REPORT DOCUMENTATION PAGE

Form Approved
OBM No. 0704-0188

1. Agency Use Only *(Leave blank).*	2. Report Date. 1993	3. Report Type and Dates Covered. Final - Proceedings

4. Title and Subtitle.
Seafloor Characterization Using Texture

5. Funding Numbers.
Program Element No. 0602435N
Project No. 03585
Task No. MOG
Accession No. DN255031
Work Unit No. 13512E

6. Author(s).
Suresh Subramaniam*, Herb Barad*, Andrew B. Martinez*, and Brian S. Bourgeois

7. Performing Organization Name(s) and Address(es).
Naval Research Laboratory
Mapping, Charting and Geodesy Branch
Stennis Space Center, MS 39529-5004

8. Performing Organization Report Number.
NRL/PP/7441--93-0013

9. Sponsoring/Monitoring Agency Name(s) and Addres[illegible]
Naval Research Laboratory
Exploratory Development Program Group
Stennis Space Center, MS 39529-5004

10. Sponsoring/Monitoring Agei Report Number.
NRL/PP/7441--93-0013

11. Supplementary Notes.
Published in IEEE.
*Tulane Univ.
New Orleans, LA 70118

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

Texture analysis is performed on multibeam sonar imagery. A set of fourteen texture features is computed using co-occurrence matrices to form the feature space. The dimensionality of the feature space is reduced by extracting the principal components from the original feature space. Classification of the image is performed on the principal components using K-Means algorithm. Results indicate that seafloor bottom types can be characterized by analyzing the texture of the bathymetric sonar images.

9 PG

14. Subject Terms.	15. Number of Pages.
Hydrography, bathymetry, optical properties, remote sensing, reverberation	8
	16. Price Code.

17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

SEAFLOOR CHARACTERIZATION USING TEXTURE

Suresh Subramaniam, Herb Barad, and Andrew B. Martinez
Department of Electrical Engineering, Tulane University, New Orleans, LA 70118
Brian Bourgeois
Naval Research Laboratory, Stennis Space Center, MS 39529

Abstract

Texture analysis is performed on multibeam sonar imagery. A set of fourteen texture features is computed using co-occurrence matrices to form the feature space. The dimensionality of the feature space is reduced by extracting the principal components from the original feature space. Classification of the image is performed on the principal components using K-Means algorithm. Results indicate that seafloor bottom types can be characterized by analyzing the texture of the bathymetric sonar images.

1 Introduction

Multibeam echo sounders have recently been used to map seafloor with high resolution. However, bathymetry does not yield other seafloor characteristics such as bottom type and seafloor roughness. These characteristics can be inferred from the fluctuations in the backscattered acoustic signal [3]. Texture is a property that provides information about the roughness of an object. In this paper, we attempt to use texture analysis to extract information about the roughness of the seafloor, and classify areas with similar features together.

A large swath of seafloor can be mapped by analyzing the backscattered data from each *ping*, i.e., transmission cycle. The data from each ping are divided into 256 parts corresponding to 256 directions. The data in each of the directions form a *bin*. A powerful tool to extract texture information, the *co-occurrence matrix*, is employed on the sonar image. A co-occurrence matrix is formed for the data in each of the bins in a ping. A set of 14 texture features is then computed from the co-occurrence matrix. The results of the texture feature extraction are combined to form a single 14-dimensional texture feature image data set. A *principal components* transform is applied to this 14-dimensional space and the most significant components are used to form the feature space for the classifier. A simple clustering algorithm is used to classify the seafloor area surveyed into "similar" regions.

2 Image Construction From Sonar

The requisite data for texture analysis are provided by a multibeam sonar system. Sonar energy from the system projector array located on the ship's hull impinges on the bottom of the ocean as a narrow beam The echo is received by an array of hydrophones mounted athwartships (perpendicular to the projector). Beamforming is performed by computing a 256 point Fast Fourier Transform (FFT) on the raw data collected. This yields an array of return intensities–an intensity for each beamformer bin and sample time. The sampling rate is such that approximately 1000 samples are obtained for each bin. This is the data on which texture analysis is performed.

3 Texture Analysis

Even though a precise definition of texture does not exist, image texture can be qualitatively described as having one or more properties of fineness, coarseness, smoothness, granulation, randomness, lineation, or being mottled, irregular, or hummocky [13]. Basically, texture refers to repetition of basic texture elements called *texels*. A texel contains several pixels whose placement could be periodic, quasi-periodic or random. The two dimensions of texture are the description of these texels, and the spatial distribution of these primitives. A texel is a set of pixels with some common tonal feature or local properties.

There is a close relationship between tone and texture. Consider a small area of an image. As the number of distinguishable tonal properties decreases, the

tonal properties will predominate. When the small-area patch is the size of one pixel so that there is only one discrete feature, the only property present is simple gray tone. As the number of distinguishable tonal properties increases, the texture property will predominate. When the spatial pattern of the tonal primitives is random and the gray tone varies widely between primitives, a fine texture results. As the spatial pattern becomes more definite and and the tonal regions increase in size, a coarser texture results. Thus we see that, to characterize texture, equal consideration must be given to both the tonal primitives and the spatial dependence between the primitives.

A number of approaches to analyzing texture have been presented in the literature. A comprehensive survey of the basic approaches is presented by Haralick [13]. Some of the methods of texture analysis are co-occurrence matrices [14], gray level run lengths [9], Markov models [5], [11], structural analysis [13], and fractal analysis [1].

Conners and Harlow [17] present a theoretical comparison of the Co-occurrence Method (their term is Spatial Gray Level Dependence Method), the Gray Level Run Length Method, the Gray Level Difference Method, and the Power Spectral Method, and conclude that the Co-occurrence Method is the most powerful algorithm for texture analysis. Hence our choice of co-occurrence matrices for the analysis of texture.

Mastin *et al.* used co-occurrence methods on SAR imagery of coastal waters for obtaining offshore wind direction and for the estimation of aerodynamic roughness parameters [4]. Aloimonos addressed the problem of determining shape from texture [7]. Haralick *et al.* tried to use textural features of photomicrographs of sandstones to identify the type of rocks and applied textural analysis to satellite imagery [14].

3.1 Co-occurrence Matrices

As remarked earlier, knowledge of the second order statistics of the image is required to adequately describe texture. A histogram is an estimate of the first order statistics of an image (or of a region). The normalized histogram is computed as

$$P(i) = \frac{N(i)}{N}, i = 0, 1, \ldots, 2^b - 1 \qquad (1)$$

where $N(i)$ is the number of pixels in the image (region) with intensity value i, N is the total number of pixels in the image (region), and b is the number of bits per pixel in the image.

The analog of the histogram for second order statistics is the co-occurrence matrix. The co-occurrence matrix is also computed in a "census" fashion by counting pairs of occurrences of pixel values given a certain spatial relationship for the pair. The normalized co-occurrence matrix elements are computed as

$$P(i, j, d, \theta) = \frac{N(x_1, x_2)}{N}, |x_1 - x_2| = D(d, \theta) \qquad (2)$$

for pairs of pixels at locations x_1 and x_2 having intensity values i and j, respectively. The distance measure $D(d, \theta)$ states that the spatial relationship of the pair of pixels is that they are located at a distance magnitude d apart and at an angle θ (or $\theta + \pi$) from each other.

A complete set of co-occurrence matrices would cover all values of d and θ over a meaningful range. The values of θ would vary between 0 and π using some number of discrete steps. The value for d would range from 1 up to some distance where the correlation between pixels is still significant.

In practice, several co-occurrence matrices are computed for several integral values of d and for four values of θ, 0, $\pi/4$, $\pi/2$, and $3\pi/4$. Figure 4 shows several computed co-occurrence matrices for a simple example 2 bit/pixel image.

One of the disadvantages of the co-occurrence matrix method is the potentially large amount of data computed for different pairs of d and θ. Only four of the many possible co-occurrence matrices are computed in Figure 4. However, co-occurrence statistics are powerful in that they are invariant under monotonic intensity transformations [15].

3.2 Texture Analysis of Multibeam Sonar Data

A co-occurrence matrix with $d = 1$ and $\theta = 0$ is formed for the array of data in each bin. We keep $\theta = 0$ since the pings are not georefernced and the process of georeferencing will suppress many texture attributes. In other words, it is assumed that the data from two adjacent pings are independent. The distance d is kept small because we expect the primitives to be relatively small. Haralick *et al.* [14] present a set of 14 texture features that can be computed from co-occurrence matrices. The meanings of some of the features are also presented. Some of the important texture features computed are *angular second moment, contrast, correlation, inverse difference moment*, and *entropy*.

We compute the 14 features from the co-occurrence matrix formed for each ping along all 256 bins. The corresponding features for all pings are concatenated to form a 14-dimensional texture feature data set. A

detailed description of texture feature extraction is provided in [6].

4 Data Reduction

The magnitude of data generated by texture analysis is inappropriate for classification purposes. We have to reduce the data and, at the same time, retain the most useful part of the data. There are several methods of reducing data. A very common approach employed is the extraction of principal components from the original data.

The principal components transform (also known as the *discrete Karhunen-Loève transform or the Hotelling transform*), is used to transform the 14-dimensional data set into another feature space of the same dimension. In our case, each pixel in the texture image is represented by a 14-dimensional vector, say $\mathbf{x}$. We have a total of 283 × 197 (=55751) such vectors. The mean vector and covariance matrix of the vectors are easily estimated. The principal components transform is computed using the equation

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m_x}) \tag{3}$$

where $\mathbf{A}$ is the matrix formed from a sorted set of eigenvectors of the covariance matrix $\mathbf{C_x}$ and $\mathbf{m_x}$ is the mean vector. By discarding those eigenvectors for which the corresponding eigenvalues are relatively small, the size of the matrix $\mathbf{A}$ can be suitably reduced to make $\mathbf{y}$ a vector of desired dimension. It can be shown that the new feature space is one in which the data from different features are uncorrelated and the particular choice of eigenvectors retains the maximum possible information [12].

We retain the first four components of the transformed data and make it the feature space for the classifier. The four principal component images are shown in Figure 5. The sorted eigenvalues of the covariance matrix are 8805.451, 2.207, 1.376, 0.8799, 0.1558, 0.04136, 0, 02413, 1.124×10^{-3}, 7.709×10^{-4}, 3.251×10^{-4}, 3.109×10^{-4}, 2.338×10^{-4}, 3.19×10^{-5}, and 1.748×10^{-5}. The first four principal components are chosen because they represent the entire feature space with a *mean square error* of 0.224104. This is computed using the relationship

$$e_{ms} = \sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{K} \lambda_j \tag{4}$$

where, in our case, $n = 14$, $K = 4$, and λ_j's are the eigenvalues.

5 Classification

Classification, the goal of pattern recognition, is the process of assigning each of the objects of interest to one of a number of categories or *classes*. The objects of interest are called *patterns*. Each of these patterns is represented by a vector of dimension, say, n, where n is the number of *features* used to represent the pattern. As an example, consider the problem of recognizing a digit from 0 to 9. Suppose that each of the digits is contained in a grid divided into n small squares as in Figure 1. Then one way to form a feature vector is to measure the area occupied by the digit in each-of the small squares.

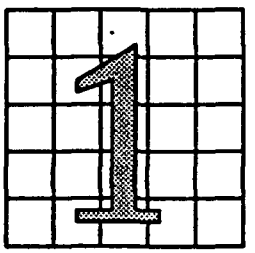


Figure 1: The digit "1" in a grid of 25 squares

Thus, $X = [x_1 x_2 \cdots x_n]^T$ is a vector representing a digit, where $x_1, x_2, \ldots, x_n$ are the areas occupied by the digit in the little squares $1, 2, \ldots, n$ respectively. The problem of pattern recognition is, therefore, to assign a class label to an unknown pattern, or a random vector in the feature space. A function which separates any two classes is called a *discriminant function* and a network which classifies a pattern based on the values of the discrimant functions is called the *classifier* [8].

The probability of misclassification is the key factor in analyzing the performance of any classifier. It is well known that the optimal classifier, assuming the distributions of the random vectors are known, is the *Bayes classifier* which is studied under *statistical hypothesis testing* [8]. However, the implementation of the Bayes classifier is difficult because of its complexity, especially when the dimensionality is high.

If there exists a set of patterns, the class assignment of which is already known, the process of classification is called *supervised classification*. A portion of the set of labeled patterns, called the *training set*, is used to derive a classification algorithm. The rest of the labeled patterns comprise the *test set* and are used to test the classification algorithm and evaluate its performance. Once the algorithm is tuned to provide the desirable performance, it can be used on initially

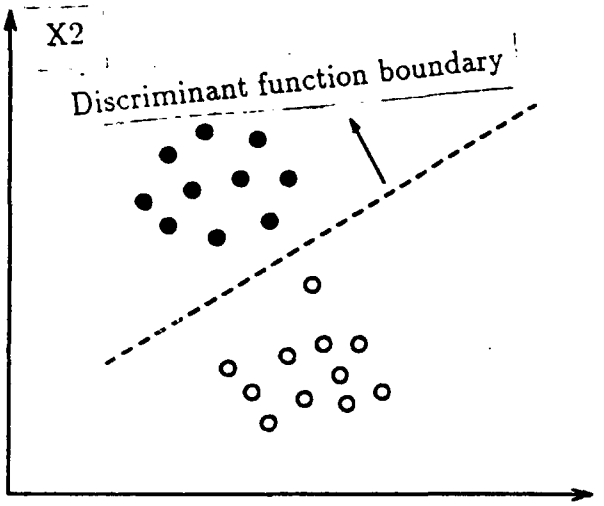


Figure 2: A clustering example

unlabeled patterns [2].

Sometimes, however, we may not have a set of labeled patterns and we may not even know the number of classes. The problem is not only to classify the data, but also to define the classes. Several approaches to this problem have been dealt with in the literature [16], [10], [19]. The ensuing discussion is solely concerned with a procedure of seeking clusters of points in the measurement space called *clustering*.

5.1 Clustering

Clustering is the process by which samples with "similar" features are combined together to form a single cluster. The fundamental issue in the clustering problem is the definition of a cluster or, equivalently, the choice of features. There are two approaches to clustering, the *parametric approach* and the *nonparametric approach*. Parametric approaches require either *clustering criteria* to be defined or assume a mathematical form for the distribution of the samples. Most often, a clustering criterion is defined and samples are assigned to classes such that this criterion is optimized. A typical example of assuming a mathematical form for the distribution is the problem of finding parameters that best fit the data, the distribution of which is assumed to be a summation of normal distributions. On the other hand, *nonparametric approaches* separate samples according to the *valley* of the density function [8]. Figure 2 shows a set of two-dimensional patterns grouped into two clusters using a distance function as the criterion.

In the absence of *ground-truth* information, *i.e.*, a training set, we are led to employ a clustering algorithm on the samples in the multidimensional texture feature image. Most of the clustering algorithms which seek to optimize a clustering criterion are iterative. These algorithms are not guaranteed to converge and even if they do converge, they may converge to a local minimum rather than the global minimum. A branch and bound procedure which is guaranteed to find the global minimum is given in [18]. This algorithm, however, is not practicable for the magnitude of data in our case. A simple clustering algorithm [2] which optimizes a criterion iteratively, is given below. This is followed by a very popular algorithm called the *K-Means Algorithm* [2] which optimizes a specific criterion.

5.2 A Simple Clustering Algorithm

Suppose the number of clusters N_c is known. Let $\mathbf{X}$ denote the set of samples $\{\mathbf{x}^{(i)}\}$ to be classified and Ω an ordered set of class labels assigned to the samples. Further suppose that $\omega_1, \omega_2, \ldots, \omega_{N_c}$ are the labels and $\Omega^{(r)}$ is the set of class labels at the rth iteration. Assume that the classification is optimal when a criterion function $J(\mathbf{X}, \Omega)$ is minimized. The following general procedure can be used in an attempt to minimize J.

1. Choose an initial classification Ω^0 and compute J.
2. Change the classification in a way that tends to decrease J.
3. If it is not possible to decrease J in step 2, then stop; else go to step 2.

Since the variables in this optimization problem are the class labels which are discrete, gradient search techniques cannot be used. One way to solve this problem is to determine the change in the class label for each sample that would result in the greatest decrease in J and apply these changes in step 2. Suppose that $\Omega^{(r)} = \{\omega_{s_1}, \omega_{s_2}, \ldots, \omega_{s_N}\}$ where N is the number of samples in $\mathbf{X}$. If ΔJ_i is the largest negative change in J that can be made by reclassifying sample $\mathbf{x}^{(i)}$, and $\omega_{s'_i}$ is the corresponding new label for $\mathbf{x}^{(i)}$, then the new set of labels is $\Omega^{(r+1)} = \{\omega_{s'_1}, \omega_{s'_2}, \ldots, \omega_{s'_N}\}$.

Observe that since ΔJ_i is evaluated by making one change at a time and $\Omega^{(r+1)}$ is obtained by making all changes simultaneously , the change in the value of J is not, in general, equal to $\sum_{i=1}^{N} \Delta J_i$. It is highly likely, though, that the criterion function has decreased.

5.3 The K-Means Algorithm

This algorithm uses a similarity measure that is the Euclidean distance of the samples and a criterion J

defined by

$$J = \sum_{k=1}^{N_c} \sum_{\mathbf{x}^{(i)} \sim \omega_k} |\mathbf{x}^{(i)} - \mu_k|^2 \qquad (5)$$

where the second sum is over all samples in the kth cluster and μ_k is the "center" of the cluster. It is easily seen that for a fixed set of samples and class assignments, J is minimized by choosing μ_k to be the *sample mean* of the kth cluster. Moreover, when μ_k is the sample mean, J is minimized by assigning $\mathbf{x}^{(i)}$ to the class of the cluster with the nearest mean. A number of other criteria are given in [8].

The complete algorithm is outlined below.

1. Make an arbitrary assignment of samples to clusters.
2. Compute the sample mean of each cluster.
3. Reassign each sample to the cluster with the nearest mean.
4. If there is no change in classification, then stop; else go to step 2.

6 Results

The K-Means algorithm was applied to the 4-dimensional texture image and the result of the classification is shown in Figure 3. The contrast of the image has been improved to make it more visible. The number of clusters chosen was 15. The solid vertical band along the middle of the image corresponds to the nadir and near-nadir portion of the seafloor. Lack of adequate backscatter information from this portion is the cause for the apparent homogeneity of the seafloor near the nadir. The presence of areas on the seafloor with different textures is evident.

7 Conclusion

In this paper, we have presented a novel approach to seafloor characterization. The results indicate that texture analysis of bathymetric sonar images is a powerful technique to determine seafloor roughness and bottom type. In the absence of ground-truth information, unsupervised techniques produced remarkably good results. More work needs to be done in analyzing the texture features and associating each cluster with a type of seafloor surface (labeling). We conclude that texture analysis is a useful tool for seafloor mapping.

Figure 3: The texture image after clustering

8 Acknowledgments

The authors acknowledge the Office of Naval Technology, project element 602435N, managed by Dr. Herb Eppert, NRL, Stennis Space Center, MS. This paper, NRL Contribution Number JA351:100:92 is approved for public release; distribution is unlimited.

Drs. Barad and Martinez are also sponsored by NSF/Louisiana Board of Regents grant NSF/LEQSF(1992-93)-ADP-04.

References

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

[2] Therrien C.W. *Decision Estimation and Classification.* John Wiley, 1989.

[3] Christian de Moustier. Beyond bathymetry. *J. Acoust. Soc. Am.*, 79(2):316–331, Feb 1986.

[4] Mastin G.A., Harlow C.A., Huh O.K., and Hsu S.A. Methods of obtaining offshore wind direction and sea-state data from X-band aircraft SAR

imagery of coastal waters. *Journ. IEEE Oceanic Eng.*, OE-10(2):159–174, Apr 1985.

[5] Cross G.R. and Jain A.K. Markov random field texture models. *Trans. IEEE Pattern Anal. and Machine Intell.*, PAMI-5(1):25–39, Jan 1983.

[6] Barad H., Martinez A.B., Bourgeois B., and Kaminsky E.J. Acoustical boundary location through texture analysis of multibeam bathymetric sonar data. Submitted to Marine Technology Society Journal, 1992.

[7] Aloimonos J. Visual shape computation. *Proc. IEEE*, 76:899–916, Aug 1988.

[8] Fukunaga K. *Introduction to statistical pattern recognition.* Academic Press, 1990.

[9] Galloway M.M. Texture analysis using gray level run lengths. *CVGIP*, 4:172–179, Jun 1975.

[10] Devijver P.A. and Kittler J. *Pattern Recognition: A Statistical Approach.* Prentice Hall, Inc., 1962.

[11] Chellappa R. and Manjunath B.S. Unsupervised texture segmentation using Markov random field models. *Trans. IEEE Pattern Anal. and Machine Intell.*, 1991.

[12] Gonzalez R.C. and Woods R.E. *igital Image Processing.* Addison-Wesley, 1992.

[13] Haralick R.M. Statistical and structural approaches to texture. *Proc. IEEE*, 67(5):786–804, May 1979.

[14] Haralick R.M., Shanmugham K., and Dinstein I. Textural features for image classification. *Trans. IEEE Systems, Man, and Cyber.*, SMC-3(6):610–621, Nov 1973.

[15] Haralick R.M. and Shapiro L.G. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.

[16] Duda R.O. and Hart P.E. *Pattern Recognition and Scene Analysis.* John Wiley, 1973.

[17] Conners R.W. and Harlow C.A. A theoretical comparison of texture algorithms. *Trans. IEEE Pattern Anal. and Machine Intell.*, PAMI-2(3):204–222, May 1980.

[18] Koontz W.L.G., Narendra P.M., and Fukunaga K. A branch and bound clustering algorithm. *Trans. IEEE Computers*, C-24:908–915, 1975.

[19] Meisel W.S. *Computer-Oriented Approaches to Pattern Recognition.* John Wiley, 1972.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

		Gray Level			
		0	1	2	3
	0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
Gray	1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
Level	2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
	3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

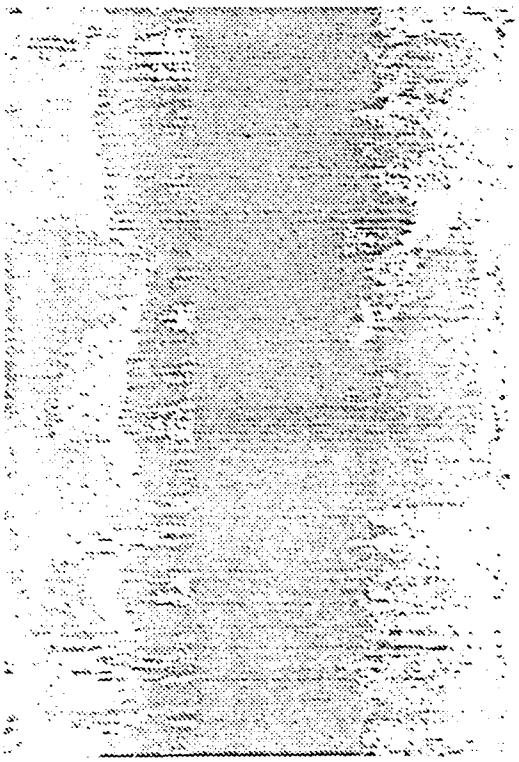
$$P_{(\theta=0°,d=1)} = \frac{1}{16}\begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix} \qquad P_{(\theta=90°,d=1)} = \frac{1}{16}\begin{pmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

$$P_{(\theta=135°,d=1)} = \frac{1}{16}\begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix} \qquad P_{(\theta=45°,d=1)} = \frac{1}{16}\begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

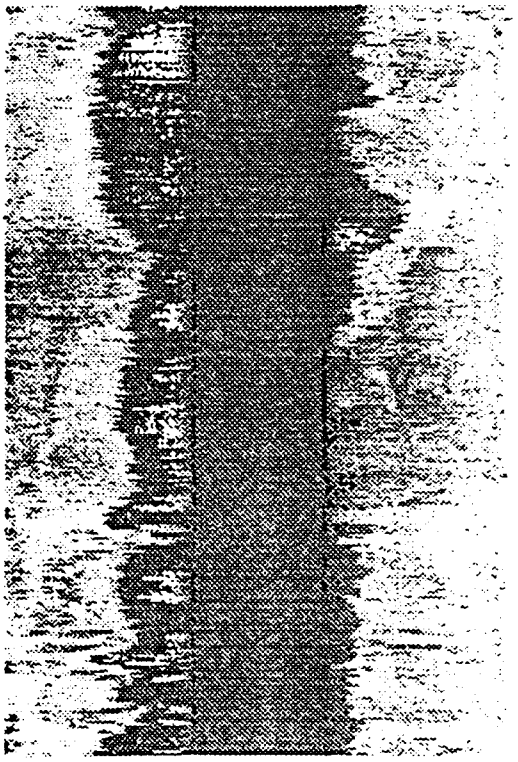
Figure 4: Sample co-occurrence calculations [14]

First component

Second component



Third component



Fourth component

Figure 5: The four principal components