Chapter IV

Texture Characterisation with Orthogonal Polynomials for Image Coding
CHAPTER 4

TEXTURE CHARACTERIZATION WITH ORTHOGONAL POLYNOMIALS FOR IMAGE CODING

4.1 Introduction

In this chapter, we propose a texture preserving image coder using the same set of orthogonal polynomials, presented in chapter 2. The proposed scheme is based on the model that represents textures using point spread operator relating to a linear system. In the proposed texture-based image coding scheme, the encoder first identifies textured regions, which are then analyzed to produce the model features. These are later transmitted to the decoder that produces a synthetic texture based on these features through the synthesis stage. The aim of the texture synthesis is to remove undesirable artifacts in the images obtained after compression-decompression process. We extend our algorithm for this purpose by replacing artifacts in the textured backgrounds. The proposed algorithm is easy to use and requires only a sample texture as input. It generates textures with perceived quality equal to or better than those produced by previous techniques. The key advantage of this approach is that it can efficiently generate high quality textures. A detailed account of how the proposed orthogonal polynomials based framework facilitates the representation and synthesis of textures for image coding is presented in this chapter.

Texture is defined as a structure composed of a large number of more or less ordered, similar elements or patterns. The primitives and their placement rules can characterize observable texture. If the primitives have gray level variation within a small image region it is known as micro texture. Micro texture commonly occurs in most of the textured images originating from natural scenes or from abraded, torn or worn surfaces of many objects. Texture analysis is a long standing and important problem in computer vision. The goal of the techniques presented in the literature for texture analysis is to duplicate the ability of the human brain to understand textural characteristics in an efficient way. Texture analysis comprises of problems like texture identification, texture classification, texture segmentation and shape from texture besides texture synthesis. Texture synthesis is an alternative way to create textures. Because synthesis textures can
be made with any size, visual repetition is avoided. Texture synthesis can also produce tile-able images by properly handling the boundary conditions. Potential applications of texture synthesis are image de-noising, occlusion fill in and compression. Textures contain repeating patterns and high frequency information that are not well compressed by transform coding technique. Compressed images are impaired by various types of artifacts such as blocking, blur, ringing etc. One well-known artifact in block-based coding schemes is the blocking effect, which occurs due to coarse quantization. In other schemes the coarse quantization leads to so-called ringing which causes side-echoes along sharp edges. Finally, there is a blurring effect. Basically, it refers to the high-frequency texture information.

To improve the visual quality of compressed images, there must be an effective method that can recapture the texture lost during encoding process. In this context, visually-lossless compression ratio is used to evaluate the performance of any image compression. This ratio represents quality criteria that significantly depend on the image being compressed, the compression scheme and the final viewing conditions. Compressing images at higher rates may introduce visual impairments in the compressed domain. Because artifacts affecting image texture are immediately perceived by human observers and lossless encoding of texture affects the coding bit rate efficiency, it becomes clear that the textured regions should be encoded separately using specially adapted techniques. Texture-based models characterize the textured regions and it would suffice to put in the image decoder. Then, the decoder uses the transmitted model features to generate a texture visually similar to the original one.

In this work, an image region is represented as a linear combination of responses of the proposed difference operators, developed from the set of orthogonal polynomials. Based on the effect of this operator, micro textures have been identified and then represented as a decimal number. Once the texture is represented in the polynomial domain, the properties of texture can be captured relatively easily and therefore modeled efficiently. The orthogonal effects due to the spatial variations and their corresponding variances are computed, for texture representation, and for subsequent texture preserving image coding.
4.2 Survey of Literatures

Among the existing methods for texture representation [Hara79, Gool83, Tuce88, Hara84], Haralick's co-occurrence matrix method [Hara73], Galloway's run length method [Gall75], Fourier power spectrum method [Wesz76] and texture number method [He90] are well known literatures on the various approaches and models that have been used for texture. These include statistical approaches with autocorrelation function, optical transforms, digital transforms, textural edgeness, structured element, gray tone occurrence, and run length autoregressive models. Over the years, numerous methods have been proposed for the texture analysis. These methods are broadly divided into four categories, namely, statistical methods [Ohan92], structural methods [Petr06], model based methods [Chel85] and filter based (or) signal processing methods [Cogg85]. Filter based methods can be grouped again into three categories (i) spatial domain filtering [Clau05] (ii) frequency domain filtering [Laws80] and (iii) spatial frequency domain filtering [Kim02]. Other works include Wigner distribution [Jack90], Gabor filter [Kama06], wavelet transformation [Do02, Liu01]. It is reported that the problems in Gabor filter and Wigner distribution can be avoided if one uses the wavelet transformation, which provides a precise and unifying framework for the analysis and characterization of a signal at different scales.

Due to the unceasing demand for a larger compression ratio with satisfactory image quality, texture modeling has gained increased interest from researchers in the field of image compression [Reic03, Fahm04, Vanh03, Chan03]. J. Reichel et.al. [Reic03] have compared the texture coding methods with motion prediction for compression purposes within the framework of JPEG2000. A texture characterization scheme has been reported by G. Fahmy in [Fahm04]. B. Vanhoof et al. utilized the wavelet transformation and proposed a visual texture compression system with optimized memory organization in [Vanh03]. Another scheme that takes care of noise environment has been reported by Chan et al. [Chan03] for texture image coding. Ryan [Ryan96] proposes an image coding scheme where the input image is segmented into texture and non-texture regions and operated directly in the wavelet domain and model the texture by an auto-regressive model. Debure and Kubota [Debu98] proposed a scheme for texture
compression based on wavelet transform and the auto regressive texture model. This scheme investigates the influences of the initial condition and the order of an Auto Regressive model on the resulting texture model. Recently, Nadenau et al. [Nade02] proposed a hybrid scheme that encodes the structural image information by conventional wavelet codec and the stochastic texture in model-based manner. In [Mani03, Chau03 and Stoi04], the advantages and disadvantages of texture analysis and synthesis methods are presented. The other methods include the known categories of statistical, structural, model-based methods and categories that include cellular automata methods, multi-resolution/multi-scale, evolutionary methods and neural networks method with the biological basis of each method.

Sarkar et al. [Sark97] proposed a scheme for identifying the appropriate autoregressive components to describe textured regions of digital images by a general class of 2-D AR models. Jeremy S. De Bonet [Jere97] proposed a scheme that provides the multi-resolution sampling procedure for analysis and synthesis for texture images. Nonparametric sampling procedure for texture synthesis has been proposed by Alexei Efro et al. [Alex99]. Zhu et al. [Zhu98] proposed a statistical theory for texture modeling. This theory combines filtering theory and Markov random field modeling through the maximum entropy principle, and interprets and clarifies many previous concepts and methods for texture analysis and synthesis from a unified point of view. Popat et al. [Popa93] presented another probabilistic modeling technique for high dimensional vector sources and its applications to texture synthesis, classification and compression. This technique combines kernel estimation with clustering. Various texture analysis and synthesis techniques, based on pyramid, wavelet coefficient and tree structure vector quantization are presented in [Davi95, Simo98, Liyi00]. In these texture models, it is claimed that the auto-regressive (AR) model can better handle the non-uniform distribution of texture through its initial conditions. Unfortunately, most existing techniques for AR texture models suffer from two major drawbacks: (i) the order of an AR model can not be chosen directly. (ii) Initial condition can take more bit rates. Hence texture synthesis stage is included in some of the decompression process.
Texture synthesis techniques can be classified as either explicit or implicit; an explicit algorithm generates all the texture samples directly while an implicit algorithm answers a query about a particular sample without computing the whole texture. Most existing statistical texture synthesis algorithms [Jere97, Alex99, Zhu98, Popa93, Davi95, Simo98, Liyi00 and Aaro01] are explicit. Since the value of each texture pixel is related to other pixels, most procedural texture synthesis techniques [Davi98, Darw85, Ken85] are implicit since they allow texels to be evaluated independently.

Motivated by these considerations a new texture analysis and synthesis algorithm is proposed for image compression in this chapter. The proposed scheme is based on model that represents textures using points spread operator relating to a linear system. The objective of this work is to clearly rebuild the missing texture during the decompression-decoding process by texture synthesis method from the extracted texture in the compression encoding process. The important steps involved in the work are highlighted below.

- An image region is represented as a linear combination of the responses of the proposed texture model in the presence of additive white Gaussian noise.
- The texture regions are extracted with texture analysis scheme and represent the texture with decimal number.
- The image is compressed with quantization and entropy coding as described in chapter 3.
- A texture synthesis scheme is devised with the proposed decimal number and is utilized during decompression process.

The proposed scheme not only offers a simple computational scheme, but also achieves a high visual quality, as the missing texture is synthesized from the model features that are transmitted.
4.3 Proposed model for texture characterization

The proposed model for texture characterization is based on the statistical design of experiment approach. We consider an \((n \times n)\) image region from the image \(I(x, y)\), where \(x\) and \(y\) are two spatial coordinates, as follows

\[
I(x, y) = g(x, y) + \eta(x, y)
\]  

(4.1)

In equation (4.1), \(g(x, y)\) accounts for the spatial variation owing to texture and \(\eta(x, y)\) is the spatial variation owing to additive noise. In order to measure the spatial variations owing to texture and noise separately, we represent \(I(x, y)\), as shown in equation (4.2), that follows in terms of a set of uncorrelated basis spatial variations.

\[
I^n_{i, j} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \beta_{ij} \phi_{ij}
\]  

(4.2)

where, \([I^n_{ij}]\) is the \((n \times n)\) gray level image matrix, \([O^n_{ij}]\) accounts for the spatial, model variation and \(\beta_{ij}\) is the \((i, j)\) coefficient of variation. \(\beta_{ij}\) is basically the effect of the variation accounted for by \([O^n_{ij}]\) over the image region \(I(x, y)\). We select \([O^n_{ij}]\)s in such a manner that the effects \(\beta_{ij}\)s are orthogonal to each other. Using the statistical design of experiments paradigm, we consider \(I(x, y)\) to be the yields of the experiments with two factors \(x\) and \(y\), each at \(n\) different levels. Two types of spatial variations are considered in this work. In one, one spatial coordinate varies at a time, when the other remains constant. In the other, both the spatial coordinates vary jointly. The orthogonal effects due to the former kind of variation are called the main effects, whereas, the orthogonal effects due to the latter kind of variation are called interaction effects. It has been observed experimentally that the spatial variation that causes the interaction effects are owing to micro texture present in the image region \([I^n_{ij}]\). The other spatial variation are owing to noise present in the image region \([I^n_{ij}]\). Hence, the texture is characterized by the interaction effects. This is because, in presence of micro texture the two factors \(x\) and \(y\) do not operate independently rather the effect of one is dependent on different levels of the other. For computing orthogonal effects, the set of orthogonal polynomials, which
have been presented in chapter 3 has been used. \( [O^\alpha_{ij}] \) in equation 4.2 are \((n \times n)\)
polynomial basis operators and \(\beta_{ij}\)s are orthogonal effects due to spatial variations of
gray levels present in the image region \([I^\alpha_{ij}]\). The spatial variations are modeled by the
polynomial basis operators \([O^\alpha_{ij}]\)s.

4.3.1 Texture detection

Various micro texture regions can be characterized by estimating the orthogonal
effects and their mean squares. In this respect the following conjectures are proposed.

**Texture conjectures** 1 For a texturized region, members of the set of the interaction effect
mean squares do not estimate the same variance.

**Texture conjecture** 2 For a texturized region, some mean squares of the set of main effect
may estimate the same variance.

**Methodology** Let the image under analysis be of size \((256 \times 256)\), \([M]\) be the polynomial
operator of size \((3\times3)\) and \([I]\) be small image region of size \((3\times3)\) extracted from the
image. The \((3\times3)\) polynomial operator \([M]\) is shown in section 2.3. The orthogonal
effects, \(\beta_{ij}\) s are computed as

\[
[\beta_{ij}] = ([M]^t[M])^{-1} ([M]^t[I]) ([M]^t[M])^{-1}
\]  

(4.3)

and the mean square variances, \([Z^2_{ij}]\)s corresponding do the orthogonal effects \(\beta_{ij}\)s are
computed as

\[
[Z^2_{ij}] = ([M]^t[M])^{-1} ([M]^t[I][M]^2) ([M]^t[M])^{-1}
\]  

(4.4)

It may be noted that \(Z_{ij} (= (Z^2_{ij})^{1/2})\) has been described in section 2.6 as the mean squared
amplitude response of the operator \(O^3_{ij}\).

The set \(A=\{Z^2_{01}, Z^2_{02}, Z^2_{10}, Z^2_{20}\}\) are the set of variances due to the main effects and the
set \(B=\{Z^2_{11}, Z^2_{12}, Z^2_{21}, Z^2_{22}\}\) are the set of variances due to the interaction effects. To test
whether a given region belongs to a texturized region, texture conjectures 1 and 2 can be
tested by applying Nair’s [Nair39] criteria for testing the homogeneity among variances. The Nair’s criteria for testing the homogeneity are explained in section 2.7. If all the mean square variances in B corresponding to all the interaction effects estimate the same variance, then one variance at a time is eliminated and the remaining variances are considered to check whether they are not estimating the same variances. In the worst case, at least two variances must be present so that they do not estimate the same variance. Otherwise it is concluded that the region under consideration is not a textured region. Similarly in the case of main effects, if all the variances present in V such that $V \subseteq A$ estimate the same variance, then it means that they are all coming from the same population. Hence, these variances in V can be used to compute the error variance. Once the conjectures are validated, the image region [I] under consideration may be considered to be a textured region.

### 4.3.2 Responses of the basis operators

In 1-D case, for an image sample size n, the set of polynomials \{u_0, u_1, ..., u_{n-1}\} is used to form the 1-D point-spread operator $|M|$. The last (n-1) number of column vectors $|M|$ are the (n-1) basis operators which are symmetric difference operators. The relationship between the mean squared amplitude response of these basis difference operators and the coefficients of expansion during approximation by the proposed polynomial operators is as shown below. The mean squared amplitude response $\xi_j$ per unit length of the $j^{th}$ operator is

$$\xi_j = \frac{\sum_{i=s}^{s+n-1} |I(x)|u_j(x)}{\left(\sum_{i=s}^{s+n-1} u_j^2(x)\right)^{1/2}}, \quad 0 \leq j \leq n-1 \quad (4.5)$$

In the similar way, the coefficients $|\xi|$ of the complete 2-D orthogonal transformation defined by the set of polynomials are mean squared amplitude responses per unit length of the $n^3$ (n=3) basis operators.

The coefficients $|\xi|$ of the complete 2-D orthogonal transformation defined by the set of polynomials are mean squared amplitude responses per unit volume of the $n^3$
(n=3) basis operators. In general, \( |\beta| = (|M| |M|)^{1/2} \) are mean squared amplitude responses of the operators, where

\[
|\xi| = (|M| |M|)^{1/2} |\beta|
\]

(4.6)

The division of each component of \( |\beta| \) by the corresponding element of \( (|M| |M|)^{1/2} \) results in unity noise gain during convolution with each of the 8 finite difference operators \( o_{ij} \). In the presence of random noise only each of the 8 \( |\xi| \) values gives an estimate of the standard deviation of a component and is the same as the standard deviation of the noise.

In the presence of additive noise the proposed polynomials based model for the image \( I \) is

\[
I(x, y) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \beta_{ij} u_i(x) u_j(y) + \eta(x, y)
\]

(4.7)

where the \( \eta \)s are the additive noise which are normally distributed with zero mean and constant variance \( \sigma^2 \) and are uncorrelated.

Since

\[
\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} I_{ij}^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} Z_{ij}^2
\]

(4.8)

where \( Z_{ij} \) is the mean squared amplitude response of the operator \( O_{ij} \), each \( Z_{ij}^2 \) is a \( \chi^2 \sigma^2 \) variate with one degree of freedom. In other words, due to completeness criterion, the local variance of the image region is decomposed into the \( (n^2 - 1) \) number of non-negative quadratic forms \( Z_{ij}^2 \) as follows.

\[
\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (I_{ij} - \beta_{ij})^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} Z_{ij}^2 \text{ at not } (i = j = 0)
\]

(4.9)

where \( \beta_{ij} \) is the local mean \( \frac{1}{n^2} (\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} I_{ij}) \)
Without any ambiguity $Z_{ij}^2$ may be termed as variances corresponding to the responses of the last $(n^2 - 1)$ basis, difference operators.

### 4.3.3 Texture Representation

The identified micro textured regions are required to be represented properly so that it can be used for texture synthesis. A $(3 \times 3)$ image region is considered as a sample for performing the test. The mean square error variance ($msv$) can be computed as follows

$$msv = \sum_{V} \frac{Z_{v}^2}{||V||}$$

(4.10)

where $||V||$ is the cardinality of the set $V$. Each of the variances in $\{A+B-V\}$ is divided by the mean square error variance ($msv$) for computing the signal-to-noise ratio, where $A$ is the set of coefficients contributing towards texture, $B$ accounts for noise and $V$ is the subset of $A$ that account for error within texture coefficients. In case of significant contribution, the pixel in the original textured image whose zonal position corresponds to the zonal position of the variance term corresponding to the interaction effect is represented as 1; otherwise, it is represented as 0. The positions corresponding to the variance terms in $V$ which are used for computing $msv$ are presented as 0s. So there is a mapping from the gray level image into a string of binary digits 0 and 1 as follows:

$$\begin{bmatrix} i_1 & i_2 & i_3 \\ i_4 & i_5 & i_6 \\ i_7 & i_8 & i_9 \end{bmatrix} \Rightarrow \begin{bmatrix} p_0 & p_1 & p_2 \\ p_3 & p_4 & p_5 \\ p_6 & p_7 & p_8 \end{bmatrix}$$

Gray level image  
Texture representation

where $p_n = \{0, 1\}$ and $n = 1, 2, ..., 8$ Now, the encrypted local description of micro texture is quantified as a decimal number called pronum. Pronum is computed as

$$\text{Pronum} = \sum_{n=1}^{8} p_n \times 2^{n-1}$$

(4.11)
The central pixel is of the image under analysis corresponds to this pronum. Subsequent regions in the image are also considered for computing pronums by sliding a window of size (3 x 3) in the raster scan fashion. The number of occurrences of these pronums is called prospectrum. A prospectrum of an image describes the texture present in the image globally. Since the pronum ranges from 0 to 255 there may be totally 256 components in a prospectrum and reflects the histogram of pronums. The entire procedure of computing the pronum and prospectrum of an image is given as an algorithm here. The usage of the prospectrum is highlighted in the case of texture synthesis in the next section.

**Algorithm**

**Input**: Gray level textured image of size ROW X COL. [ ] denotes the matrix and the suffix denotes the elements of the matrix. Let \([M]\) be the (3 x 3) polynomial operator and \([I]\) be a (3 x 3) image region extracted from the image. PROARR holds the pronums obtained from the image. Output prospectrum i.e. frequency of occurrences of pronums.

**BEGIN**

**Step 1**: Compute \([W]=[M]^T[M]\)

**Step 2**: Repeat through Step 15 for \(k=2\) to \(ROW-1\)

**Step 3**: Repeat through Step 14 for \(l=2\) to \(COL-1\)

**Step 4**: Extract a small region \([I]\) from the image centered at \((k,l)\)

**Step 5**: Compute \(\beta^1=[M]^T[I][M]\)

**Step 6**: Compute \(\beta=([M]^T[M])^{-1}([M]^T[I][M])([M]^T[M])^{-1}=[\beta^1]/([W]_{k,l} \times [W]_{i,j})\)

**Step 7**: \([Z^2]=([M]^T[M])^{-1}([M]^T[I][M])\times([M]^T[M])^{-1}=([\beta^1]_{k,l})^2/([W]_{k,l} \times [W]_{i,j})\)

**Step 8**: \(A=\{Z^2_{01}, Z^2_{02}, Z^2_{10}, Z^2_{20}\}\) are variance due to the main effects and \(B=\{Z^2_{11}, Z^2_{12}, Z^2_{21}, Z^2_{22}\}\) are variance due to the interaction effects.
Step 9 : Perform Nair’s test for set A and B (A to be more convergent and B to be more divergent). If the test fails, pronum=-1, indicating there is no texture. Go to Step 14. (While performing Nair’s test, if all the four variances do not pass the test, then eliminate one variance at a time and perform the test again. In worst case, there must be two variances present)

Step 10 : Let set $V \subseteq A$ has variance terms which pass the Nair’s Test and $|V|$ be cardinality of set $V$.

Step 11 : Compute the mean square error variance, $msv = \left( \sum z_{i,v}^2 \right) / |V|$

Step 12 : Perform the variance ratio test (F-ratio test) with numerator as one of the variances from $\{A+B-V\}$ and $msv$ as the denominator against the chosen significance level. If the outcome of the test is positive, i.e., it nullifies the null hypothesis, then the corresponding position $p_i$ of numerator in the image region $[I]$ is marked as 1, else as 0

Step 13 : Compute the pronum for the image region $[I]$ as, $pronum = \sum_{i=1}^{8} p_i \times 2^{i-1}$. $p_i = 1$ if the $i^{th}$ position is 1; otherwise 0.

Step 14 : Store the pronum in PROARR[k][l] and increment l by l

Step 15 : Increment k by 1

Step 16 : Compute the frequency of occurrences of pronum from PROARR. The prospectrum is pronum vs frequency.

END

4.4. Proposed Texture Synthesis Scheme

Normally, the loss of high frequency texture information and quantization in block-based coding schemes lead to an undesirable blurring effect. This blurring effect makes the decompressed image look unnatural due to the fact that the loss of image texture is highly perceptible to the human eye. This has led to the idea that textured regions should be encoded using specially adapted techniques. So the proposed coding scheme describes such regions by a texture model designed with the orthogonal polynomials based transformation. The process of the proposed texture modeling can be
decomposed into analysis and synthesis steps. The analysis step identifies a small set of features to represent the texture information and the synthesis step reconstructs the texture information with the characteristics similar to the original. The encoder performs the analysis of the texture and transmits the features to the decoder which then performs the synthesis.

The proposed coding scheme first makes distinction between texture blocks and non-texture blocks in the image under analysis. The blocks containing textures (especially micro textures) are identified by computing the Pronum of the block as described in section 4.3. Let the transformed coefficients of a texture block identified to contribute significantly towards micro texture be denoted by $\beta t_{ij}$. Only these coefficients of the texture blocks are considered for texture synthesis. At the encoder the proposed scheme then characterizes the local histogram of the transformed coefficient amplitudes. Then the small sets of features representing the micro textures are computed. The features include the standard deviation ($\sigma$), the angle ($\phi$) determined from the shape of the distribution of coefficients; for a Gaussian distribution $\gamma = 2$ and for a Laplacian distribution the value $\gamma = 1$. Besides these two features the maximum amplitude $\alpha$ among $\beta t_{ij}$s is also computed. The resulting texture model features $\sigma$, $\nu$ and $\alpha$ are stored as additional information using a bit stream.

At the decoder, the model-features for each of the texture block are read from the bit stream and are used to instantiate a random noise process with a probability density function that is described by the received model features. The probability density function $p(x)$ is given by

$$p(x) = \left[ \frac{\nu \omega (\nu , \sigma)}{2 \Gamma \left( \frac{1}{\nu} \right)} \right] \exp \left( - \[ \omega (\nu , \sigma) \times |x| \right) \right)$$

with $\omega(\nu, \sigma)$ defined as

$$\omega (\nu , \sigma) = \sigma^{-1} \left[ \frac{\Gamma (\frac{3}{\nu})}{\Gamma (\frac{1}{\nu})} \right]^{\nu - 1}$$
where $\sigma$ is the standard deviation and $\nu$ is the exponent that determines the shape of the General Gaussian Distribution (GGD); and the gamma function is given by

$$\Gamma(x) = \int_0^\infty \exp(-t) t^{x-1} dt$$

(4.13)

This synthesizes the orthogonal polynomials based coefficients that are replaced at positions where no information about the original coefficients is available. Thus, the texture information lost during encoding process is replaced by the synthesized texture.

The algorithm for texture synthesizes of a textured image is given below.

**Algorithm**

**Input**: Original Image $I$, Quality Factor $Q_f$

**BEGIN**

**Step 1**: Partition the original image $I$ into $(N \times N)$ blocks. Apply orthogonal polynomials based transformation to obtain the transformed blocks $\{[\beta_{ij}^*]: 0 \leq i \leq N; 0 \leq j \leq N\}$ and identify the texture blocks as described in Section 4.3.2. Let the set transform coefficients of texture blocks be $T = \{[\beta_{ij}^*]\}$.

**Step 2**: $\{\forall [\beta_{ij}^*] | [\beta_{ij}^*] \in T\}$ identify the coefficients contributing towards texture with pronum. Let these coefficients be $\beta_{ij}^*$. 

**Step 3**: For every texture block in $T$, repeat steps 4 to 8

**Step 4**: Construct histogram with the transformed coefficients $\beta_{ij}^*$. 

**Step 5**: Identify the type of distribution $D$ and assign the value for $\gamma$ as follows

$$\gamma = \begin{cases} 
2 & \text{if } D \text{ is Gaussian} \\
1 & \text{if } D \text{ is Laplacian}
\end{cases}$$

**Step 6**: Find the maximum amplitude $\alpha$ of the coefficients $\beta_{ij}^*$. 

**Step 7**: Compute the standard deviation $\sigma$ of $[\beta_{ij}^*]$. 

**Step 8**: Form the feature set $P = \{[\alpha, \sigma, \nu, k]\}$ where $k$ is the block number.

**Step 9**: Apply quantization with quality factor $Q_f$ and entropy coding on transformed input block to obtain the compressed image $I_c$ as described in chapter 3. Finally send the compressed image $I_c$ along with texture feature set $P$ to the receiver.

61
Step 10: At the receiver, perform entropy decoding and dequantization on $I_c$.

Step 11: Using orthogonal polynomial basis functions get back the transform coefficients.

Step 12: Identify the set of texture blocks $T \{ \forall [\beta'_{ij}] \mid [\beta'_{ij}] \in T \}$

Step 13: Synthesize the orthogonal polynomials based transform coefficients using the feature set of the corresponding blocks with the probability density function as described in equation (4.12)

Step 14: Replace the coefficients in $[\beta'_{ij}]$ with the corresponding synthesized values.

Step 15: Finally the decompressed image using the synthesized values.

Step 16: Reconstruct the image pixels by applying basis functions on the obtained transform coefficients.

END

4.5 Experiments and Results

The proposed orthogonal polynomials based texture preserving compression has been experimented with 2000 test images, having different textual primitives. For illustration two test images viz, Boat and Girl images both of size (256 x 256) with gray scale values in the range 0–255 are shown in figure 4.1(a) and figure 4.1(b) respectively. The input images are partitioned into various non-overlapping sub-images of size (4 x 4), and are subjected to the proposed orthogonal polynomials based transformation to obtain the transform coefficients $\beta'_{ij}$. All these blocks containing $\beta'_{ij}$ are then classified into texture block or non-texture block as described in section 4.3. If the block contains texture then the coefficients contributing towards textures are identified with pronum as described in section 4.3.3. Let these coefficients be $\beta'_{ij}$. Then we construct the histogram with the transformed coefficients $\beta'_{ij}$ and the type of distribution $D$ is computed. Based on the value $D$ the value for $\nu$ is assigned as follows

$$ \nu = \begin{cases} 
2 & \text{if } D \text{ is Gaussian} \\
1 & \text{if } D \text{ is Laplacian}
\end{cases} $$
Maximum amplitude $\alpha$ and the standard deviation $\sigma$ of $[\beta'_{ij}]$ are then computed as described in section 4.4. From these the feature set $P = \{[\alpha, \sigma, \gamma, k]\}$ where $k$ is the block number is formed and it is transmitted to the receiver. The resulting coefficients $\beta'_{ij}$ are scale quantized and re-ordered to 1-D zig-zag sequence. Then these scale quantized 1-D zig-zag sequence is subjected to entropy coding as in JPEG baseline system as described in chapter 2 and then the compressed image is transmitted to the receiver. In the decompression process, compressed image bits are entropy decoded and dequantized to form the 1-D sequence as described in JPEG baseline system. These 1-D transform coefficients are reordered to the original 2-D array. The decompressed original image is then obtained with the orthogonal polynomials basis functions. For each texture block synthesis on the orthogonal polynomials based transform coefficients is applied using the feature set of the corresponding blocks with the probability density function as described in section 4.4. Using this proposed scheme, a compression ratio of 87.31% with a PSNR value of 31.13dB is achieved when the quality factor is 5 for Boat image.

Figure 4.1: Original images considered for texture preserving transform coding

(a) Boat  
(b) Girl
The corresponding Boat image result, after decompression process with orthogonal polynomials basis function and texture synthesis, as described in section 4.4, is shown in figure 4.2(a). For the same quality factor, a compression ratio of 89.92% with a PSNR value of 32.46dB is obtained for the Girl image and resulting output is shown in figure 4.2(b). Experiments are conducted on the proposed orthogonal polynomials based transform coding with texture characterization scheme on relaxed quality factor. For a quality factor 20, a compression ratio of 93.89% with PSNR value 27.03dB is achieved for the Boat image and compression ratio of 93.21% with a PSNR value of 25.31dB for the Girl image. These images are shown in figure 4.3(a) and figure 4.3(b) respectively. The experiment is repeated for the quality factors 1, 5, 10, 15 and 20 and the results obtained by the proposed coding scheme are presented in table 4.1. It can be observed from table 3.1 in chapter 3 and table 2.1 in chapter 2, the orthogonal polynomials based texture preserving coding technique gives better PSNR value than the orthogonal polynomials based transform coding. This scheme is also experimented with different texture images and here we present the results of two standard texture images. These original images namely D96 and D38 both of size (256 x 256) with pixel values in the range 0–255 are shown in figure 4.4. A compression ratio of 89.31% with a PSNR value of 31.93dB is achieved when the quality factor is 5 for D96 image. The corresponding D96 image result, after decompression process with orthogonal polynomials basis function and texture synthesis, as described in section 4.4 is shown in figure 4.5(a). For the same quality factor, a compression ratio of 86.11% with a PSNR value of 33.26dB is obtained for the D38 image and resulting output is shown in figure 4.5(b). The experiment is repeated for the quality factors 1, 5, 10, 15 and 20 and the results obtained by the proposed coding scheme are incorporated in table 4.1.
Figure 4.2: Results of the proposed scheme when the Quality Factor is 5

(a) Boat
(b) Girl

Figure 4.3: Results of the proposed scheme when Quality Factor is 20

(a) Boat
(b) Girl
Table 4.1: Compression Ratio (CR) and PSNR values obtained by Proposed scheme for different Quality Factors

<table>
<thead>
<tr>
<th>QF</th>
<th>Boat</th>
<th>Girl</th>
<th>D96</th>
<th>D38</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CR%</td>
<td>PSNR(dB)</td>
<td>CR%</td>
<td>PSNR(dB)</td>
</tr>
<tr>
<td>1</td>
<td>67</td>
<td>38.24</td>
<td>70</td>
<td>37.96</td>
</tr>
<tr>
<td>5</td>
<td>86</td>
<td>32.07</td>
<td>84</td>
<td>32.46</td>
</tr>
<tr>
<td>10</td>
<td>89</td>
<td>28.49</td>
<td>88</td>
<td>29.47</td>
</tr>
<tr>
<td>15</td>
<td>91</td>
<td>27.82</td>
<td>89</td>
<td>27.06</td>
</tr>
<tr>
<td>20</td>
<td>92</td>
<td>27.03</td>
<td>90</td>
<td>25.31</td>
</tr>
</tbody>
</table>

(a) D96  
(b) D38

Figure 4.4: Original texture images considered for texture preserving transform coding
The proposed texture preserving transform coding is compared with the DCT based JPEG. Here, the transform coefficients obtained after the proposed transformation with different quality factors and the bit allocated using variable length code. For the D38 image with quality factor of 5, the DCT based scheme gives a compression of 88% with PSNR value of 31.22 and for a quality factor of 20, a compression of 92.61% with PSNR value of 25.93 is obtained. With regard to D96 image, a compression of 88.42% with PSNR value of 31.34 is achieved when the quality factor of 5. When the quality factor is relaxed to 20, a compression of 92.89% with the PSNR value 25.99 is obtained and the DCT based resulting output in figure 4.6.

For the quality factor of 5, on the same D38 image, the wavelet-based scheme gives a compression of 86% with a PSNR 32.76 dB and for a quality factor of 20; a compression of 92% with a PSNR value of 25.12 dB is resulted. For the quality factor of 5, on the same D96 image, the wavelet-based scheme gives a compression of 85.11% with a PSNR 32.53 dB and for a quality factor of 20; a compression of 91.34% with a PSNR value of 24.48 dB is resulted.
The corresponding wavelet based reconstructed images are shown in figure 4.7 and the results are tabulated in the table 4.2. It is evident from the table 4.2 that the proposed texture preserving transform coding is giving better compression ratio than the DCT and wavelet based coding results.

Table 4.2: Compression Ratio (CR) and PSNR values obtained by Proposed scheme, DCT and Wavelet based scheme

<table>
<thead>
<tr>
<th>Image</th>
<th>Quality Factor</th>
<th>Proposed Transform Coding</th>
<th>DCT based JPEG</th>
<th>Wavelet based coding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CR (%)</td>
<td>PSNR (dB)</td>
<td>CR (%)</td>
</tr>
<tr>
<td>D96</td>
<td>5</td>
<td>89</td>
<td>31.93</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>91</td>
<td>28.23</td>
<td>90.42</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>93</td>
<td>27.02</td>
<td>92.56</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>93</td>
<td>26.73</td>
<td>92.61</td>
</tr>
<tr>
<td>D38</td>
<td>5</td>
<td>86</td>
<td>32.76</td>
<td>85.01</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>90</td>
<td>29.11</td>
<td>89.55</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>91</td>
<td>26.86</td>
<td>90.11</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>92</td>
<td>25.12</td>
<td>91.23</td>
</tr>
</tbody>
</table>

Figure 4.6: Reconstructed images at quality factor 5 by DCT based scheme
Figure 4.7: Reconstructed images at quality factor 5 by Wavelet based scheme

4.6 Conclusion

A new texture preserving image coder using orthogonal polynomial has been presented in this chapter. The proposed scheme is based on the model that represents textures using point spread operator relating to linear system. The encoder first identifies textured regions, and these regions are analyzed to produce the model features. These features are transmitted to the decoder that produces a synthetic texture based on these features through the synthesis stage. The texture synthesis is used to remove the undesirable artifacts in image obtained after compression-decompression process. The key advantage of this approach is that it can efficiently generate high quality textures. Even though the proposed texture characterization with orthogonal polynomials could work well for texture images, the artifacts due to edges remain unsolved. Hence in the next chapter a new edge preserving transform coding is proposed with the same orthogonal polynomials.