image Processing*, 45:150–160, 1989.

[87] J. R. Kender. Shape from texture: a computational paradigm. In *Proc. Image Understanding Workshop*, pages 134–138, 1979.

[88] J. R. Kender. Shape from texture: an aggregation transform that maps a class of textures into surface orientation. In *Proc. 6th Int. Joint Conf. on Artificial Intelligence*, pages 475–480, 1979.

[89] A. Khotanzad and J. Y. Chen. Unsupervised segmentation of textured images by edge detection in multidimensional features. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-11:414–421, 1989.

[90] L. Kirvida. Texture measurements for the automatic classification of imagery. *IEEE Trans. on Electromagnetic Compatibility*, 18:38–42, 1976.

[91] W. C. Krumben. Size frequency distribution of sediments. *J. Sedimentary Petrology*, 4:65–77, 1934.

[92] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel. Adaptive noise smoothing filter for images with signal dependent noise. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-7:165–177, 1985.

[93] P. Kundu and B. B. Chaudhuri. Fuzzy feature-based texture classification. *Pattern Recognition Letters*, 14:825–832, 1993.

[94] G. H. Landeweerd and E. S. Gelsema. The use of nuclear texture parameters in automatic analysis of leukocytes. *Pattern Recognition*, 10:57–61, 1978.

[95] K. I. Laws. Rapid texture identification. *Proc. SPIE*, 238:376–380, 1980.

[96] K. I. Laws. *Textured image segmentation.* PhD thesis, Univ. Southern California, Los Angeles, CA, 1980.

[97] J. G. Leu and W. G. Wee. Detecting the spatial structure of natural textures based on shape analysis. *Computer Vision, Graphics, and Image Processing*, 31:67–88, 1985.

[98] J. Levy Vehel, P. Mignot, and J. P. Berroir. *Texture and multifractals: new tools for image analysis.* Technical Report, INRIA, Roquencourt, France, 1992.

[99] B. S. Lipkin and L. E. Lipkin. Textural parameters related to nuclear maturation in the granulocytic leukocytic series. *Journal of Histochemistry and Cytochemistry*, 22:583–593, 1974.

[100] A. I. Loversen. *Geology of petroleam.* CBS Publishers and Distributors, Delhi, 1985.

[101] G. E. Lowitz. Can a local histogram really map texture information. *Pattern Recognition*, 16:141–147, 1983.

[102] S. Y. Lu and K. S. Fu. A syntactic approach to texture analysis. *Computer Graphics and Image Processing*, 7:303–330, 1978.

[103] A. Lundervold. *Ultrasonic tissue characterization - A Pattern Recognition approach.* Technical Report, Norwegian Computing Center, Oslo, Norway, 1992.

[104] W. S. MacKenzie. *Atlas of igneous rocks and their textures.* John Wiley, New York, 1982.

[105] B. B. Mandelbrot. *Fractal Geometry of Nature.* Freeman Press, San Francisco, 1982.

[106] B. B. Mandelbrot. *Fractals – From, Chance and Dimension.* Freeman Press, San Francisco, 1977.

[107] B. B. Mandelbrot and J. Van Ness. Fractional brownian motion, fractional noise and applications. *SIAM rev.*, 10:422–437, 1968.

[108] J. Mao and A. K. Jain. Texture classification and segmentation using simultaneous autoregressive models. *Pattern Recognition*, 173–188, 1992.

[109] G. Matheron. *Random sets and integral geometry.* Wiley, New York, 1975.

[110] E. F. McBride. Mathematical treatement of size distribution data. In R. E. Carver, editor, *Procedures in Sedimentary Petrology*, Wiley-Interscience, New York, 1971.

[111] B. H. McCormick and S. N. Jayaramamurthy. Time series model for texture synthesis. *Int. J. Comput. Inform. Sci.*, 3:329–343, 1974.

[112] F. Meyer. Iterative image transformations for an automatic screening of cervical smears. *Journal of Histochemistry and Cytochemistry*, 27:128–135, 1979.

[113] L. Mezei, M. Puzin, and P. Conroy. Simulation of patterns of nature by computer graphics. *Inform. Process.*, 74:52–56, 1974.

[114] O. R. Mitchel, C. R. Myres, and W. Boyne. A max-min measure for image texture analysis. *IEEE Trans. on Computer*, 2:408–414, 1977.

[115] W. W. Moorhouse. *The study of rokcs in thin section.* Harper and Row, New York, 1964.

[116] J. L. Muerle. Somethoughts on texture discrimination by computer. In *Picture Processing and Psychoppictorics*, Academic Press, New York, 1970.

[117] W. Muller and W. Hunn. Texture analyzer system. *Industrial Research*, 49–54, 1974.

[118] M. Nagao and T. Matsuyama. Edge preserving smoothing. *Computer Graphics and Image Processing*, 9:394–407, 1979.

[119] M. Nagao, H. Tanabe, and K. Ito. Agricultural land use classification of of aerial photographs by histogram similarity methods. In *Proc. 3rd Int. Joint Conf. on Pattren Recognition*, pages 669–672, 1976.

[120] H. Nakatani, S. Kimura, O. Siato, and T. Kitahashi. Extraction of vanishing point and its application. In *Proc. 5th Int. Conf. on Pattren Recognition*, pages 370–372, 1980.

[121] P. P. Ohanian and R. C. Dubes. Performance evaluation for four classes of texture features. *Pattern Recognition*, 25:819–833, 1992.

[122] Y. Ohta, K. Maenobu, and T. Sakai. Obtaining surface orientation from texels under perspective projection. In *Proc. 7th Int. Joint Conf. on Artificial Intelligence*, pages 746–751, 1981.

[123] J. D. Orford and W. B. Whalley. The use of fractal dimension to characterize irregular-shaped particle. *Sedimentology*, 30:655–668, 1983.

[124] T. Pavlidis. *Algorithms for graphics and image processing*. Springer-Verlag, New York, 1988.

[125] T. Pavlidis and P. C. Chen. Segmentation by texture using co-occurrence matrix and split-and-merge algorithm. *Computer Graphics and Image Processing*, 10:172–182, 1979.

[126] H. O. Peitgen and D. Saupe. *The science of fractal images*. Springer-Verlag, New York, 1988.

[127] S. Peleg. New probabilistic relaxation scheme. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 2:72–75, 1980.

[128] S. Peleg, J. Naor, R. Hartley, and D. Avnir. Multiple resolution texture analysis and classification. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-6:518–523, 1984.

[129] A. P. Pentland. Fractal based description of natural scenes. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-6:661–674, 1984.

[130] A. P. Pentland. Shading into texture. *Artificial Intelligence*, 29:147–170, 1986.

[131] F. J. Pettijohn. *Sedimentary Rocks*. Harper and Row, New York, 1975.

[132] C. Pickover and A. Khorasani. Fractal characterization of speech waveform graphs. *Computer Graphics*, 10:51–61, 1986.

[133] M. Pietikainen and A. Rosenfeld. Image segmentation by texture using pyramid node linking. *IEEE Trans. on System, Man, and Cybernetics*, SMC-11:822–825, 1981.

[134] M. Pietikainen, A. Rosenfeld, and L. S. Davis. Texture classification using averages of local pattern matches. In *Proc. IEEE 6th Int. Conf. Pattern Recognition*, pages 301–303, 1982.

[135] W. A. Pryor. Grain shape. In R. E. Carver, editor, *Procedures in Sedimentary Petrology*, Wiley-Interscience, New York, 1971.

[136] A. R. Rao. *A taxonomy for texture description and identification*. Springer-Verlag, New York, 1990.

[137] A. R. Rao and G. L. Lohse. Identifying high level features of texture perception. *Graphical Model and Image Processing*, 55:218–233, 1993.

[138] A. R. Rao and B. G. Schunck. Computing oriented texture fields. In *Proc. Conf. Computer Vision and Pattern Recognition*, page , 1989.

[139] J. P. Rigaut. Automated image segmentation by mathematical morphology and fractal geometry. *Journal of Microscopy*, 150:21–30, 1988.

[140] E. Rignot and R. Kwok. Extraction of textural features in sar images: statistical model and sensitivity. In *Proc. Int. Geoscience and Remote Sensing Symposium*, pages 1979–1982, 1990.

[141] M. C. Riley. A new roundness scale for sedimentary particles. *J. Sedimentary Petrology*, 11:94–97, 1953.

[142] A. Rosenfeld and B. Lipkin. Texture synthesis. In B. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*, Academic Press, New York, 1970.

[143] A. Rosenfeld and E. Troy. *Visual texture analysis*. Technical Report, Univ. of Maryland, College Park, Maryland, 1970.

[144] N. Sarkar and B. B. Chaudhuri. An efficient approach to estimate fractal dimension of texture image. *Pattern Recognition*, 25:1035–1041, 1992.

[145] N. Sarkar and B. B. Chaudhuri. An efficient differential box counting approach to compute fractal dimension in image. *IEEE Trans. on System Man and Cybernetics*, SMC-24:115–120, 1994.

[146] N. Sarkar and B. B. Chaudhuri. Multifractal and generalized dimension of gray-tone digital images. *Signal Processing*, 42:, 1995.

[147] B. Schacter, A. Rosenfeld, and L. S. Davis. Random mosaic models for textures. *IEEE Trans. on System, Man, and Cybernatics*, SMC-8:694–702, 1978.

[148] P. H. Schiller, B. L. Finlay, and S. F. Volman. Quantitative studies of single-cell properties in monkey striate cortex ii. orientation specificity and ocular dominance. *J. Neurophysiol.*, 39:1320–1333, 1976.

[149] J. Serra. Theoretical bases of the leitz texture analysis systems. *Leitz Sci. Tech. Inform.*, 1:101–136, 1974.

[150] J. Serra and G. Verchary. Mathematical morphology applied to fiber composite materials. *Film Sci. Tech*, 6:141–158, 1973.

[151] L. H. Siew, R. M. Hodgson, and E. J. Wood. Texture measures for carpet wear assessment. *IEEE Trans Pattern Anal. Machine Intell.*, PAMI-10:92–105, 1988.

[152] J. Sklansky. Image segmentation and feature extraction. *IEEE Trans. on System, Man, and Cybernetics*, SMC-8:237–247, 1978.

[153] A. Spry. *Metamophic rocks*. Pergamon Press, London, 1969.

[154] S. Sternberg. Biomedical image processing. *Computer*, 16:22–33, 1983.

[155] R. Sutton and E. L. Hall. Texture measures for automatic classification of pulmonary disease. *IEEE Trans on Computers*, C-21:667–676, 1972.

[156] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Trans on System, Man, and Cybernetics*, SMC-8:460–473, 1978.

[157] D. Terzopoulus and S. Zucker. Detection of osteogenesis imperfecta by automated texture analysis. *Computer Graphics and Image Processing*, 20:229–243, 1982.

[158] F. Tomita, Y. Shirai, and S. Tsuji. Description of textures by a structural analysis. *IEEE Trans. Pattern Anal. and Machine Intell.*, 4:183–191, 1982.

[159] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley, Reading, MA, 1982.

[160] M. Tuceryan and A. K. Jain. Texture analysis. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *The handbook of pattern recognition and computer vision*, World Scientific Publishing Co., New York, 1992.

[161] L. Van Gool, P. Dewaele, and A. Oosterlick. Texture analysis anno 1983. *Computer Vision, Graphics, and Image Processing*, 29:336–357, 1985.

[162] F. M. Vilnrotter, R. Nevatia, and K. E. Price. Structural analysis of natural textures. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 8:76–89, 1986.

[163] H. Voorhees and T. Poggio. Computing texture boundary from images. *Nature*, 333:364–367, 1988.

[164] R. Voss. Random fractals: characterization and measurement. In R.Pynn and A. Skjeltorp, editors, *Scaling Phenomena in Disordered Systems*, Plenum, New York, 1986.

[165] R. F. Voss. Fourier synthesis of Gaussian fractals: 1/f noises, landscapes and flakes. In *Tutorials on state of the art image synthesis, Detroit*, 1983.

[166] H. A. Wadell. Sphericity and roundness of rock particles. *J. Geology*, 41:310–331, 1933.

[167] E. L. Walker and T. Kanade. *Shape recovery of a solid of revolution from apparent distortion of patterns.* Technical Report, Carnegie-Mellon University, Pittsburg, USA, 1984.

[168] J. Wan, J. Mao, and C. Wang. Multiresolution rotation invariant simultaneous autoregressive model for texture analysis. In *Proc. IEEE 9th Int. Conf. Pattern Recognition*, pages 845–847, 1988.

[169] M. Werman and S. Peleg. Min-max operators in texture analysis. *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-7:730–733, 1985.

[170] J. Weszka, C. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-6:269–285, 1976.

[171] A. P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, 1981.

[172] R. Yokoyama and R. M. Haralick. Texture pattern image generation by regular Markov chain. *Pattern Recognition.*, 11:225–234, 1979.

[173] R. Yokoyama and R. M. Haralick. Texture synthesis using a growth model. *Computer Graphics and Image Processing*, 8:369–381, 1978.

[174] C. Zhuang and S. Dunn. The amplitude varying rate statistical approach for texture classification. *Pattern Recognition Letters*, 11:143–149, 1990.

[175] G. Zorpette. Fractal: not just another pretty picture. *IEEE Spectrum*, 25:29–31, 1988.

[176] S. W. Zucker. Computational and psychophysical experiments in grouping. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, Academic Press, New York, 1983.

[177] S. W. Zucker and D. Terzopoulus. Finding structure in co-occurrence matrices for texture analysis. *Computer Graphics and Image Processing*, 12:286–308, 1980.

Publications of the Author which are related to this thesis

1. "An Efficient Approach to Estimate Fractal Dimension of Textural Images", *Pattern Recognition*, v-25, pp. 1035-1041, 1992
   Co-author : B. B. Chaudhuri.

2. "Detection and gradation of oriented texture", *Pattern Recognition Letters*, v-14, pp. 147-153, 1993
   Co-author : B. B. Chaudhuri and P. Kundu.

3. "An Efficient Approach to Compute Fractal Dimension of Texture Image", *Proc. 11th IAPR, The Hague*, 1992 pp. 358-361
   Co-author : B. B. Chaudhuri.

4. "An Improved Fractal Geometry Based Texture Segmentation Technique", *Proc IEE part-E*, V-140, pp. 233-241, 1993
   Co-author : B. B. Chaudhuri and P. Kundu.

5. "An Efficient Differential Box Counting Approach to Compute Fractal Dimension of Image", *IEEE System, Man, and Cybernatics*, V-24, pp. 115-120, 1994
   Co-author : B. B. Chaudhuri.

6. "Texture Segmentation using Fractal Dimension", *IEEE Pattern Analysis and Machine Intellligence*, PAMI-17, no-1, 1995 (to appear)
   Co-author : B. B. Chaudhuri.

7. "Multifractal and Generalized Dimension of Gray-tone Digital Images", *Signal Processing*, V-42, no-2, 1995 (to appear)
   Co-author : B. B. Chaudhuri.