Hierarchical Maximum Entropy Partitioning of Visual Feature Frequency Matrices for Texture Classification

D.K.Y. Chiu
Department of Computing & Information Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1
C.Y.C. Bie H.C. Shen
Department of Systems Design Engineering
Univeristy of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Abstract

In this paper, a new representation scheme is presented for texture images. Texture is represented by feature frequency matrices. The feature frequency matrices are then partitioned into a set of feature submatrices by hierarchical maximum entropy partitioning. The feature frequency matrices for four important visual textural properties are investigated. The efficacy of the representation is tested by texture classification experiments.

1 Introduction

Texture is a very useful surface feature in image analysis such as classification and segmentation of image regions. Three aspects are important in texture analysis, namely, representation, classification and segmentation of texture images. The purpose of texture representation is to capture the essential information to reduce the need of extensive computation for further analysis such as classification and segmentation. Thus, a compact, efficient and reliable texture representation is the basic requirement for texture analysis. Many researchers have developed different texture features [4]. Texture features corresponding to human visual perception aspects such as coarseness, contrast etc. have been explored by many researchers. Julesz [5] began working on visual pattern discrimination of textures. Tamura et al. [10] developed a set of textural features corresponding to some visual perception. Amadasun et al. [1] proposed a new set of textural features corresponding to certain textural properties. Shen and Bie [8] used 2-D feature frequency matrix (FFM) to measure four important visual properties (coarseness, contrast, directionality, and line-likeness).

2-D FFM is proven to be an efficient measurement of visual textural properties [8]. However, the large amount of data in the representation makes it difficult to be used. It is necessary to develop a compact form of 2-D FFM. In this paper, a set of 2-D FFMs will be used to measure the visual texture features including coarseness, contrast, directionality and lineliness. Then, a hierarchical maximum entropy partitioning scheme (HMEP) applied to the 2-D FFMs can identify peaks and valleys of sample frequency observation in a compressed form. It is based on a dynamic selection and compression of relevant texture characteristics with minimum loss of expected information or entropy [3]. Instead of using a single numerical value, the HMEP scheme on a 2-D FFM represents the visual texture properties in terms of their statistical and structural relationship. Classification experiments has been conducted to demonstrate the effectiveness of the proposed representation.

2 The FFM Representation

In [9], we have proposed a scheme of representation of textures, i.e. a set of feature frequency matrices (FFM) of different dimensions at various resolution levels. In this section, we shall briefly give the definitions of feature frequency matrices. It is believed that images at different resolution levels include different information for texture differentiation [7]. Given an image at certain resolution level, a set of operators can be used to extract the corresponding features. By applying one operator to the image, a feature image can be obtained. From the feature images, feature frequency matrices in different dimensions can be extracted and they are defined as follows:

Definition 1: Feature Image, (F)

> Given an $(N1 \times N2)$ image S at any resolution level and an $n \times n$ feature extraction operator OP, a feature image F is obtained by convolving OP over the image S.

Definition 2: 1-D Feature Frequency Matrix

> Given a feature image F, the 1-D feature frequency matrix is defined by a vector FFM_1, where the element $ffm(i)$ is the number of pixels with feature value i in F.

Definition 3: 2-D Feature Frequency Matrix

Given two feature images, $F1$ and $F2$, the 2-D feature frequency matrix for $F1$ and $F2$ is defined as a matrix FFM_2 where the element $ffm(i,j)$ is the number of pixels with feature value i in $F1$ and feature value j in $F2$.

3 FFM Representation of Visual Textural Properties

Visual textural properties are reflected by the distributions of image features, or simply, the changes of the interrelationship of grey levels under certain structural constraints in an image. There are many proposed visual textural properties. Four of them here are considered to be very important for visual perception [1, 10]. They are: **coarseness, contrast, directionality, and line-likeness.**

3.1 Coarseness

Coarseness is probably one of the most fundamental visual property of texture. It is resolution dependent, i.e. the coarseness characteristic changes according to the size of the neighborhood in the evaluation. Coarseness can also be measured based on its "edgeness" and the extent of repeatedness per unit area. From these intuitive ideas, we design the following operators. We define the deviation over a local window and the edges per unit area to measure the coarseness. One deviation operator and four edge operators are used to extract coarseness information.

Deviation operator calculates the grey value deviation over an $m \times m$ window on the original image. By applying the operator to the original image, deviation feature image D_σ is obtained. The deviation at pixel (i,j) is computed by:

$$\sigma_{i,j} = \frac{1}{m^2} \sum_{k=i-(m-1)/2}^{i+(m-1)/2} \sum_{l=j-(m-1)/2}^{j+(m-1)/2} |g(k,l) - \mu_{i,j}| \quad (1)$$

where

$$\mu_{i,j} = \frac{1}{m^2} \sum_{k=i-(m-1)/2}^{i+(m-1)/2} \sum_{l=j-(m-1)/2}^{j+(m-1)/2} g(k,l) \quad (2)$$

and $g(k,l)$ is the grey level at pixel (k,l). Four edge detectors for 0, 45, 90, 135 degree directions shown in Figure 1 are used to detect edge response in every pixel of the original image. The maximum response of the four edge detectors in a pixel is the edge response in the pixel. One threshold is set for edge response. If the edge response is larger than the threshold, the output of edge detector is '1'; otherwise the output is '0'. Therefore, by applying the four edge detectors to the original image, we obtain an edge feature image with pixel value either '1' for large edge response or '0' for small edge response. Then we apply an edge-per-unit-area operator on the edge feature image. The operator simply counts the '1's in an $m \times m$ region. Thus,

$$\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix} \qquad \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix}$$

(a) $\qquad$ (b)

$$\begin{matrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{matrix} \qquad \begin{matrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{matrix}$$

(c) $\qquad$ (d)

Figure 1: Four edge detectors

the output of the edge-per-unit-area operator $epa_{i,j}$ is between 0 and m^2. From the output of the edge-per-unit-area operator, we obtain another feature image of edge-per-unit-area D_{epa}. From D_{epa} and D_σ, the 2-D FFM F_{crs} can be obtained.

Independently, $(\sigma_{i,j})$ and $(epa_{i,j})$ both measure the coarseness information. Jointly, i.e. represented as 2-D FFM, they provide the statistical-structural information about coarseness $(epa_{i,j}, \sigma_{i,j})$. $(\sigma_{i,j})$ to some extent is related to $(epa_{i,j})$. However, two pixels have the same epa value does not mean they have the same σ value, and vice versa. If texture is coarse, $\sigma_{i,j}$ should be relatively small indicating large primitives. Edge-per-unit-area value should be small too. Therefore, the peaks of the FFM, in this case, will occur close to the origin of the axes.

3.2 Contrast

Two operators are used to extract contrast information. One operator extracts information about the local dynamic range of grey levels D_{drg}, the other extracts the grey level information at each pixel D_{gl}. From the 2 feature images, a 2-D FFM for contrast measure F_{con} can be obtained.

Let $s_1, s_2, ..., s_{m^2}$ represent the $m \times m$ grey level values in observation neighborhood of m-window at (i,j). The grey value is between 0 and $L-1$. The output of dynamic range operator is defined as:

$$drg_{i,j} = \frac{1}{m^4} \sum_{k=1}^{m^2} \sum_{p=1}^{m^2} |s_k - s_p| \quad (3)$$

where $0 \le drg_{i,j} \le L/2 - 1$. The 2-D FFM for contrast is a $L \times L/2$ matrix. While the dynamic range operator measures the local contrast, the grey level distribution reflects the global contrast. If the texture has a high local contrast, the dynamic grey level range have high values. If the texture has a high global contrast, the dynamic range of grey values will be relatively low and the peaks of grey level distribution will be far apart.

3.3 Directionality

Directionality measures the directional property of grey level change. Directionality can be measured based on the gradient operators, e.g. the Sobel operators as shown in Figure 2. In general, two $(m \times m)$ gradient operators can be defined to extract gradient

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

(b) Sx (a) Sy

Figure 2: Sobel Operators

information (magnitude and directionality) for a m-window. Then the magnitude and directionality can be normalized to P and K. Therefore, from the magnitude feature image and directionality feature image, $P \times K$ 2-D directionality FFM F_{dir} can be formed.

If texture image has a high directionality, F_{dir} should have some "banded" areas corresponding to the normalized directionality values. The extent of grey level changes in certain direction is reflected by the normalized magnitude.

3.4 Line-likeness

Line detectors and line-likeness detector can be used to detect line-likeness. Line detectors will find how sharp the edges for the texture are and line-likeness detector will find how edges are distributed in the texture. Four line detectors are used to detect lines in the directions: 0^o , 90^o, 45^o and 135^o. For (3×3) window, the four operators are defined in Figure 1. The outputs of four line detectors at (i, j) are denoted as $E_0(i, j)$, $E_{90}(i, j)$, $E_{45}(i, j)$, $E_{135}(i, j)$. Traditional methods use one edge detector to obtain one edge image. We combine the four edge feature images as one edge feature image, $E(i, j)$, and together with a line-like feature image, $B(i, j)$, to form a 2-D FFM. The edge feature image, $E(i, j)$, is defined as the maximum of $\{E_0(i, j), E_{90}(i, j), E_{45}(i, j), E_{135}(i, j)\}$ and is normalized to between 0 and $(L - 1)$. From the edge feature image, we obtain an edge register image, $R(i, j)$, which is defined as :

$$R(i,j) = \begin{cases} 0 & E(i,j) < T \\ 1 & E(i,j) = E_0(i,j) \\ 2 & E(i,j) = E_{45}(i,j) \\ 3 & E(i,j) = E_{90}(i,j) \\ 4 & E(i,j) = E_{135}(i,j) \end{cases}$$

where T is a threshold for edge response. Line-like operator will reveal how the edges are distributed in a $W \times W$ local observation window over $R(i, j)$. If pixel (i, j) is very line-like, the directionality of edges in the window will be relatively consistent. If the edges in the observation window are in different directions, then it is not very line-like at pixel (i, j) since the edges may 'cancel' each other and tend to form blobs. To account for the cancelling effect within a window, we define the line-likeness detector, $B(i, j)$, at pixel (i, j) in the following way:

Let K be the maximum value in the edge register image;

r_k be the number of pixels in the $W \times W$ window with edge register value k; and

a function $\beta(k)$ be defined as

$$\beta(k) = \begin{cases} 0 & k = 0; \\ |\beta(k-1) - r_k| & k > 0 \end{cases}$$

Then,

$$B(i,j) = \beta(K).$$

$B(i, j)$ is between 0 and W^2. If the texture within the window centred at (i, j) is more line-like, the directionality will be relatively consistent and $B(i, j)$ will be relatively large. If the texture is more blob-like, the directionality will be relatively evenly distributed and $B(i, j)$ will be relatively small. Thus, the 2-D FFM formed based on $E(i, j)$ and $B(i, j)$ is a $L \times (W^2 + 1)$ matrix.

If an image is not line-like, line-like response will be relatively low, F_{lin} will spread out along the low value line-likeness frequency axis. If an image has strong edge response, F_{lin} will spread out over high edge value areas. Usually, line-likeness response will be more spread out if there are more edges or if the edge responses are high.

4 HMEP of the Visual FFM

As demonstrated in [9], the 2D-FFM representation scheme is very useful for analyzing textures. The major disadvantage of the FFM representation is that the dimension of the matrices can be very large (in the order of 256 by 256) and the matrices are usually sparse. In [9], we have computed moments from the FFM and used them as features in the classification process which reached over 90% correct classification rate. The moment as a feature can be used to provide statistical information of the distributions. However, as feature vector for classification, much information is not retained from the FFM. Therefore, we propose the approach based on a hierarchical maximum entropy partitioning scheme [3]. The scheme partitions the FFM into submatrices dynamically to attain maximum entropy. Further partitioning of the submatrices can be performed to refine the partitions. Hence, a recursive algorithm is developed to performed the partitioning process. The end result of the scheme is a set of submatrices which encapsulates the original FFM.

The detailed theoretical development of the hierarchical maximum entropy partition (HMEP) scheme is described in [3]. In this section, we shall adapt the scheme to be applied to FFM. Let $FFM(i, j)$ be a 2D-feature frequency matrix with dimensions N_1 and N_2 and $FPM(i, j)$ be the corresponding probability matrix, defined by

$$FPM(i,j) = \frac{FFM(i,j)}{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} FFM(i,j)}.$$

The entropy of FFM is defined by Shannon's entropy measure,

$$H(FFM) = -\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} FPM(i,j) log FPM(i,j).$$

The partitioning process is to determine the number and locations of intervals (K_1, K_2) in N_1, N_2 respectively ($K_1 \times K_2 \ll N_1 \times N_2$) such that the set of ($K_1 \times K_2$) submatrices attains maximum entropy. By maximizing the entropy in the partitioning process, the distribution of FFM will be characterized by the submatrices. In order to dynamically compress the FFM, K_1 and K_2 must be sufficiently small. Mathematically, the process can be formulated as a dynamic programming problem. The problem can be stated as marmizing the objective function $H(FFM)$:

$$max\ H(FFM) = -\sum_{p=1}^{K_1}\sum_{q=1}^{K_2} FPM(R_{p,q}) log FPM(R_{p,q})$$

subject to

$$2 \le K_1 \le N_1$$

$$2 \le K_2 \le N_2$$

$$K_1 \times K_2 \ll N_1 \times N_2$$

where $R_{p,q}$ is the submatrix corresponding to the p^{th} and q^{th} intervals and $FPM(R_{p,q})$ is the probability associated with the submatrix.

Therefore, any suitable software to solve dynamic programming problem can be used to determine K_1 and K_2 and corresponding partitioning. Alternatively, we employ the hierachical maximum entropy partitioning scheme which is based on the principle that if we partition the FFM into n submatrices with maximum entropy, each submatrix will have the same frequency. The partitioning is performed recursively with the initial values of K_1 and K_2 (denoted by K_1^0 and K_2^0 respectively) determined empirically by the following steps:

1. Calculate the entropy of the original FFM:

$$H_0(FFM) = -\sum_{i=1}^{N1}\sum_{j=1}^{N2} FPM(i,j)\ log FPM(i,j)$$

2. Minimize the loss of entropy from partitioning by computing the number of submatrices, n, such that the entropy of the n submatrices, H_n is at least reasonably high [1] of that of the original, $H_0(FFM)$ and

$$H_n = -\sum_{i=1}^{n}\frac{1}{n} log \frac{1}{n} = log\ n$$

3. Choose the smallest integers, K_1^0 and K_2^0 such that

$$K_1^0 \times K_2^0 \ge n.$$

[1] In our experiment, we have chosen the value to be 50% of $H_0(FFM)$.

Note that the above method choose an initial small number of partitioning which minimizes loss of entropy (on Shannon's information). For computational convenience in the hierarchical partitioning scheme, we choose K_1^0 and K_2^0 to be powers of 2.

Once K_1^0 and K_2^0 are determined, further partitioning can be performed on each submatrix, $R(p,q)$, depending on (i) its statistical significance; (ii) its size; and (iii) the summed probability in it. The size and the probability (or frequency) value are threshold input parameters. To determine the statistical significance of a submatrix $R_{p,q}$, we consider whether the frequency of joint features in the submatrix deviates significantly from the expected frequency. A statistics of the deviation can be formulated as:

$$t(R_{p,q}) = \frac{|FFM(R_{p,q}) - exp(R_{p,q})|}{\sqrt{exp(R_{p,q})}}, \ \forall\, p, q \qquad (4)$$

where $exp(R_{p,q})$ is the expected frequency[2]. A submatrix, $R_{p,q}$ is selected if $t(R_{p,q})$ is greater than a threshold T. Otherwise, we say $R(p,q)$ is an uncovered submatrix and no further partitioning will be performed. Since $t(R_{p,q})$ is normally distributed [6], a threshold based on the normal distribution with a presumed confidence level can be chosen.

The process is applied to every submatrix generated until either it is not statistically significant, the submatrix or the frequency is too small. The result from the hierarchical partitioning is a set of submatrices representing the original FFM.

5 Experiments

Classification experiments are conducted to demonstrate the efficacy of the proposed texture features. Twenty texture images from [2] (Figure 3) are used in the experiments. Each image is digitized to 512×480 pixels with grey levels between 0 and 255. In order to reduce the computational time and the memory requirement, we transform the grey levels to between 0 and 63. Two hundred and ten texture training samples of size 64×64 are extracted from each of the 20 images (Figure 3). For each training sample, four types of 2-D FFM are built in the experiments: FFM of coarseness (denoted as Coa-FFM of size 26×32), FFM of contrast (denoted as Con-FFM of size 64×32), FFM of directionality (denoted as Dir-FFM of size 64×33), and FFM of line-likeness (denoted as Lin-FFM of size 64×26). A common partitioning scheme is necessary for each type of FFM in the classification experiments. This is achieved by considering the entire training set (20×210 samples) as the "texture universe". HMEP scheme is performed on each of the four types of FFM which is the sum of the respective FFM in the texture universe. Thus,

[2] There are various ways to determine $exp(R_{p,q})$. In our experiment, we have chosen one from [3] based on the uniformity distribution assumption.

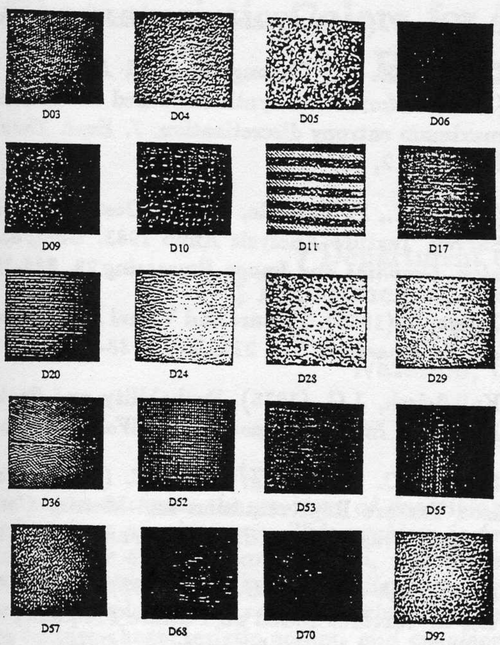

Figure 3: Twenty Texture Images from Brodatz

for the four types of 2-D FFM, we obtain four partitioning schemes: the Coa-FFM is partitioned into 34 submatrices, the Con-FFM to 66, the Dir-FFM to 84, and Lin-FFM to 29 submatrices. Figure 4 shows four types of summed FFM of texture universe and figure 5 illustrates the corresponding partitioning diagrammatically. With this partitioning scheme, individual samples can be characterized by the location and the number of "covered" submatrices in their respective FFM.

By comparing the frequencies of the corresponding submatrices, textures can be differentiated. We shall treat the four partitioned FFM as feature vectors, i.e.

$$Coa-FFM \rightarrow V_{coa}[0,1,...,33]$$

$$Con-FFM \rightarrow V_{con}[0,1,...,65]$$

$$Dir-FFM \rightarrow V_{dir}[0,1,...,83]$$

$$Lin-FFM \rightarrow V_{lin}[0,1,...,28]$$

where $V_s[i]$ is the summed frequency in the i^{th} partitioned submatrices of the respective FFM. Taking the Euclidean distance between two vectors as the distance measure, we can use the nearest neighbour rule as our classification scheme.

Two sets of test samples are chosen. The first set consists of 10 samples of size 64 × 64 pixels taken randomly from each of the 20 images, i.e. a total of 210 test samples. Special care was taken to ensure

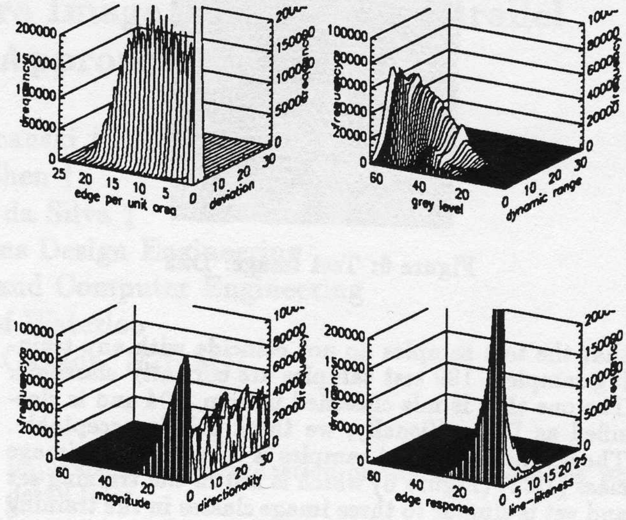

Figure 4: Texture Universes for Four Types of 2-D FFM

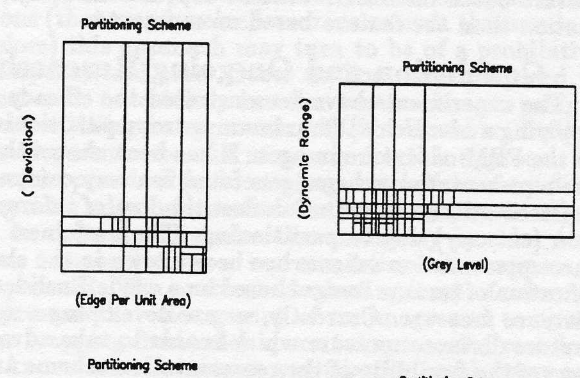

Figure 5: Partitioning based in the Texture Universe

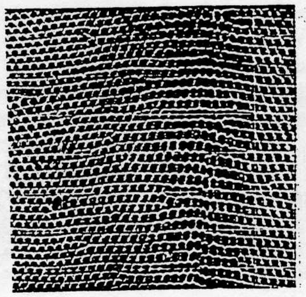

Figure 6: Test Image: D22

that the test samples do not coincide with any training samples. 199 test samples are correctly classified. The one that is mis-classified is from D04 and is classified as D92. Visually, we think this is acceptable. The second set of test samples is taken from an image class (D22) (Figure 6) which is not in the training set and yet is similar to three image classes in the training set, i.e. D03, D10 and D36. 56 test samples of size 64 $\times$ 64 pixels are taken from D22 to be classified. Feature based on the HMEP scheme attained 96% correct classification, i.e. 54 test samples are correctly classified. The remaining 2 samples are mis-classified as D05 and D28.

By comparing with the classification results in [9], where features are based on the moments of three 2-D FFMs and a 100 % correct classification are obtained for the first set experiment and 93 % correct classification for the second set, we can conclude that the feature based on HMEP scheme captures more information than the feature based on moments.

6 Conclusion and On-going Research

The experiments have demonstrated the efficacy of applying a hierarchical maximum entropy partitioning to the FFM of texture images. It has been shown that the representation scheme generated is a very compact representation and yet minimizes the loss of information (entropy) due to partitioning. The usefulness of the representation scheme has been shown in the classification of texture images based on a crude Euclidean distance measure. Currently, we are developing a new texture distance measure which is able to take advantage of the flexibility of the representation scheme and will be reported in due time.

7 Acknowledgement

The research is supported by the Manufacturing Research Corporation of Ontario which is one of the Centers of Excellence Supported by the Premier's Council Technology Fund of the province of Ontario, Canada and also by the individual Natural Sciences and Engineering Research Council of Canada Operating Grant and a grant from University of Guelph President's Research Enhancement Fund.

References

[1] Amadasun, M. and King, R. (1989). Textural Features Corresponding to Textural Properties. *IEEE Trans. on System, Man, and Cybernectics* Vol. 19, No. 5, 1264-1274.

[2] Brodatz, P. (1968). *Textures*, Reinhold, New York.

[3] Chiu, D. K. Y., Cheung, B., and A. K. C. Wong (1990). Information synthesis based on hierarchical maximum entropy discretization. *J. Expt. Theor. Artif. Intell.* 2, 117-129.

[4] Gool, L.V., P. Dewaele, and A. Oosterlinck (1985). Survey, Texture Analysis Anno 1983. *Computer Vision, Graphics and Image Processing* 29, 336-357.

[5] Julesz, B. (1965). Texture and Visual Perception. *Scientific American.* Vol. 212, No. 2, 38-54.

[6] Kalbfleisch, J.G. (1975). Probability and Statistical Inference. *Internal Report, U. of Waterloo, Canada.*

[7] Shen, H.C. and Wong, A. K. C. (1983). Generalised Texture Representation and Metric. *Computer Vision, Graphics, and Image Processing* 23, 187-206.

[8] Shen, H.C. and C.Y.C. Bie (1991). Visual Properties of Textures Based on Feature Frequency Matrices. *Proc. of Vision Interface '91*, Calgary, Alberta, Canada, 170-175.

[9] Shen, H.C. and C.Y.C. Bie (1991). Feature Frequency Matrices as Texture Image Representation, *Pattern Recognition Letters*, to appear.

[10] Tamura, H., S. Mori and I. Yamawaki, (1978). Textural Features Corresponding to Visual Perception. *IEEE Trans. on System, Man and Cybernetics*, Vol. SMC-8, No. 6.