Chapter 3

Texture Representation

3.1 INTRODUCTION

Texture analysis plays a major role in computer vision applications and image understanding, since most of the images contain textures. It includes the problems such as texture identification, representation, classification, texture segmentation, texture synthesis, object recognition and textured image compression. As the texture is a key feature for these problems, it must be properly represented before analyzing it. The proper representation of textures will lead to a right path on texture analysis. The texture of a surface is characterized by properties such as fine, coarse, smooth, granulated, rippled, mottled, irregular, random, lineated, etc. [Hara78]. Despite its ubiquity in scene analysis, there exists no precise definition for texture. Several authors have given a definition for textures in different ways based on its characteristics. Some of them are given below:

Picket [Pick70] has defined the texture as consisting of a large number of elements, each with some degree visibility, and on the whole, densely and evenly (possibly randomly) arranged over the field of views such that there is a distinct characteristic spatial repetitiveness in the pattern.
Vilnrotter, et al. [Viln86] described the texture as the pattern of the spatial arrangement of different intensities.

Ganesan and Bhattacharyya [Gane95] viewed textures as a composition of a large number of more or less ordered, similar elements or patterns.

Here we propose a definition of texture, which may be more appropriate for all types of textures.

Definition: A texture can be defined as a structure encompassing a large number of similar primitives or patterns that are scattered randomly or almost ordered with distinct characteristics of spatial repetitiveness.

Over the recent years, a number of approaches have been developed to solve the low-level texture analysis problem. The existing research works are broadly classified based on the first-order and second-order statistical properties, geometrical, structural, model-based and signal processing techniques [Tuce93]. Over a decade, Multichannel methods, Multiresolution analysis, Wavelet features, Gabor features [Chan93, Mojs97, Zhan98, Grig02] have been used extensively for low-level analysis of textures.

Among the existing methods [Bajs76, Hara78, Gool83, Tuce88, Chan93, Karu96, Zhan98, Grig02, Mahm03, Camp04], Haralick’s co-occurrence method [Hara73, Hadd93], Galloway’s run length method [Gall75], Fourier power spectrum method [Wesz76] and texture number method [He90] are well known. Ohanian and Dubes [Ohni92] compared the Markov Random Field model parameters, multi-channel
filtering features, fractal based features and co-occurrence features and reported that the performance of the co-occurrence matrix method was the best one for texture analysis of 2-D gray scale images. Strand and Taxt [Stra94] compared various methods of texture analysis, namely, co-occurrence matrix, run length methods, Fourier power spectrum and autocorrelation and reported that the co-occurrence based features give better results for classification, segmentation and object recognition. The co-occurrence matrix method [Hara73] is based on the joint probability distribution of pixels in an image. The matrix represents the joint probability of occurrence of gray levels between two pixels with a defined spatial relationship in an image. The spatial relationship is defined in terms of distance $d$ and angle $\theta$. Galloway [Gall75] proposed the run length method, in which, the run of gray levels has been used to characterise the textures for further analysis. This method faces some difficulties to represent the textures in the presence of noise. The Fourier power spectrum method was proposed by [Wesz76] and subsequently used by many researchers [Ausl83] to characterise the micro level textures present in the textural images. It considers the texture primitives to be appearing at regular periodic manner due to its periodicity. It does not characterise well the texture primitives in stochastic patterns and also shown that it is less efficient when compared to others. This is the main disadvantage of the Fourier power spectrum method.

Later, He and Li Wang [He90] proposed, a scheme for texture characterization and discrimination based on local as well as global properties, in which, a $3\times3$ subimage is considered and the neighbourhood pixel values in all the eight directions are compared to
its centre pixel value. Based on the outcome of this comparison the
subimage is mapped on to a set of ternary numbers called texture
number. The texture number ranges from 0 to 6560, and describes
the micro textures in the subimage and is called local descriptor. The
frequency of occurrences of these texture numbers represents the
global properties in the image. Ganesan and Bhattacharyya [Gane95]
proposed a new technique based on statistical approach. It is reported
that the scheme proposed by He and Wang does not distinguish the
textured region from the untextured regions with edges and also it is
time consumable to handle the texture number, since it is as high as
6561 components. The work proposed by Ganesan and
Bhattacharyya, reports that they mapped the small image region of
interest at centre pixel on to a set of binary numbers called pronum
where the pronum is an integer, and ranges from 0 to 255. The
frequency of occurrences of the pronum is called the prospectrum
that describes the global properties in the image. In the former
scheme, a small image region is represented by a number through the
positional number system with radix 3 and in the latter, the
positional number system with radix 2. In this thesis, it is believed
that the texture unit number measured based on the positional
number system could not give better results. The reasons are as
follows.

For instance, let us assume that the texture unit is 2 at locations
(1,1) and (3,3) of the subimage with size (3×3) in the former scheme.
This represents the texture unit numbers 2 (2×3^0 = 2) and 4374 (2×3^7
= 4374) at locations (1,1) and (3,3) respectively. In the later scheme,
let us assume that, the texture unit is 1 in those locations. This
represents the texture unit numbers 1 ($1 \times 2^0 = 1$) and 127 ($1 \times 2^7 = 127$) respectively. This positional number system shows a great discrimination between the pixels in the neighbouring locations in opposite directions to the centre pixel. The influences of pixels at locations (1,1) and (3,3) on its centre pixel are spatially same. So, the texture unit number measured based on the assignment of the positional number system is not rational, since it shows great differences on the weightage given to the positional number assigned to the pixels at locations (1,1) and (3,3) in both the schemes. This may lead to the wrong or partial representation of micro textures in the small image region.

Over a decade, Multichannel [Chen95], Multiresolution [Kris97, Come99], Wavelets transform [Chan93, Adam00, Deev03] and Gabor filters [Grig02] are widely used in low level image processing. Some of them are providing good results for any kind of applications. But at the same time, many of them produce a very low classification rate when texture samples are of small dimensions [Mojs95, Mojs97] especially in the field of medical images such as myocardial tissues, ultrasound images, echocardiography images, etc.

In recent years, model based methods, such as, Hidden Markov field [Chen95], Markov random field [Li00], Autoregressive [Delp79, Kada98], Multiresolution Guassian Autoregressive [Come99] and Gibbs field [Chal89] models have attracted many researchers for texture analysis such as classification, segmentation and textured image compression. They reported that these models gave better results for low-level texture analysis. Mahmoudi, et al., [Mahm03] used autocorrelation for image retrieval based on similarity by edge
orientation. Campisi, et al. [Camp04] have studied the autocorrelation and reported that it extracts the features of textured images for classification.

The texture in an image can be described by a number of primitives and their types, and the spatial organisation or layout of their primitives. The spatial organisation may be stochastic or periodic and may have a pairwise dependence of one primitive on a neighbouring primitive, or may have a dependence of $n$ primitives at a time. The dependence may be structural, probabilistic or functional like a linear dependence.

To characterise the texture, we must therefore characterise the tonal primitive properties as well as the spatial interrelationships between them. This implies that texture-tone is really a two-layered structure, the first layer having to do with specifying the local properties that manifest themselves in tonal primitives and the second layer having the organisation among the tonal primitives.

The autoregressive (AR) model is a way to use linear estimates of pixel's gray tone given the gray tones in a neighbourhood containing it in order to characterise the texture. For coarse texture, the coefficients will all be similar while for the Fine texture, the coefficients will have wide variation.

The linear dependence of one pixel of an image to another is well known and can be illustrated by the autocorrelation function. The autoregressive model for texture analysis exploits this linear dependence. The autocorrelation function is a feature, that tells about the size of the tonal primitives. If the tonal primitives of the image are
relatively large, then the autocorrelation will drop off slowly with distance. If the tonal primitives are small, then the autocorrelation will drop off quickly with distance. If the tonal primitives are spatially periodic, the autocorrelation function will drop off and rise again in a periodic manner. The relationship between autocorrelation function and the power spectral density function is well known: they are Fourier transforms of one another. In this sense, the combination of autoregressive and autocorrelation approaches is sufficient to capture everything about a texture.

A detailed account of how the proposed Full Range Autoregressive (FRAR) model based framework facilitates texture identification, representation and classification are discussed in this Chapter. The aforementioned low-level texture analyses are performed based on the autocorrelation function derived from the coefficients \( \Gamma_r \) of the proposed model. The coefficients \( \Gamma_r \) are computed from the model parameters \( K, \alpha, \theta \), and \( \phi \). The estimation procedures of these parameters are discussed in detail in the preceding chapter. The micro textures in a small image region are identified by employing the test for homogeneity of variances as described in [Pena02] among the autocorrelation coefficients. If the autocorrelation is highly significant, then it represents the responses towards the micro textures, otherwise the responses towards the untexturedness in the images is represented. Based on this autocorrelation value, we propose a number called \( texnum \) to represent the micro texture present in a small image region under analysis. By considering non-overlapping regions with raster scan fashion, we compute the frequency of number of occurrences of \( texnum \) in the entire image
and is called texspectrum. The texspectrum describes the global information of the texture present in the image. Based on the proposed representation scheme, the supervised and unsupervised classifications have been performed with the different textspectra as the training set for different textures present in the target image. The supervised classification is performed based on autocorrelation coefficients, which are subject to homogeneity tests, such as *t*-test and *Bartlett’s test* [Bere96, Bhat97] at desired levels of significance. The *k-means* [Fuku90, Lika03] algorithm is applied for unsupervised classification. A detailed discussion about both the classifications are given in section 3.4 of this Chapter.

### 3.2 Texture Identification

To identify the textures in the image, the model parameters $K$, $\alpha$, $\theta$ and $\phi$ are estimated, as discussed in Chapter 2, for small image region. The small image regions are considered by dividing the whole image into various regions of size $3 \times 3$. The model coefficients $\Gamma_r$s, ($r = 1,2$) are determined by applying the estimated parameters $K$, $\alpha$, $\theta$ and $\phi$ in equation (2.1), as introduced in section 2.2 of Chapter 2. The autocorrelation function $(\rho_k)$ is derived from the model coefficients $\Gamma_r$s as follows:

\[
\rho_1 = \frac{\Gamma_1}{1 - \Gamma_2}
\]

\[
\rho_2 = \frac{\Gamma_1^2 + \Gamma_1 - \Gamma_2^2}{1 - \Gamma_2^2} \quad \text{and}
\]

\[
\rho_3 = \frac{\Gamma_1(\Gamma_1^2 + 2\Gamma_1 - \Gamma_2^2)}{1 - \Gamma_2^2}
\]
Similarly, the $k^{th}$ order autocorrelations can be obtained by solving the equation (3.1) using recurrence relation. Their pattern is governed by the second order linear difference equation.

$$\rho_k = \Gamma_1 \rho_{k-1} + \Gamma_2 \rho_{k-2};$$

$$1 \leq k \leq m$$

From equation (3.1), the autocorrelation coefficients ($\rho_k$) are computed. To identify the micro level textures in the small image region, we have to test the autocorrelation for its significance. The test statistic as introduced in [Pena02] for autocorrelation is defined as follows.

$$D_m = n \left[ 1 - \left| \mathbf{R}_m \right|^{1/m} \right]$$

where, $\mathbf{R}_m$ is the correlation matrix built by using the standardized autocorrelation coefficients $\tilde{\rho}_k$. That is,

$$\mathbf{R}_m = \begin{bmatrix}
1 & \tilde{\rho}_1 & \ldots & \tilde{\rho}_m \\
\tilde{\rho}_1 & 1 & \ldots & \tilde{\rho}_{m-1} \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{\rho}_m & \tilde{\rho}_{m-1} & \ldots & 1
\end{bmatrix}$$

where, $\tilde{\rho}_k = \frac{(n+2)}{(n-k)} \rho_k^2$, $n$ is the number of samples and $m$ is the lag variable.

The significance test about the statistic $D_m$ can be based on the measure $\pm \alpha (\sigma/\sqrt{n})$, where $\alpha$ is level of significance and $\sigma$ is standard
deviation. To make a confidence test at a given significance level \( \alpha \) for the null hypothesis of no autocorrelation at lag \( k \), we have to compare the value of sample coefficient with the aforementioned measure. If the sample coefficient falls outside the given bands then the hypothesis is rejected at level of significance \( \alpha \) i.e. the autocorrelation does exist among the data. Otherwise the null hypothesis is accepted. If the autocorrelation is highly significant, then it is identified as the responses towards micro textures in the small image region. Otherwise, it is identified as the responses towards untexturedness.

### 3.3 Texture Representation

In order to represent the identified micro textured regions, the computed autocorrelation value is placed at the centre position of the small image region (3×3) in another matrix array with the same size corresponding to the actual image. This procedure is continued for all the subsequent image regions by sliding the window in raster scan fashion. The computed autocorrelation values fall in the range from \(-1\) to \(+1\). We use a simple transformation \((\rho \times 100) + 100\) on the autocorrelation values, to obtain decimal number, which ranges from 0 to 200, where \( \rho \) is the autocorrelation coefficient. Now, the encrypted local description of micro texture is quantified as a texture number. We call this number as \textit{texnum}. The texnum is a local descriptor as it describes the local information of the textured image region. The calculation of texnum is repeated for the entire image by considering subsequent overlapping image regions taken in the raster scan fashion. The number of occurrences of those texnums is called \textit{texspectrum}. A texspectrum of an image describes the textures present in the image globally. Since the texnum ranges from 0 to 200,
there will be totally 201 components in a texspectrum. Based on this texture number, the proposed scheme characterises and represents the different types of textures. It also explores the spatial interrelationship between the pixels and tonal primitives of the micro textures in the small image region since autocorrelation represents the relationship among the pixels.

3.4 Texture Classification

To validate the performance of the proposed texture representation scheme, we carry out the classification analysis on the global descriptors obtained in the previous section. The task of classification is the identification of groups of pixels or a region in the image that are cohesive to be clustered from other types of pixels or a region in the image and assign each possible cohesive group of pixel or regions to a known class of texture. The clustered regions are mutually exclusive and each region corresponds to a particular class of homogeneous texture. Generally, the method of classification is broadly categorized into two types, namely, supervised and unsupervised. The prior information about the image to be recognized is available in the supervised classification whereas no prior information is available in the unsupervised classification. The use of proposed texture representation scheme is highlighted in both the supervised and unsupervised classifications.

3.4.1 Supervised Classification

In order to classify the different types of textures present in the input test image, the supervised classification is presented in this section. The global descriptor, that is, texspectrum computed from the known input mages is used to classify the textures in the target
image. The well known two simple test statistics, *Bartlett’s test statistic* and *t-statistic* [Bere96, Bhat97] are employed to classify the input target image into known L classes of images with the use of prior information about the reference images Rj where j = 1,2, ... , L.

**3.4.1.1 Bartlett’s Test for the Homogeneity of variances**

The Bartlett’s test is used to test the homogeneity of variances among k samples. Equal variances across the samples are called homogeneity of variances.

Hypotheses:

\[ H_0: \sigma_1 = \sigma_2 = \ldots = \sigma_k \]  
\[ (Null \ hypothesis) \]

\[ H_a: \sigma_i \neq \sigma_j \]  
\[ for \ at least \ one \ pair \ (i, j). \]  
\[ (Alternative \ hypothesis) \]

Test Statistic: The Bartlet test statistic is used to test the equality of variances across the groups against the alternative hypothesis (H\(_a\)) that variances are unequal for at least two groups.

\[
B = \frac{(N - k) \ln(s_p^2) - \sum_{i=1}^{k} (N_i - 1) \ln(s_i^2)}{1 + (1/(3(k-1)))\left(\sum_{i=1}^{k} 1/(N_i - 1) - 1/(N - k)\right)}
\]

where, \( s_i^2 \) is the variance of the \( i^{th} \) group, N is the total sample size, \( N_i \) is the sample size of the \( i^{th} \) group, k is the number of groups and \( s_p^2 \) is the pooled variance. The pooled variance is the weighted average of the group variances and is defined as

\[
s_p^2 = \sum_{i=1}^{k} (N_i - 1)s_i^2/(N - k)
\]
The variances are judged to be unequal if, $B > \chi^2_{(a, k-1)}$ [Fish47], where, $\chi^2_{(a, k-1)}$ is the upper critical value of the chi-square distribution with $(k - 1)$ degrees of freedom and a significance level of $\alpha$.

### 3.4.1.2 Independent Two-sample t-test

The t-test is used to test the equality of means with the basic assumptions: (i) the samples are independent and (ii) the sample variances are equal.

The *two-sample t-test* is defined as follows:

**Hypotheses:**

- $H_0 : \mu_1 = \mu_2$ (Null hypothesis)
- $H_\alpha : \mu_1 \neq \mu_2$ (Alternative hypothesis)

**Test statistics:**

$$T = \frac{\bar{X}_1 - \bar{X}_2}{S_p \sqrt{\frac{1}{N_1} + \frac{1}{N_2}}}$$

where, $N_1$ and $N_2$ are the sample sizes, $\bar{X}_1$ and $\bar{X}_2$ are the sample means and $s_1^2$ and $s_2^2$ are the sample variances.

where, $S_p^2 = \frac{(N_1 - 1)s_1^2 + (N_2 - 1)s_2^2}{N_1 + N_2 - 2}$

**Critical region:** Reject the null hypothesis that the two means are equal if $T < -t_{(\alpha/2, v)}$ or $T > t_{(\alpha/2, v)}$, where $t_{(\alpha/2, v)}$ [Fish47] is the critical value of the t-distribution with $v = (N_1 + N_2 - 2)$ degrees of freedom.
The global texture descriptor, texspectrum of the reference images \( R_j \) \( 1 \leq j \leq L \) contain important textural information and are used as features of the images during classification. The reference feature vector is measured for each pixel in the test image as follows. A window of larger width is slid over the test image. For each position of the window the texspectrum is calculated from the encompassing image region. Now, the Bartlett's test is applied to compare the variance of the global descriptor of the sample drawn from the window of the test image with the variances of the global descriptor of the reference images \( R_j \), where \( R_j \) is associated with pattern class \( \Pi_i \). If the variances passed the test then the t-test is applied to test the homogeneity of means of the global descriptor of the sample and the reference images. If the hypotheses of both the tests are accepted at a desired level of significance, then it is assumed to be that the samples in the window of test image have come from the corresponding reference image \( R_j \). Then the centre pixel of that window is classified to the corresponding class \( \Pi_i \), \( 1 \leq i \leq L \) and assigned the label \( i \) to that pixel. The same procedure is carried out for the entire test image.

The above discussed mathematical procedure implemented in this thesis work is presented in an algorithmic form, in the following subsection.

**3.4.1.3 Algorithm**

**Input**: A matrix \textit{texspect} contains texspectrum.

**Output**: Classified homogeneous image regions.

*Step 1*: Input target image \textit{texspect} of size 254x254.

*Step 2*: Input \textit{ref_image}, \( R_j \) of size 254x254; \( j = 1, 2, \ldots, L \).
Step 3: Partition the target image into various blocks of size (28×28).

Step 4: Compute Bartlett’s statistic $B$ value using $\text{t sexist}$ and $\text{ref}_\text{image}$ as expressed in sub section 3.4.1.1.

Step 5: If $B < \chi^2_{(a, k-1)}$ then go to Step 6; Otherwise go to Step 2 to take another reference image.

Step 6: Compute t-statistic $T$ value using $\text{t sexist}$ and $\text{ref}_\text{image}$ as expressed in sub section 3.4.1.2.

Step 7: If $T < t_{(a/2, v)}$ then

- it is concluded that the image region of target image is sampled from the corresponding reference image ($R_i$);
- Otherwise go to Step 2 to input another reference image.

Step 8: The tested image region is classified to the corresponding class $[i]$, $1 < i < L$ and assign the label $i$ to the pixel at the centre of the image region.

Step 9: Repeat Step 2 through Step 8 until process the entire target image.

Step 10: Stop.

3.4.2 Unsupervised Classification

In this subsection, usage of the proposed global descriptor, texspectrum is highlighted in unsupervised classification of a target image with the help of modified k-means algorithm. The entire procedure of unsupervised classification is explained in the following subsection.

3.4.2.1 Methodology

Fixing outliers: Generally, the average value of a cluster is sensitive to outliers and it may be misrepresentative of the concerned cluster. The best centre value is actually not the average, but it is the centre value
of the more densely suited points, which is more likely to be the
centre value, for the cluster. The median is better than the average in
this respect, but a combination of average and median is so better. To
determine such a value for a set of points $x_1, x_2, ..., x_n$ in a cluster, we
use the $\alpha$-trimmed means algorithm. Actually, what this algorithm
does is that it removes the largest values and least values and then it
averages the remaining values. To remove the largest and least
values, we use the confidence interval at a specified level of
significance. The values in each cluster are assumed to be
independent and normally distributed. The values that fall outside the
intervals are treated as outliers and they are removed from the
concerned cluster.

**Merging Clusters**: Initially the pixels in the input test image that are
too close to each other are clustered arbitrarily. Now, the $k$-means
type of algorithm is iterated with these plentiful cluster centers. The
problem at this point is that there may be likely too many centres.
The centre of each cluster is tested for closeness of any two clusters.
Closeness is a relative measure and we use the minimum distance
classifier as the measure. The closest two clusters are merged and the
empty cluster is removed from merging. This procedure is continued
until the distance becomes too large between the clusters i.e. all the
clusters become mutually exclusive.

To find the typical distance between these $k$ clusters, we compute
the distance $d[C_i, C_j]$ between the $C_i^{th}$ and $C_j^{th}$ centres over all pairs of
centres. The $\alpha$-trimmed mean of these values is computed to obtain
the typical distance. Next we fix a smallest proportion of the typical
distance to use as the threshold $t$ for testing whether a distance is too close or not.

Here, we have taken the same global descriptor considered for the supervised classification as discussed in subsection 3.4.1 of this chapter. The computed global descriptor of the given test image is divided into various nonoverlapping windows with size $(28 \times 28)$. Each window is assigned a cluster number. First the $\alpha$-trimmed mean is calculated for each cluster. The closest centres of the clusters are identified and are merged as single cluster. Now the empty cluster is removed from merging. This procedure is repeated until all the clusters become mutually exclusive.

The typical modified $k$-means procedure implemented in this thesis work is presented in an algorithmic form, hereunder:

### 3.4.2.2 Algorithm

**Input**: Textured image of size $256 \times 256$.

**Output**: Classified homogeneous regions

**Step 1**: Divide the given test image into various windows $(W_i)$ of size $28 \times 28$.

**Step 2**: Initially, each window is assigned a cluster number $C_i$, $(i = 1, 2, \ldots, L)$.

**Step 3**: Perform Procedure $\alpha_{trimmed\_mean}(\ )$ for each cluster.

**Step 4**: Perform Procedure $\alpha_{trimmed\_mean\_centre\_distance}(\ )$

**Step 5**: Perform Procedure $merge\_cluster(\ )$

**Step 6**: Perform Procedure $\alpha_{trimmed\_mean}(\ )$ for updated cluster centers.

**Step 7**: Perform Step 5 and Step 6 until the least distance is too large to merge the clusters; go to next Step.

**Step 8**: Stop
The following procedure finds $\alpha$-trimmed mean.

**Procedure** alpha_trimmed_mean( )

{  
    **Step 1:** Initialise $y[n]$  
    **Step 2:** Input $x[i]$ ; $i = 1, 2, \ldots, n$  
    **Step 3:** Find $\bar{x} \leftarrow \sum_{i=1}^{n} x[i]$  
    **Step 4:** Find $sd \leftarrow \sum_{i=1}^{n} (x[i] - \bar{x})^2$  
    **Step 5:** $\text{low Lim} \leftarrow \bar{x} - \alpha \cdot (sd/\sqrt{n})$  
    $\text{up Lim} \leftarrow \bar{x} + \alpha \cdot (sd/\sqrt{n})$  
    **Step 6:** $m \leftarrow 1$  
    for $i = 1$ to $n$  
    if $x[i] \le \text{up Lim}$ or $x[i] \ge \text{low Lim}$ then  
    $y[m] \leftarrow x[i]$  
    $m \leftarrow m+1$  
    endif  
    // dense data to  
    // new array $y$  
    **Step 7:** Find $\bar{y} \leftarrow \sum_{i=1}^{n} y[i]$  
    // for remaining  
    // values  
    **Step 8:** Return
}

The following procedure computes the $\alpha$-trimmed mean for centre distances.

**Procedure** alpha_trimmed_mean_centre_distance( )

{  
    **Step 1:** for $C1 = 1$ to $L - 1$  
    for $C2 = 1$ to $L$  
    $d[r] \leftarrow \text{dist}(C1, C2)$ ;  
    // distance between  
    $i1 \leftarrow C1$  
    // clusters $C1$ and $C2$  
    $i2 \leftarrow C2$  
    $R \leftarrow r$
}
The procedure given below performs the operation of merging two clusters that are too close.

Procedure merge_cluster() {
    Step 1: Input threshold $t$
    Step 2: Initialise centre pair index $rr$ with 1
    Step 3: for $r = 2$ to $K(K-1)/2$
            if $d[r] < d[rr]$ then
                $rr ← r$
                $rr ← r$
            end if
        end for
    Step 4: if $d[rr] < t*\text{atave}$ then
                Merge two clusters $C$ and $C2$.
                stop $←$ false
            else
                stop $←$ true;
        end if
    Step 5: Eliminate empty clusters from merging
    Step 6: if (!stop) go to Step 1;
    Step 7: Return
}

After classifying the target image by the proposed method, the misclassification error is evaluated as follows.

\[
\text{misclassification error} = \frac{\text{Number of misclassified pixels}}{\text{Total number of Pixels in that region}}
\]
The mathematical procedures adopted in this Chapter of this thesis work for texture analysis, that is, texture identification, texture representation, and the classification of both supervised and unsupervised based on the representation are given in the algorithmic form as follows:

**Algorithm**

**Input**: Target image and reference images \( R_j; j = 1, 2, ..., L \) of size \( 256 \times 256 \).

**Output**: Classified homogeneous regions.

**Step 1**: Input target image of size \( 256 \times 256 \).

**Step 2**: Compute autocorrelation coefficients \( (\rho_k) \) on target and reference images.

**Step 3**: Compute \( \text{texnum} \) by apply the transformation \((\rho * 100) + 100\) on autocorrelation coefficients computed in **Step 2**.

**Step 4**: Compute \( \text{texspectra} \), by calculating the number occurrences of \( \text{texnum} \).

**Step 5**: Supervised classification is performed by using the algorithm discussed in sub section 3.4.1.3.

**Step 6**: Unsupervised classification is performed based on the algorithm discussed in sub section 3.4.2.2.

**Step 7**: Stop.

3.5 Experiments and Results

The proposed family of Full Range Autoregressive (FRAR) model based texture analysis has been carried out with different 2-D monochrome images. Initially in order to measure the suitability of the proposed model for texture identification, we consider a large
number of different types of hand-coded texture primitives. Few of such texture primitives are shown in Figure 3.1.

![Texture Primitives](image)

Figure 3.1: Texture Primitives

As described in section 3.2, we first compute the model coefficients $\Gamma_s$ from the model parameters $K, \alpha, \theta$ and $\phi$ and then the autocorrelation coefficients $\rho_k$ are computed. They are then subjected to test for homogeneity of variances at a desired level of significance, we compare the said coefficients for rejecting/accepting the null hypothesis as described in section 3.2. With significance in the autocorrelation, we regard the image region under analysis to have texture or not. Such identified textures are then represented as described in section 3.3, as texnum. The texnum values for the hand-coded primitives given in the Figure 3.1 are presented in table 3.1.

<table>
<thead>
<tr>
<th>132</th>
<th>155</th>
<th>174</th>
<th>152</th>
<th>109</th>
<th>141</th>
<th>102</th>
<th>145</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>134</td>
<td>151</td>
<td>148</td>
<td>154</td>
<td>150</td>
<td>128</td>
<td>170</td>
</tr>
<tr>
<td>169</td>
<td>133</td>
<td>138</td>
<td>118</td>
<td>144</td>
<td>164</td>
<td>124</td>
<td>147</td>
</tr>
</tbody>
</table>
In our next experiment, we wish to strengthen the measures of texture, by identifying the untexturedness present in the image. For this purpose, different images, each having the texture and untexture of their combination are considered. One such original image containing both texture and untexture viz. boat image is given in Figure 3.2, whose size is 256×256 with pixel values in the range 0-255.

![Boat Image](image)

Figure 3.2: Boat Image

After computing the autocorrelation coefficient, we use the test of hypothesis. The coefficients that fall inside the confidence limit at 75% level of significance is found to contribute the untexturedness. The output of this experiment carried on the image shown in Figure 3.2 is presented in Figure 3.3.

![Identification of untexturedness because of coefficients that](image)

Figure 3.3: Identification of untexturedness because of coefficients that
Having satisfied with different hand-coded texture primitives and identifying untexturedness, we next experiment for the exact micro texture present in the image. For this purpose different 2-D monochrome images taken from standard Brodatz [Brod66] Album and nature scenes are used in our experiment. Four such images viz. D16, D109, D38 and D93, taken from the standard Brodatz Album are given in Figure 3.4. All these images are of size 256×256 with pixel values in the range 0-255.

As proposed in section 3.2, autocorrelation coefficients are computed for these images by considering small image regions of size
(3×3), after computing the FRAR model coefficients \( \Gamma, s \) from the model parameters \( K, \alpha, \theta \) and \( \phi \). By testing the homogeneity of variances, textures are identified, and are numerically represented as texnum, a local descriptor. By considering sliding window, in the raster scan fashion, the entire image is then subjected to texture analysis and texspectrum as a global descriptor are obtained. Two such texspectra for the images shown in Figure 3.4(c) and 3.4(d) at the 5% and 20% of significance levels are shown in Figure 3.5.
The performance of the proposed FRAR model based local and global descriptors are then highlighted in classification. A target image is formed by considering the above four textured images. This target image is shown in Figure 3.6.
As explained in the previous section the target image is subjected to both supervised and unsupervised classifications. As explained in section 3.4.1, two test statistics, viz. Bartlett’s test statistic and t-statistic are employed to test the homogeneity of variances and means of the target image and the reference images at various significance levels, with different combinations and the misclassification errors are computed.

A window of size (28×28) is considered for this purpose. The classification of 98.46%, 97.38%, 95.23% and 86.12% are obtained in each of the image regions of D16, D109, D38 and D93 respectively. This leads to an average correction of 94.3% with the significance level as 10% in t-test and 25% in the Bartlett’s test. Based on this classification, the target image is segmented, and the segmented image for the target image shown in Figure 3.6 is presented in Figure 3.7.

Figure 3.6: Target image formed by four different images for classification
The proposed FRAR model is also employed on the same target image for unsupervised classification. The modified \textit{k-means} algorithm as explained in section 3.4.2 is used to classify the merged regions of the target image using the proposed global descriptor \textit{texspectrum}. The minimum distance classification is used to classify the homogeneous image region from the target image by using their \textit{texspectra}. The classification of 99.62\%, 98.27\%, 96.81\% and 87.34\% are obtained in each of the image regions of D16, D109, D38 and D93 respectively. This leads to an average correct classification up to 95.51\%. The classified pixels corresponding to various texture classes are grouped to form various segments. The segmented homogeneous image regions are shown in Figure 3.8.
It is evident from the results that the proposed Full Range Autoregressive model based scheme could give most of the pixels of the same class into homogeneous regions.

To examine the classification efficiency of the proposed model, furthermore we considered the combination of different types of textured images in the experiment. Such type of images, namely D16, D109, D4, D38 and D93 are taken from the Brodatz Album. A target image is formed by considering the aforementioned five different textured images and is presented in Figure 3.9.

![Target image formed by five different images for classification](image)

Figure 3.9: Target image formed by five different images for classification

The autocorrelation coefficients are computed on the target image. Based on this autocorrelation, the local and global descriptors are calculated. The global descriptor is utilized for the classification of the target image. First, we performed supervised classification using Bartlett’s statistic and the t-statistic. The classification of 89.68%, 100%, 73.72%, 87.85% and 79.75% are obtained in each of the image regions of D16, D109, D4, D38 and D93 respectively. This leads to an average correct classification up to 86.2% with the significance level at 10% for t-test and 25% for Bartlett’s test. The segmented homogeneous image regions are shown in Figure 3.10.
The modified *k-means* algorithm is employed to classify the same target image shown in Figure 3.9 based on the global descriptor obtained for the target image. The classified pixels corresponding to various textured classes are grouped to form various homogeneous segments. The segmented classes of textured images are shown in Figure 3.11.

The correct classification of 90.68%, 100%, 76.18%, 88.12%, and 80.26% are possible in each of the image regions of D16, D109, D4, D38 and D93 respectively. The average correction of the classification is 87.05%. It is observed from the results that the proposed FRAR model based framework could give good classification results for five different types of images also.
3.6 Conclusion

In this chapter, we have presented, a new statistical approach based on a new family FRAR models and autocorrelation function. The autocorrelation coefficient is derived from the model coefficients. We propose two texture descriptor: (i) texnum, the local descriptor and texspectrum, the global descriptor. We have proposed decimal number to represent the textures that ranges from 0 to 200. These numbers uniquely represent the texture primitives. Totally it has 201 components. Based on this texnum, we have tested good number of images formed with different types of textures. Also, two different sets of textured images are analysed in our scheme. The first one consists of four different types of textured images and the second one is contains five different types of textured images. The proposed scheme results the average classification up to 94.3% and 92.85% for supervised and unsupervised classifications respectively for test image merged with four different types of textured images. The supervised and unsupervised algorithms are employed to classify the test image, which is formed with five different types of textured images, the average correct classification is possible up to 86.2% and 87.05%.