CHAPTER 2

COLOUR TEXTURE ANALYSIS-BRIEF REVIEW

2.1 INTRODUCTION

Texture is one of the vital information for identification and classification of objects. In deterministic texture, the patterns are strictly ordered. In stochastic textures, the spatial distribution of the pattern is random. A micro texture means the texture patterns have sub patterns within themselves. Colour texture analysis is an important and useful area of study in Machine Vision.

There are three major areas of research in textures namely,

- Texture synthesis
- Texture segmentation and classification
- Edge detection

When texture is integrated with colour, identifying and describing characteristics of texture are accelerated, so that, the details of the important features of image objects for human vision can be provided. One crucial distinction between colour and texture features is, that colour is a point, or pixel, property, whereas texture is a local-neighborhood property. Each pixel in an image has a three-dimensional colour vector and different colour space approaches exist to represent colour information. Colour models are discussed in detail in the following sections.
2.1.1 Colour Spaces

Colour is a perceptual phenomenon related to the human response to different wavelengths in the visible electromagnetic spectrum. A small number of basic functions can perform good spectral approximations of most perceivable colours, even though the number of basic functions needed to completely describe the full spectrum is infinite. Generally, a colour is described as a weighted combination of three primary colours that form a natural basis.

2.1.2 Colour Representation

Colour histograms (Flickner et al 1995; Swain and Ballard 1991) are used to represent the colour distribution in an image or a video frame. Mainly, the colour histogram approach counts the number of occurrences of each unique colour on a sample image. Since, an image is composed of pixels and each pixel has a combination of colours, the colour histogram of an image can be computed easily by visiting every pixel once. By examining the colour histogram of an image, the colours existing in the image can be identified with their corresponding areas as the number of pixels. Another possible method is to have a single colour histogram for all the colour channels. In the latter approach, the colour histogram is simply a compact combination of three histograms. The histogram approach is commonly used in most of the existing systems supporting query-by-colour content.

Smith and Chang (1996) proposed Coloursets as an opponent to colour histograms. The coloursets are binary masks on colour histograms and they store the presence of colours as 1 without considering their amounts. For the absent colours, the coloursets store 0 in the corresponding bins. The coloursets reduce the computational complexity of the distance between two
images. Besides, by employing coloursets, region-based colour queries are possible to some extent. On the other hand, processing regions with more than two or three colours is quite complex.

Another image content storage and indexing mechanism is colour correlograms (Huang et al 1997). It involves an easy-to-compute method and includes not only the spatial correlation of colour regions but also the global distribution of local spatial correlation of colours. In fact, a colour correlogram is a table, each row of which is for a specific colour pair of an image. The k-th entry in a row for colour pair (i; j) is the probability of finding a pixel of colour j at a distance k from a pixel of colour i. The method resolves the drawbacks of the pure local and pure global colour indexing methods, since it includes local spatial colour information, as well as, the global distribution of colour information.

There are many colour models currently being used and are discussed below.

2.1.3 RGB Model

RGB is a colour space originated from Cathode Ray Tube (CRT) (or similar) display applications, when it was convenient to describe colour as a combination of three colored rays (red, green and blue). It is one of the most widely used colour spaces for processing and storing of digital image data. However, high correlation between channels, significant perceptual non-uniformity, mixing of chrominance and luminance data make RGB not a very favorable choice for colour analysis and colour based recognition algorithms. This colour space was used in Jones and Rehg (2002).
In Red-Green-Blue Model (RGB), the colour vector of a pixel \( p \) is the compound of red, green and blue channels \( v_p = (r,g,b) \). The primary colours are additive; that is, by varying their combinations, other colours can be obtained (John and Koegel Buford 1994; Foley et al 1990; Hardeberg 1999). The model is visualized as a unit cube (Figure 2.1), with corners of black, white, the three primary colours (red, green, blue), and the three secondary colours (cyan, magenta, yellow).

![Figure 2.1 Colour Cube for Normalized RGB Coordinates](image)

**Figure 2.1 Colour Cube for Normalized RGB Coordinates**

![Figure 2.2 Three layers of Colour image](image)

**Figure 2.2 Three layers of Colour image**

### 2.1.4 Normalized RGB Model

Normalized RGB space is formed independently from varying lighting levels. The red, green and blue components of normalized RGB space
(Figure 2.2) can be obtained from the three components of RGB space using the following formulation equation (2.1).

\[
\begin{align*}
    r &= \frac{R}{R + G + B} \\
    g &= \frac{G}{R + G + B} \\
    b &= \frac{B}{R + G + B}
\end{align*}
\]  

(2.1)

The components of the normalized RGB space are redundant because \(r+g+b=1\). The remaining components are often called "pure colours", for the dependence of \(r\) and \(g\) on the brightness of the source; RGB colour is diminished by the normalization. A remarkable property of this representation is that for matter surfaces, while ignoring ambient light, normalized RGB is invariant (under certain assumptions) to changes of surface orientation relatively to the light source (Skarbek and Koschan 1994). This, together with the transformation simplicity, helped this colour space to gain popularity among the researchers (Zarit et al 1999; Yang and Ahuja 1999; Soriano et al 2000).

2.1.5 CMY Model

The CMY Colour model is based on the secondary colours of the RGB colour space model, that is - cyan (green plus blue), magenta (red plus blue) and yellow (red plus green) (Foley et al 1990). The subset of the cartesian coordinate system for the CMY colour model is similar to that of the RGB colour space, except that the white colour occupies the origin (Figure 2.3). Other colours are obtained by performing either an addition or a subtraction on the white component (Hardeberg 1999). The colour model, however, bears the limitation that each of the three base colours are never available as pure colours, and are always adulterated by a certain proportion of each other. It is, therefore, impossible to create pure black colour using this model. In order to overcome this problem, the colour model has been
extended to form another colour model, referred to as the CMYK colour model, which uses black as the fourth colour.

![Colour Cube for Normalized CMY Co-ordinates](image)

**Figure 2.3 Colour Cube for Normalized CMY Co-ordinates**

### 2.1.6 HSI, HSV, HSL Model

Hue-Saturation based colour spaces were introduced, when there was a need for the user to specify colour properties numerically. They describe colour with intuitive values, based on the artist’s idea of tint, saturation and tone. Hue defines the dominant colour (such as red, green, purple and yellow) of an area; saturation measures the colourfulness of an area in proportion to its brightness. The “intensity”, “lightness” or “value” is related to the colour luminance. The intuitiveness of the colour space components and explicit discrimination between luminance and chrominance properties made these colour spaces popular. The HSV (Hue, Saturation, and Value) colour model (also referred to as HSB model, with B for brightness) is suitably equipped to meet human perception of colour.

Other colour models are illustrated in earlier references (Poynton 1995; Fairchild 1998; Phung et al 2002; Zarit et al 1999; Yang and Ahuja 1999). Several researchers have evaluated different colour models for the
purpose of image retrieval under varying sets of imaging conditions (Gevers and Smeulders 1996). It has been argued that the RGB colour model closely corresponds with the physical sensors of the human eye, although the human perception is more accurately reflected using the HSV colour space (Wesolkowski and Jernigan 1999). The RGB colour space is an additive colour space in which red, green, and blue lights are combined in various ways to create other colours. Nevertheless, the RGB colour space is most frequently used (Vertan and Boujemaa 2000; Sharma and Trusell 1997), and also forms the basis of this research.

2.2 TEXTURE SYNTHESIS

Texture Synthesis is mainly used to improve the realism of graphics. Synthesis is performed directly on the surface. Texture synthesis is the problem of synthesizing a new texture, from a given texture, that, when perceived by a human observer, appears to be generated by the same underlying stochastic process. The synthesis of colour texture is examined, in order to find an optimal set of colour texture parameters describing colour textures. Among many generation methods, for example, structural and reaction-diffusion-like ones, methods considering textures as samples from probabilistic distributions are of increasing interest. By determining the form of these distributions (i.e. the model), textures can be generated. The performance of the methods depends on the structure of the probabilistic density estimator being used. In this context, Markov Random Fields (MRF) (Cross and Jain 1983; Hassner and Sklansky 1980) and Autoregressive models (Sarkar et al 1997) have been successfully used for generation of textures.

The main advantage of texture synthesis, in this case, is that, it can naturally handle boundary condition and avoid verbatim repetitions. In
computer vision, texture synthesis is of interest also, because it provides an empirical way to test texture analysis. Because a synthesis algorithm is usually based on texture analysis, the result justifies effectiveness of the underlying models. Compared to texture classification and segmentation, texture synthesis poses a bigger challenge on texture analysis, because it requires a more detailed texture description and also reproducing textures is generally more difficult than discriminating them.

Other applications of texture synthesis include image editing (Brooks and Dodgson 2002), image completion (Drori et al 2003) and video synthesis (Kwatra et al 2003), etc.

2.2.1 Auto Regressive Model

Autoregressive models correspond to a statistical approach for synthesis of textures. It assumes a local interaction between the image pixels as a weighted sum of neighboring pixel intensities. Assuming the image \( f \) is a zero random field, an autoregressive casual model can be defined in equation (2.2)

\[
f_s = \sum_{r \in N_s} \theta_r f_r + e_s
\]

(2.2)

where \( f_s \) is image intensity at site \( s \), \( e_s \) denotes an independent and identically distributed (i.i.d.) noise and is calculated using equation (2.3), \( N_s \) is a neighborhood of \( s \), and \( \theta_r \) is a vector of model parameters of random field. Using the AR model for image segmentation consists in identifying the model parameters for a given image region and then using the obtained parameter values for texture discrimination. There are unknown model parameters. They
are standard deviation $\sigma$ and model parameter vector $\theta$. The parameters can be estimated through the following equations (2.4 and 2.5).

$$\sum_{s} e_{s}^{2} = \sum_{s} (f_{s} - \hat{\theta} w_{s})^{2}$$  \hspace{1cm} (2.3)

$$\hat{\theta} = \left( \sum_{s} w_{s} w_{s}^{T} \right)^{-1} \left( \sum_{s} w_{s} f_{s} \right)$$  \hspace{1cm} (2.4)

$$\sigma^{2} = N^{-2} \sum_{s} (f_{s} - \hat{\theta} w_{s})^{2}$$  \hspace{1cm} (2.5)

where $w_{s} = \text{col}[f_{i}, i \in N_{s}]$, and the square $N \times N$ image is assumed.

Recursively identified AR model parameters were used (Sukissian et al 1994) for texture segmentation by means of an ANN classifier. Sarkar et al (1997) considered the problem of selecting the AR model for texture segmentation.

### 2.2.2 Markov Random Field Model

A Markov Random Field (MRF) is a probabilistic process in which all interactions are local; the probability that a cell is in a given state is entirely determined by probabilities for states of neighboring cells. Direct interaction occurs only between immediate neighbors. However, global effects can still occur as a result of propagation. The link between the image energy and probability is shown in equation (2.6)

$$p_{\alpha} \exp \left( -\frac{E}{T} \right)$$  \hspace{1cm} (2.6)

where ‘$T$’ is a constant.
There is a potential advantage in Hidden Markov Models (HMM) over other texture discrimination methods, that is, an HMM attempts to discern an underlying fundamental structure of an image, that may not be directly observable. Experiments of texture discrimination using identified HMM parameters are described in Povlow and Dunn (1995), showing better performance than the autocorrelation method, which required much larger neighborhood, on both synthetic and real-world textures.

The MRF was used for colour texture segmentation (Panjwani and Healey 1995). A maximum pseudo likelihood scheme was elaborated for estimation model parameters from texture regions. The final stage of the segmentation algorithm is a merging process that maximizes the conditional likelihood of an image. The problem of selecting neighbors during the design of colour MRF is still to be investigated. Its importance is justified by the fact that, large number of parameters, that can be used to define interactions within and between colour bands, may increase the complexity of the approach. Multiresolution approach to using GMRF for texture segmentation appears more effective compared to single resolution analysis (Krishnamachari and Chellappa 1997).

2.2.3 Fractal Model

A popular area of computer graphics, that does use models of topological texture and does take illumination into account, is that of fractals (Mandelbrot 1985; Voss 1988). Fractal models describe objects that have high degree of irregularity. Statistical model for fractals is fractional Brownian motion (Chen et al 1989; Malerka and Strzelecki 1998). The 2D fractional Brownian motion (fBm) model provides a useful tool to model textured surfaces, whose roughness is scale-invariant. The average power spectrum of
an fBm model follows a $1/f$ law; it is characterized by the self-similarity condition in equation (2.7)

$$\text{Var}[f(t+s) - f(t)] = \sigma^2 |s|^{2H}$$

(2.7)

where $0<H<1$ is known as the Hurst parameter. The major disadvantage of fBm is, that the appearance of its realization is controlled by the single Hurst parameter $H$. Thus, the roughness of the realizations is invariant to scale. Another disadvantage is that the model is isotropic. The Extended Self Similarity (ESS) model was proposed in Kaplan and Kuo (1995) to deal with these limitations are defined in equation (2.8)

$$\text{Var}[f(t+s) - f(t)] = \sigma^2 g(s)$$

(2.8)

where $g(1) = 1$. The function $g(s)$ is called the structure function, which determines the appearance of the 2D random model of a texture. It is related to the image correlation function. For the ESS model, a generalized Hurst parameter is defined for isotropic images.

The ability of fractal features, to segment mosaics of natural texture images, was investigated (Dubuisson and Dubes 1994). It was concluded that fractal dimensions will not segment all types of texture. There were attempts to segment the gray and white matters and lateral ventricles in Magnetic Resonance (MR) images based on fractal models – as reported (Lachmann and Barillot 1992).

To summarize, texture synthesis researchers have explicitly considered and used models. However, as their primary concern is the
appearance of the final image, they have no requirement or motivation to develop mathematical models of the resulting image texture.

2.3 TEXTURE ANALYSIS

Texture and Colour are widely accepted as being two keys in low-level image analysis. The combination of both colour and texture information, have been presented by Caelli and Reye (1993); Scharcanski et al (1994); Kondepudy and Healey (1993). Analysis of texture requires the identification of proper attributes or features that differentiate the textures in the image for segmentation, classification and recognition. There are two alternatives to be used to feature extraction for colour texture analysis. They are

- Processing each colour band separately by applying gray level texture analysis techniques.
- Deriving textural information from luminance plane along with pure chrominance features.

The former approach represents a straightforward method of extending gray level algorithms to colour images and have been used in colour texture segmentation and classification. The latter approach allows a clear separation between texture and colour features. This is useful in segmentation, where gray level algorithms can be applied to luminance plane with colour information used as a cue. The performance of an image analysis system can strongly depend on the choice of the colour representation (Ohta et al 1980). However, this does not appear to be a systematic means of determining an optimum colour-coordinate system for a particular task.

Four major application domains related to texture analysis (Tuceryan and Jain 1993) are texture classification, texture segmentation, shape from texture and texture synthesis. Shape from texture will be discussed in section 2.3.2, texture synthesis has been discussed in section 2.2 and other domains are described briefly below.

### 2.3.1 Segmentation and Classification

Texture segmentation (Derin et al 1984; Mao and Jain 1992) partitions an image into a set of disjoint regions based on texture properties, so that, each region is homogeneous with respect to certain texture characteristics. Results of segmentation can be applied to further image processing and analysis, for instance, to object recognition. Similar to classification, segmentation of texture also involves extracting features and deriving metrics to segregate textures. However, segmentation is generally more difficult than classification, since, boundaries that separate different
texture regions have to be detected in addition to recognizing texture in each region.

Texture segmentation could also be supervised or unsupervised depending on, if prior knowledge about the image or texture class is available. Supervised texture segmentation identifies and separates one or more regions that match texture properties shown in the training textures.

Unsupervised segmentation (Manjunath and Chellappa 1991) has to first recover different texture classes from an image before separating them into regions. Compared to the supervised case, the unsupervised segmentation is more flexible for real world applications, despite that, it is generally more computationally expensive. Xiaoyan Dai, Maeda 2002 presented an approach for colour texture segmentation by using L*a*b* color space as colour feature and adapted statistical geometric features as texture descriptors.

Partitioning an image into homogeneous regions is very useful in a variety of applications of pattern recognition and machine learning. For example, in remote sensing and GIS analysis, texture segmentation could be applied to detect landscape change from an aerial photo, based on their distinct colour texture properties appeared in the image (Yu and Gimel’ Farb 2003). Mirmehdi and Petrou (2000) in present an approach to perceptual segmentation of colour image textures.

Textures can be characterized by two major texture description approaches, such as, statistical approaches (Tuceryan and Jain 1993) and structural approaches apart from other approaches (Haralick 1979). But in the recent literature, many model based methods have been employed in texture analysis (Chellappa et al 1993), including Autoregressive Model (Sarkar et al
Gaussian Markov Random Fields, Gibbs Random Fields, Wavelet Model, Multichannel Gabor model and Steerable Pyramid, etc. These models provide more powerful tools for invariant texture analysis.

Texture classification assigns a given texture to some known texture classes (Varma and Zisserman 2002; Smith and Burns 1997; Chellappa and Chatterjee 1985). Two main classification methods are supervised and unsupervised classification. Supervised classification is provided by examples of each texture class as a training set. A supervised classifier is trained using the set to learn a characterization for each texture class. Unsupervised classification does not require prior knowledge, which is able to automatically discover different classes from input textures. Another class is semi-supervised with only partial prior knowledge available.

The majority of classification methods involve a two-stage process. The first stage is feature extraction, which yields a characterization of each texture class in terms of feature measures. It is important to identify and select distinguishing features that are invariant to irrelevant transformation of the image, such as translation, rotation, and scaling. Ideally, the quantitative measures of selected features should be very close for similar textures. However, it is a difficult problem to design a universally applicable feature extractor, and most present ones are problem dependent and require more or less domain knowledge.

The second stage is classification, in which classifiers are trained to determine the classification for each input texture based on obtained measures of selected features. In this case, a classifier is a function, which takes the selected features as inputs and texture classes as outputs.
Texture classification can sort image data into more readily interpretable information, which is used in a wide range of applications such as industrial inspection, image retrieval, medical imaging and remote sensing.

2.3.1.1 Statistical analysis

Colour is an intrinsic attribute of an image and provides more information than a single intensity value. There have been few attempts to incorporate chrominance information into texture features (Paschos 2000; Mirmehdi and Petrou 2000). Julesz and Bergen (1983) used descriptions such as colour, width, length, and orientations of local features, namely textons, to explain differences in artificially generated images. Statistical methods compute different properties and are suitable, if texture primitive sizes are comparable with pixel sizes. In statistical methods, texture is described by a collection of statistics of selected features.

The techniques used within the family of statistical approaches make use of the intensity values of each pixel in an image, and apply various statistical formulae to the pixels in order to calculate feature descriptors. Texture feature descriptors, extracted through the use of statistical methods, can be classified into two categories according to the order of the statistical function that is utilized: First-Order Texture Features and Second Order Texture Features (Haralick et al 1973).

First Order Texture Features are extracted exclusively from the information provided by the intensity histograms, thus yield no information about the locations of the pixels. Another term used for first-order texture features is Gray Level Distribution Moments. First-order statistics, such as the
mean, standard deviation and higher-order moments of the histogram, concern with properties of individual pixels.

In contrast, Second-Order Texture Features take the specific position of a pixel relative to another into account. The most popularly used second-order methods is the Spatial Gray Level Dependency Matrix (SGLDM) method. The method roughly consists of constructing matrices by counting the number of occurrences of pixel pairs of given intensities at a given displacement. To extract location-based statistical values, it is necessary to devise a means of describing the location of each pixel, and its relative position to pixels of certain intensity more accurately. Spatial Gray Level Dependency Matrix, as called by Haralick et al (1973), is a matrix comparing the intensities of all pixels. Haralick et al (1973) define 14 second-order statistical functions that can be calculated on a SGLDM. Gray Level Co-occurrence Matrices (GLCM), gray level differences (Weszka et al 1976), autocorrelation function, and local binary pattern operator (Ojala et al 1996) are the most popular second-order statistics for texture description.

Higher than second-order statistical features have also been investigated (Galloway 1975; Tsatsanis and Giannakis 1992), but the computational complexity increases exponentially with the order of statistics.

A survey of texture analysis is given in (Wechsler 1982) and a review of texture segmentation and feature extraction is presented in Reed and Du Buf (1993). One of the statistical techniques for texture feature extraction is the co-occurrence approach (Haralick 1979). Several comparative studies have found that the co-occurrence measures are superior to other texture measures in their classification performance (Ohanian and Dubes 1992; Du Buf et al 1990). A fast algorithm for co-occurrence matrix computation was proposed by Argenti et al (1990) and a modification of the
method, that is efficiently applicable to texture description of detected regions, was proposed in Carlson and Ebel (1988), in which, co-occurrence array size varies with the region size. There is a vast forum of work on colour image segmentation (Ohta et al 1980; Healey 1992; Liu and Yang 1994; Skarbek and Koschan 1994; Wain 1990).

In general, most colour texture representation schemes either use a combination of gray level texture features together with pure colour features, or they derive texture features computed separately in each of the three colour spectral channels. Coleman and Andrews (1979) used K-means clustering in each colour band and maximized a cluster fidelity parameter for a more psycho visually acceptable segmented image. Tan and Kittler (1993) used eight DCT texture features computed from the intensity image and six colour features derived from the colour histogram of a textured image for classification. Panjwani and Healey (1993) presented an unsupervised segmentation technique based on Markov Random Fields, which clustered a colour image in the RGB space. The Markov Random Fields approach made use of the spatial interaction of RGB pixels within each colour plane and the interaction between different colour planes. Matas and Kittler (1995) grouped colour pixels by taking into account simultaneously, both their feature space similarity and spatial coherence. Zhang and Wandell (1996) studied systematically the colour perception of human subjects for different frequencies of spatial colour variation. Zhang and Wandell (1996) use the opponent colour space, which consists of three different colour planes, O1, O2 and O3, representing the luminance, the red-green and the blue- yellow planes respectively.

The colour texture information is also obtained via modeling with a class of random field models called the Multispectral Simultaneous Auto Regressive (MSAR) Random field model and the general colour content is characterized by ratios of sample colour means. The retrieval process involves
segmenting the image into regions of uniform colour texture using an unsupervised histogram clustering approach that, utilizes the combination of MSAR and colour features.

Davis and Mitiche (1980) used edge frequencies in texture for texture characterization, in which, edges can be detected either as micro edges, using small edge operator’s masks or as macro edges using larger masks. Texture edges can be detected by using gradient as a function of distance between pixels. Several other texture properties such as coarseness, contrast, randomness from first order statistics of edge distributions and linearity, periodicity and size from second order statistics of edge distributions are used for texture characterization (Tomita and Tsuji 1990).

A zero crossing operator was applied to edge based texture descriptor in which image regions of a constant texture was determined, assuming no prior knowledge about the image texture types or scale (Perry and Lowe 1989). A slightly different approach to texture segmentation may require detection of borders between homogeneous textured image segmentation, which is described in Fan (1989) and a two stage contextual classification and segmentation of textures based on a coarse to fine principle of edge detection is given in Fung et al (1990). A noise tolerant texture classification approach, based on a Canny type edge detector, is discussed in Kjell and Wang (1991), where texture is described using periodicity measures derived from noise insensitive edge detection. A large number of neighboring pixels of the same level represent a coarse texture, a small number of these pixels represent a fine texture and the length of texture primitives in different directions can serve as a texture description (Galloway 1975).

A texture transform, represents another approach, in which, each texture type present in an image is transformed into a unique gray level by
means of some neighborhood processing. If micro textures are analyzed, a small neighborhood must be used and an appropriately large neighborhood should be used for description of macro textures. In addition, a prior knowledge can be used to guide the transformation and subsequent texture recognition and segmentation (Simaan 1990). Linnet and Richardson (1990) used local texture orientation to transform a texture image into a feature image, after which, supervised classification is applied to recognize textures. He and Wang (1990) proposed texture unit and texture spectrum for monochrome texture analysis. In this approach, any small texture unit has been quantified for texture description locally by texture number and globally by texture spectrum. Texture number is obtained by simply comparison of gray levels between central pixel and the neighborhood pixel. The main drawback is, that, it has not considered the levels of comparison like greater or far greater and less or lesser than.

Ojala et al (1996) proposed a simplified version of the Texture Unit descriptor named Local Binary Pattern (LBP). The LBP reduces the range of its possible values to 256. However, this simplification carries with loss of discriminatory power. In Lee et al (1998) introduced the concept of Fuzzy Uncertainty Texture Spectrum, to be used as the texture feature within texture analysis process. Later on, the same authors (Lee et al 1995) proposed a method for texture analysis that is based on genetic algorithms and fuzzy sets. In this method, the local texture feature for a given block is characterized by the corresponding degree to fit to three optimal texture patterns, dynamically generated by genetic algorithms, and the global texture aspect of the image is revealed by its Texture Spectrum.

Taur and Tao (1998) proposed a texture feature based on the fuzzified relative gray levels between pixels. The simplest unit representing a pixel is a 3-vector called Texture Vector. An image block is represented by
the histogram of its Texture Vector, and its texture is, then, characterized by using a neural network classifier. Aina Barcelo et al (2007) proposed a method for evaluating texture features, based on human perception. This method allows a better representation of real textures. Topi Maenpaa and Matti Pietikainem (2006) concluded that adding colour information to texture measure increases accuracy. The above approaches are used only for gray-scaled texture images.

Auto correlation function is one of the spatial frequency methods, in which, texture spatial organization is described by the correlation coefficient that evaluates linear relationship between primitives. Liu and Jernigan (1990) extracted a set of 28 spatial frequency domain features and derived a subset of features in sensitive to noise. Spatial frequency channels and perceptual grouping in texture segmentation was presented in Bek et al (1987).

Laws (1980) has proposed a “Texture energy measure” method, which involves convolving the image with small masks, and then, computing variance like expected values over all neighborhood. The 2-D convolution kernels, typically used for texture discrimination, are generated from the following set of one-dimensional convolution kernels of length five:

\[
L_5 = [ 1 \ 4 \ 6 \ 4 \ 1 ] \\
E_5 = [ -1 \ -2 \ 0 \ 2 \ 1 ] \\
S_5 = [ -1 \ 0 \ 2 \ 0 \ -1 ] \\
W_5 = [ -1 \ 2 \ 0 \ -2 \ 1 ] \\
R_5 = [ 1 \ -4 \ 6 \ -4 \ 1 ]
\]

These measures determine the textural properties by accessing level, edge, spots, ripples and waves in the texture image. It is noted, that, all
kernels except $L_5$ are zero-sum. Using the above five $1 \times 5$ vectors, Laws $5 \times 5$ masks are generated,

$$
L_5^T \times s_5 = \begin{pmatrix}
-1 & 0 & 2 & 0 & -1 \\
-4 & 0 & 8 & 0 & -4 \\
6 & 0 & 12 & 0 & -6 \\
-4 & 0 & 8 & 0 & -4 \\
-1 & 0 & 2 & 0 & -1
\end{pmatrix}
$$

The texture samples are first convolved with one of the masks. Then a measure of the ‘Local Texture Energy’ is computed, which is the average absolute value of convolutions across all neighborhoods. The average absolute values for a set of masks can be jointly used to classify textures. Laws (1979) obtained Classification rates of over 80% in his studies.

Later, Local linear transformations are used to compute texture features (McEvoy and Hartel 1995). The above traditional statistical approaches to texture analysis, such as, auto correlation function, co-occurrence matrices, edge frequencies, texture energy measure, gaussian random fields and local transformations are restricted to the analysis of micro textures only (Chellappa and Chatterjee 1985).

### 2.3.1.2 Structural analysis

The structural approach is based on the theory of formal languages. A textured image is considered as a sentence in a language, of which, the alphabet is a set of texture primitives called textons, constructed in accordance with a certain grammar determining the layout of such texture primitives within a pattern. Structural texture analysis extracts texture
elements in the image, determines their shapes and estimates their placement rules, which corresponds to the global properties of textures. Placement rules describe, how the texture elements are placed, relative to each other on the image and include measures, such as, the number of immediate neighbors, the number of elements in unit space and whether they are laid out homogeneously. Commonly used element properties are average element intensity, area, perimeter, eccentricity, orientation, elongation, magnitude, compactness, Euler number, moments, etc. In an early structural approach (Tuceryan and Jain 1993), for instance, texture elements refer to small elementary regions (windowed sub-pattern) and their placement rules are expressed using a set of tree grammars. A structural method suits better for a description of a macro texture. Structural texture methods have been extended to invariant texture classification.

Structural approaches employ a variety of spatial analytical techniques for detecting the periodicity and analyzing the regularity of textures in order to recover the geometric structure and placement rules of texture elements. For example, Matsuyama et al (1983) detect texture periodicity by finding the peaks from the Fourier spectrum of a textured image. Similarly, Liu and Jernigan (1990) detect the peaks and the output of autocorrelation functions of a texture. Co-occurrence matrices are also used for recovering the periodicity structure from a texture (Parkkinen et al 1990). Besides various geometric techniques, texture elements can also be retrieved based on statistical measures (Ojala et al 1996).

2.3.1.3 Syntactic texture discrimination

Fractal analysis and structural approach are two main methods used to create syntactic models for textures. Syntactic and Hybrid texture description methods are not as widely used as statistical approaches. Syntactic
texture description is based on the texture primitive spatial relations and the structure of a formal language. Syntactic texture models equate tokens of a formal grammar with structural primitives of the texture. A highly structured, but, non-deterministic texture may be generated, if probabilities are assigned to rewrite or expansion rules of the grammar (Lu and Fu 1978).

Shape chain grammars are the simplest grammars that can be used for texture description. They generate textures beginning with a start followed by application of transform rules called shape rules. Characteristic properties of some graphs used practically are described in Urquhart (1982); Ahuja (1982). Hybrid methods of texture description combine the statistical and syntactic approaches; the technology is partly syntactic, because the primitives are exactly defined and partly statistical, because spatial relations between primitives are based on probabilities (Conners and Harlow 1980).

Description of strong textures is based on the spatial relations of texture primitives and bidirectional interactions between primitives. The hybrid multi level texture description and classification method is based on primitive definition and spatial description of inner primitive relations. Another hybrid method uses fourier descriptors for shape coding and a texture is modeled by a reduced set of joint probability distributions obtained by vector quantization.

2.3.1.4 Multi resolution techniques

Texture Description is highly scale dependent. There are several approaches to textures using banks of gabor filters with different Scale and orientation tunning (Porat and Zeevi 1989; Turner 1986). They are used for texture discrimination (Fogel and Sagi 1989). Gabor phase is used as texture features and for texture discrimination (Du Buf 1990). Multi channel gabor
decomposition is used for unsupervised texture image segmentation (Bovik et al 1990). This method requires a large number of 2-D filter operations due to its frequency and orientation selectivity requirements, if an accurate segmentation is desired. The selection of the best filters is based on a priori knowledge of the textural properties, derived from a spectral fourier analysis of the entire input image.

Analysis of linear textures is carried out by Ng and Petrou (1991) and Ng et al (1992), described local linear transform and gabor filter representation of textures. Localized measurement of emergent image frequency by Gabor Wavelets is discussed in Bovik et al (1990). Unsupervised texture segmentation is carried out using N-folded symmetrics by complex moments in Gabor space. Then, a 2- D Gabor elementary functions are used for texture segmentation. Later, Gabor filter is compared with circular merlin function based approaches for texture segmentation (Dunn et al 1995). Dunn and Higgins (1989) used optimal filters for texture segmentation. Texture analysis, using gabor wavelets, is proposed by Naghdy et al (1996). Then, adaptive filters are used for texture segmentation. Optimal gabor filters are designed for texture segmentation using stochastic optimization (Tsai et al 2001). Idrissa and Acherog (2002), performed texture classification using Gabor Filters. The advantage of wavelet transformation over gabor filter is the low pass and high pass filters used in the wavelet transform remain the same between two consecutive scales, while the gabor approach requires filters of different parameters (Chang et al 1993). Gabor filters require proper tuning of filter parameters at different scales. Manian Vidya et al 2000 presented a texture based algorithm is developed for classifying color images. Here the images are filtered by a set of Gabor filters at different scales and orientations. The energy of the filtered images in each channel and between channels are computed and used for classification. The normalized RGB, xyY and HIQ color spaces are used to identify the best
space for classifying the color images. A filter selection process using texture similarity is adopted. A feature reduction process is applied before using a classifier.

2.3.1.5 Wavelet features

Multiresolution representations give rise to an interesting class of texture analysis methods. Strong arguments for their use can be found in psycho visual research, which offers evidence that the human visual system processes images in a multiscale way (Lee 1996). Wavelets provides a convenient way to obtain a multiresolution representation (Daubechies 1992; Mallat 1989), from which, texture features are easily extracted. The energy signatures have proven to be very powerful for texture analysis (Chang et al 1993; Laine and Fan 1993; Unser 1995). The wavelet decomposition of a signal \( f(x) \) is performed by a convolution of the signal with a family of basis functions, \( \psi_{2^s,t}(x) \) and is defined in equation (2.9)

\[
\langle f(x), \psi_{2^s,t}(x) \rangle = \int_{-\infty}^{\infty} f(x) \psi_{2^s,t(x)} dx
\]  

(2.9)

where \( s, t \) are referred to as the translation and dilation parameters, respectively. The wavelet decomposition is obtained with separable filtering along the rows and along the columns of an image. Figure 2.4 illustrates the level 1 and level 2-image decomposition.
The HH subimage represents diagonal details (high frequencies in both directions – the corners), HL gives horizontal high frequencies (vertical edges), LH gives vertical high frequencies (horizontal edges), and the image LL corresponds to the lowest frequencies. Mallat (1989) expressed the similarity between Julez’s “Theory of texons” and wavelet theory. The wavelet representation can also be interpreted as texon decomposition, where each texon is equivalent to a particular function of the wavelet orthogonal basis. Arivazhagan et al 2005 presented an approach for colour texture classification using wavelet domain. They extracted the features of colour information in first order statistics and texture in second order property.

### 2.3.1.6 Moments

Another class of spatial filters is moments, which correspond to filtering the image with a set of spatial masks. The resulting images are, then, used as texture features. Moment-based features are successfully used in texture segmentation by Tuceryan (1992). The Fourier transform describes the global frequency content of an image, without any reference to localization in the spatial domain, which results in poor performance. Spatial dependency is incorporated into the presentation with a window function, resulting in a

---

**Figure 2.4 One-scale decomposition (left), two-scale decomposition (right)**

<table>
<thead>
<tr>
<th>LL</th>
<th>LH</th>
<th>LLLL</th>
<th>LLLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL</td>
<td>HH</td>
<td>LLHL</td>
<td>LLHH</td>
</tr>
<tr>
<td>HH</td>
<td>HH</td>
<td>HL</td>
<td>HH</td>
</tr>
</tbody>
</table>
short-time fourier transform. The squared magnitude of the two-dimensional version of the short-time fourier transform is called a spectrogram, which is used in analyzing shape from texture.

Coroyer Lacos et al (1997) presented a texture classification scheme based on high order statistics, such as, bicorrelation in the spatial domain. Classification rate of 92% has been reported.

2.3.2 Shape from Texture

Shape from texture is the problem of estimating a 3D surface shape by analyzing texture property of a 2D image. Weak homogeneity or isotropy of a texture is likely to provide a shape cue (Clerc 2002). For instance, texture gradient is usually resulted from perspective projection, when the surface is viewed from a slant, which infers the parameters of surface shape or the underlying perspective transformation. Therefore, via a proper measure of texture gradient, a depth map and the object shape could be recovered.

Shapes from texture have been used for recovering true surface orientation, reconstructing surface shape, and inferring the 3D layout of objects, in many applications (Ikeuchi 1984; Leu and Wee 1985; Forsyth 2002). For example, the plane vanish line could be computed from texture deformation in an image (Criminisi and Zisserman 2000), which could be used to affine rectify the image.

Texture segmentation and classification researchers have developed large number of statistical, structural and spectral models to characterize texture effectively. Further, the development of multiresolution technique in this area leads to effective characterization of texture at multiple scales, and thereby, improved the success of texture analysis technique. The third
category, shape from texture yielded some frequency domain model, which allows the effect of illuminant variation on image of topological texture to be predicted.

2.4 TEXTURE RECOGNITION METHOD APPLICATIONS

Analysis of texture finds a variety of applications in the areas of Remote Sensing, Medical Image Analysis, and Automated inspection. Any real world image consists of regions of homogeneous textures. The heterogeneity among these regions is used for classification of categories, such as, Forest, Agriculture land in Satellite images and White matter, Gray matter and Cerebral Spinal Fluid (CSF) in Brain image. The texture features can be used to perform segmentation and classification of medical images.

2.4.1 Inspection

There have been a limited number of applications of texture processing to automated inspection problems. These applications include defect detection in images of textiles and automated inspection of carpet wear and automobile paints.

Dewaele et al (1988) used signal-processing methods to detect point defects and line defects in texture images. They have sparse convolution masks, in which, the bank of filters are adaptively selected, depending upon the image to be analyzed. Texture features are computed from the filtered images. A Mahalanobis distance classifier is used to classify the defective areas. Chetverikov (1988) defined a simple window-differencing operator to the texture features obtained from simple filtering operations. This allows one to detect the boundaries of defects in the texture. Chen and Jain (1988) used a structural approach to defect detection in textured images. They extract a
skeletal structure from images, and by detecting anomalies in certain statistical features in these skeletons; defects in the texture are identified. Conners et al (1983) utilized texture analysis methods to detect defects in lumber wood automatically. Dividing the image into subwindows and classifying each subwindow into one of the defect categories such as knot, decay, mineral streak, etc., perform the defect detection. The features, they use to perform this classification, are based on tonal features such as mean, variance, skewness, and kurtosis of gray levels along with texture features computed from gray level co-occurrence matrices in analyzing pictures of wood. The combination of using tonal features, along with textural features, improves the correct classification rates over using either type of feature alone.

In the area of quality control of textured images, Siew et al (1988) proposed a method for the assessment of carpet wear. They used simple texture features that are computed from second-order gray level dependency statistics and from first-order gray level difference statistics. They showed that the numerical texture features obtained from these techniques can characterize the carpet wear successfully. Jain et al (1990) used the texture features computed from a bank of gabor filters to automatically classify the quality of painted metallic surfaces.

2.4.2 Medical Image Analysis

Image analysis techniques have played an important role in several medical applications. In general, the applications involve the automatic extraction of features from the image, which is then used for a variety of classification tasks, such as, distinguishing normal tissue from abnormal tissue. Depending upon the particular classification task, the extracted features capture morphological properties, colour properties, or certain
textural properties of the image. The textural properties computed are closely related to the application domain to be used. For example, Sutton and Hall (1972) discuss the classification of pulmonary disease using texture features. Some diseases, such as interstitial fibrosis, affect the lungs in such a manner that the resulting changes in the X-ray images are texture changes, as opposed to clearly delineated lesions. In such applications, texture analysis methods are ideally suited for these images. Sutton and Hall (1972) propose the use of three types of texture features to distinguish normal lungs from diseased lungs. These features are computed based on an isotropic contrast measure, a directional contrast measure, and a fourier domain energy sampling. In their classification experiments, the best classification results were obtained using the directional contrast measure. Harms et al (1986) used image texture in combination with colour features to diagnose leukemic malignancy in samples of stained blood cells. They extracted texture micro-edges and “textons” between these micro-edges. The textons were regions with almost uniform colour. They extracted a number of texture features from the textons, including the total number of pixels in the textons, which have a specific colour, the mean texton radius and texton size for each colour and various texton shape features. In combination with colour, the texture features significantly improved the correct classification rate of blood cell types compared to using only colour features.

Landeweerd and Gelsema (1978) extracted various first-order statistics (such as mean gray level in a region) as well as second-order statistics (such as gray level co-occurrence matrices) to differentiate different types of white blood cells. Insana et al (1986) used textural features in ultrasound images to estimate tissue scattering parameters. They made significant use of the knowledge about the physics of the ultrasound imaging process and tissue characteristics to design the texture model. Chen et al (1989) used fractal texture features to classify ultrasound images of livers, and
used the fractal texture features to do edge enhancement in chest X-rays. Texture is represented as an index at each pixel, being the local fractal dimension within an window estimated according to the fractal Brownian motion model proposed by Chen et al (1989). The texture feature is used in addition to a number of other traditional features, including the response to a Kirsch edge operator, the gray level, and the result of temporal operations. The fractal dimension is expected to be higher on an average in blood than in tissue due to the noise and backscatter characteristics of the blood, which is more disordered than that of solid tissue. In addition, the fractal dimension is low at non-random blood/tissue interfaces representing edge information.

Other useful applications of machine vision and image analysis have been in the area of document image analysis and character recognition. They are illustrated in earlier references (Wahl et al 1982; Fletcher and Kasturi 1988).

2.4.3 Remote Sensing

Texture analysis has been extensively used to classify remotely sensed images. Land use classification, where homogeneous regions with different types of terrains (such as wheat, bodies of water, urban regions, etc.) need to be identified, is an important application. Haralick et al (1973) used gray level co-occurrence features to analyze remotely sensed images. They computed gray level co-occurrence matrices for a distance of one with four directions. For a seven-class classification problem, they obtained approximately 80% classification accuracy using texture features. Rignot and Kwok (1990) have analyzed SAR images using texture features computed from gray level co-occurrence matrices. However, they supplement these features with knowledge about the properties of SAR images. For example, image restoration algorithms were used to eliminate the specular noise present
in SAR images in order to improve classification results. Schistad and Jain (1992) analysed, the use of various texture features for analyzing SAR images. SAR images were used to identify land use categories of water, agricultural areas, urban areas, and other areas. Fractal dimension, autoregressive markov random field model, and gray level co-occurrence texture features were used in the classification. The classification errors ranged from 25% for the fractal based models to as low as 6% for the MRF features. Du Buf (1990) used texture features derived from Gabor filters to segment SAR images. He successfully segmented the SAR images into categories of water, new forming ice, older ice, and multi-year ice. Lee and Philpot (1990) also used spectral texture features to segment SAR images.

2.5 CLASSIFIER DESIGN

Classification is a procedure, whose objective is to classify objects into a number of known categories or classes. There are two main approaches for pattern classification, namely, syntactic and statistical. The syntactic classifiers are based on obtaining a grammar or linguistic rule relating certain strings of patterns to each other, whereas, the statistical classifiers evaluate certain statistical measure of patterns or shape. Another category of classifier (Jain et al 2000) is characterized by a geometric approach. Neural network classifiers are able to construct non-linear decision boundaries. The classifiers use training sets of learning and they are termed as supervised classifier Schemes. In unsupervised classification methods, do not need information about the class of objects in the learning stage, but learn them without a teacher.

Each cluster contains patterns representing objects that are similar according to their description and similarity criteria. Objects, that are not
similar, reside in different classes. Popularly used two algorithms are explained below.

2.5.1 K-Means Algorithm

K-Means (MacQueen 1967) one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. Clustering refers to the classification of objects into groups, according to certain properties of these objects. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids. After obtaining these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, it is noticed that the k centroids change their location step by step until no more changes are done. In other words, centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function is defined in equation (2.10)

\[ J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2 \]  

(2.10)

where \( \| x_i^{(j)} - c_j \|^2 \) is a chosen distance measure between a data point \( x_i^{(j)} \) and the cluster centre \( c_j \), is an indicator of the distance of the \( n \) data points from their respective cluster centres.
The algorithm is composed of the following steps:

**Step 1:** Define the number of clusters $k$.

**Step 2:** Initialize (randomly) the clusters prototypes $\mathbf{p}_i$ ($i = 1\ldots k$).

**Step 3:** For each pattern $\mathbf{x}$, assign $\mathbf{x}$ to the nearest cluster $\mathbf{p}_i$ ($i = 1\ldots k$).

**Step 4:** Recomputed $\mathbf{p}_i$ ($i = 1\ldots k$).

**Step 5:** Repeat steps 3 and 4 until the prototypes do not change.

The algorithm has remained extremely popular because it converges extremely quick in practice. In fact, many have observed that the number of iterations is typically much less than the number of points. Recently, however, Arthur and Vassilvitskii showed that there exist certain point sets on which k-means takes superpolynomial time $2^{\Omega(\sqrt{n})}$ to converge.

In terms of performance, the algorithm is not guaranteed to return a global optimum. The quality of the final solution depends largely on the initial set of clusters, and may, in practice, be much poorer than the global optimum. Since the algorithm is extremely fast, a common method is to run the algorithm several times and return the best clustering found.

Another main drawback of the algorithm is, that, it has to be told the number of clusters (i.e. $k$) to find. If the data is not naturally clustered, you get some strange results. Also, the algorithm works well only when spherical clusters are naturally available in data.

### 2.5.2 Fuzzy c-Means Algorithm

Clustering methods aims to find a natural grouping or clusters in multidimensional data based on measured or perceived similarities among the
patterns. A cluster consists of a relatively high density of points separated from other clusters by a low density of points. FCM clustering method partitions the universe $F$ comprised of ‘n’ data samples (feature vectors) into ‘c’ clusters. The data Universe $F$ is denoted as $F = \{ f_1, f_2, f_3, \ldots f_n \}$ . Fuzzy Clustering assigns a membership degree for each of the sample to every cluster. The clustering starts with an initial partition and proceeds as follows (Sonka et al 1999).

i. Assigning each sample to its closest cluster centre generates a new partition.

ii. New Cluster centres are computed as the centroids of the clusters

iii. The above two steps are repeated until an objective function is minimum.

The fuzzy c partition matrix $U$ for grouping a collection of $n$ data sets into $c$ classes, we define an objective function $J_m$ for a fuzzy c-partition is defined as equation (2.11)

$$J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{m} (\mu_{ik})^m d_{ik}^2$$  \hspace{1cm} (2.11)

where $d_{ik}$ is the Euclidean distance of $k^{th}$ sample to the center of $i^{th}$ cluster is defined in equation (2.12).

$$d_{ik} = d(x_k - v_i) = \left[ \sum_{j=1}^{m} (x_{kj} - v_{ij})^2 \right]^{1/2}$$  \hspace{1cm} (2.12)

where, $\mu_{ik}$ is the membership value of the $k^{th}$ data sample in the $i^{th}$ class. It is in between 0 to 1.
‘$m^1$’ is a weighing parameter and is assigned a value ‘2’ in this work. It ranges from 1 to $\alpha$ and it controls the amount of fuzziness in the process.

$V_i$ is $i^{th}$ cluster center, which is described by $m$ features and arranged in vector form as shown in equation (2.13).

$$V_i = \{V_{i1}, V_{i2}, \ldots, V_{im}\}$$ (2.13)

Each of the cluster centre co-ordinates for each class can be calculated as specified in equation (2.14)

$$V_{ij} = \frac{\sum_{k=1}^{n} \mu_{ik}^m x_{ij}}{\sum_{k=1}^{n} \mu_{ik}^m}$$ (2.14)

where $j$ is a variable on the feature space, $j = 1, 2, \ldots, m$.

The optimum fuzzy c-partition will be smallest of the partitions described in equation (2.11), that is given in equation (2.15).

$$J_m^* \left( U^*, v^* \right) = \min_{M, v} J(U, v)$$ (2.15)

An effective algorithm for fuzzy classification, called iterative optimization, was proposed by (James C. Bezdek 1981). The steps in this algorithm are as follows.
Algorithm

1. Fix $c$ ($2 \leq c \leq n$) and select a value for parameter $m$. Initialize partition matrix.

$$U = \begin{bmatrix}
\mu_{11} & \mu_{12} & \mu_{13} & \cdots & \mu_{1n} \\
\mu_{21} & \mu_{22} & \mu_{23} & \cdots & \mu_{2n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mu_{c1} & \mu_{c2} & \mu_{c3} & \cdots & \mu_{cn}
\end{bmatrix}$$

2. Calculate ‘c’ centres $\{v_i^{(r)}\}$ for each step. Each step will be labeled ‘r’ where ‘r’=0,1,2…

3. Update the partition matrix $U^{(r)}$ for the $r^{th}$ step using equation (2.11) and calculated using $\mu_i^{(r+1)}$ equations (2.16 to 2.20).

$$\mu_i^{(r+1)} = \frac{1}{\sum_{k=1}^{m-1} \left[ \frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right]^{\frac{2}{m-1}}} \text{ for } I_k = \phi \quad (2.16)$$

or

$$\mu_i^{(r+1)} = 0 \text{ for all classes } i \text{ where } i \in \tilde{I}_k \quad (2.17)$$

where

$$I_k = \{ i \mid 2 \leq c \leq n; d_{ik}^{(r)} = 0 \} \quad (2.18)$$
and
\[ I_k = \{1, 2, 3, \ldots, c\} - I_k \quad (2.19) \]
and
\[ \sum_{i \in r} \mu_{ik}^{(r+1)} = 1 \quad (2.20) \]

iv. If \( \| U^{(r+1)} - U^{(r)} \| \leq \varepsilon_k \), Stop; otherwise set \( r = r+1 \) and return to step (ii).

In step (iv), we compare a matrix norm \( \| \) of two successive fuzzy partitions to a prescribed level of accuracy, \( \varepsilon_k \) to determine whether the solution is good enough.

In the step (iii), there is considerable amount of logic involved in equations 2.16 to 2.20. Equation 2.16 is straightforward enough, except when the variable \( d_{jk} \) is zero. Since this variable is in the denominator of a fraction, the operation is undefined mathematically, and computer calculations are abruptly halted. So the parameters \( I_k \) and \( \tilde{I}_k \) compromise a bookkeeping system to handle situations when some of the distance measures, \( d_{ij} \), are zero, or extremely small in a computational sense. If a zero value is detected, equation 2.17 sets the membership for that partition value to be zero. Equations 2.18 and 2.19 describe the bookkeeping parameters \( I_k \) and \( \tilde{I}_k \), respectively, for each of the classes. Equation 2.20 says that all the nonzero partition elements in each column of the fuzzy classification partition, \( \tilde{U}_r \), sum to unity. If the difference is within a present value of \( \xi_k \), the convergence is achieved and the resultant partition matrix gives the information about the membership values of the samples to all clusters. Each cluster is assigned
with a pseudo Colour. Ooi, Lim 2006 presented an approach for segmenting colour texture segmentation by integrating Colour texture features using fuzzy c-means clustering algorithm.

In this chapter, an elaborate survey has been presented in texture analysis, synthesis and recognition. Various approaches used for performing texture analysis have been presented. The usage of this review in formulating our problem for texture representation is discussed in the next chapter.