2007

Comparison Of Sparse Coding And Jpeg Coding Schemes For Blurred Retinal Images.

Balaji Chandrasekaran
University of Central Florida

Comparison of Sparse coding and JPEG coding schemes for blurred retinal images

by

Balaji Chandrasekaran
B. E. Anna University, 2005

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the School Of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2007

ABSTRACT

Overcomplete representations are currently one of the highly researched areas especially in the field of signal processing due to their strong potential to generate sparse representation of signals. Sparse representation implies that given signal can be represented with components that are only rarely significantly active. It has been strongly argued that the mammalian visual system is highly related towards sparse and overcomplete representations. The primary visual cortex has overcomplete responses in representing an input signal which leads to the use of sparse neuronal activity for further processing. This work investigates the sparse coding with an overcomplete basis set representation which is believed to be the strategy employed by the mammalian visual system for efficient coding of natural images.

This work analyzes the Sparse Code Learning algorithm in which the given image is represented by means of linear superposition of sparse statistically independent events on a set of overcomplete basis functions. This algorithm trains and adapts the overcomplete basis functions such as to represent any given image in terms of sparse structures. The second part of the work analyzes an inhibition based sparse coding model in which the Gabor based overcomplete representations are used to represent the image. It then applies an iterative inhibition algorithm based on competition between neighboring transform coefficients to select subset of Gabor functions such as to represent the given image with sparse set of coefficients.

This work applies the developed models for the image compression applications and tests the achievable levels of compression of it. The research towards these areas so far

proves that sparse coding algorithms are inefficient in representing high frequency sharp image features. So this work analyzes the performance of these algorithms only on the natural images which does not have sharp features and compares the compression results with the current industrial standard coding schemes such as JPEG and JPEG 2000. It also models the characteristics of an image falling on the retina after the distortion effects of the eye and then applies the developed algorithms towards these images and tests compression results.

To my parents Sekar and Viji

ACKNOWLEDGMENTS

I thank my advisor, Dr. Lei Wei for his valuable guidance and support. I also thank the members of my thesis committee, Dr. Sundaram and Dr. Kasparis Takis. I acknowledge the support that Libo Yang, Harish Ramakrishnan, Prabhakar Mohan, Kaushik, Karthik, Bharath and Giridhar has given me. I also thank the FEEDS department for supporting me in all the hard times.

I thank my parents and Prathibha for having played such a huge part in getting me this far. I acknowledge the immense part Prashanth Chandran, Shrinaresh Subramaniyan, Manikandan and our beloved group in all my success.

I thank Mathworks Inc. and the LaTeXcommunity for reducing greatly the labor involved in writing this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AWGN – Additive White Gaussian Noise

BER – Bit Error Rate

BPP – Bits per Pixel

CG – Conjugate Gradient

DCT – Discrete Cosine transform

DFT – Discrete Fourier Transform

DWT – Discrete Wavelet Transform

FCC – Federal Communications Commission

FDCT – Forward Discrete Cosine Transform

FFT – Fast Fourier Transform

ICA – Independent Component Analysis

IDCT – Inverse Discrete Cosine Transform

IDFT – Inverse Discrete Fourier Transform

IDWT – Inverse Discrete Wavelet Transform

IFFT – Inverse Fast Fourier Transform

IID – Independent Identically Distributed

ISO – International Standards Organization

JPEG – Joint Photographic Experts Group

LDPC – Low-Density Parity-Check codes

LSF – Line Spread Function

LLR – Log-Likelihood Ratio

MAP – Maximum A Posteriori

ML – Maximum Likelihood

MRE – Minimum Reconstruction Error

MoF – Method of Frames

PCA – Principal Component Analysis

PDF – Probability Density Function

PSNR – Peak Signal to Noise Ratio

PSF – Point Spread Function

RMS – Root Mean Square

SCL – Sparse Code Learning

SNR – Signal-to-Noise Ratio

SPC – Single Parity-Check codes

SVD Singular Value Decomposition

TQ – Trellis Quantization

CHAPTER 1: INTRODUCTION

1.1 Motivation for the thesis

Claude E. Shanon in 1948 developed the mathematical foundation of information theory, which establishes the limits or bounds called channel capacity for the maximum reliable information transfer that can be possible in a noisy channel [51]. Since then researches in the field of communications are directed towards building coding schemes that can achieve the Shanon limit. In the early 1960's LDPC (Low Density Parity Check) codes were discovered by Gallager [52]. But they were almost forgotten until recently revived by the discovery of turbo codes. It has been proved that this LDPC codes can achieve coding performance which are just 0.0045 dB away from the Shanon limit [53].

The remarkable feature of the LDPC codes are its sparse connectivity. The codes were defined to be low density since they are defined based on sparse parity check matrix [52]. In the field of communications it has been already proved that these LDPC codes achieves optimal coding performance [53]. So research is directed towards searching whether the sparse structures or connectivity exists in any other fields ? If so they exist, questions are raised whether they provides the optimal performance in the respective fields.

It is quiet remarkable that human's neural system has this kind of sparse connectivity in upper layers of the brain [3]. So this raises the question, whether this connectivity achieves optimal way of representing the input information to the brain. There are significant developments to prove that the human vision system has very close linkage to these

sparse coding structures [3][4]. This forms the primary motivation for this thesis to analyze the sparse codings.

In this thesis we try to study the sparse coding structures that are needed to represent an image to the brain by the vision system. We tried to answer the question whether these sparse codings are optimal way of encoding images that a typical human vision system can encounter. We try to apply the sparse coding models and compare the compression results of it with the existing image compression standards such as JPEG and JPEG 2000.

1.2 Organization of the thesis

In this thesis we aim to compare the sparse coding techniques with the standard JPEG algorithms for blurred retinal images. It has been known that the quality of retinal image is much poorer due to the optical imperfections of the eye and also eye motions. This thesis aims to answer the question whether sparse coding is near optimal way of coding for blurred retinal images.

JPEG standards including JPEG and JPEG 2000 are the best known practical image compression algorithms. Sparse coding has been known as the best possible technique adopted in the human vision system to encode retinal images. In his previous work Bruno showed that with clear images the sparse coding technique performs closely to JPEG but not JPEG 2000. So we further extend his work in this thesis by comparing the sparse coding algorithm for blurred retinal images.

The thesis is organized as follows. In chapter 3 we discuss the statistical properties of natural images which has close relationship to the coding scheme. This chapter also discusses

some of the mathematical models that simulates similar kind of images falling on the retina of the eye after different impairments in the optical part of the eye. It also discusses the filters such as whitening filter and the effect of point spread functions to simulate the similar kind of retinal images.

Chapter 4 overviews the current image compression standards JPEG and JPEG 2000. The compression results of applying these compression techniques for the case of natural images and also for the retinal images are provided.

Chapter 5 and 6 forms the core of this thesis. In chapter 5, SCL (Sparse Code Learning) algorithm based on the work by Bruno is investigated in detail. This chapter is divided into two halves: 1) The first half discusses the training methods to extract the base functions by learning algorithms. 2) The second half deals in application of the extracted base functions to represent any given image by sparsely distributed coefficients and also the compression levels that can be achieved by coding these coefficients are analyzed. The compression achieved by coding in the above mentioned way is analyzed for both natural and retinal images.

Chapter 6 discuss the inhibition based sparse coding algorithm in detail. It also discusses the Gabor and log Gabor filters and its design issues in detail. This chapter also discusses the inhibition based sparse coding algorithm which is one form of greedy based sparse coding technique. The compression results achieved by applying the inhibition algorithm for natural images and retinal images are produced.

Chapter 7 compares the compression results of the JPEG, SCL, JPEG2000 and inhibition algorithms. It also discusses the problems in the sparse coding techniques and also throws light on the future works required to solve the current problems.

1.3 Contributions

The important contributions from this thesis are discussed below:

1) Detailed comparison of performance of JPEG, JPEG 2000 and sparse coding algorithms like SCL and inhibition algorithm for natural and blurred whitened and normalized images.

2) Detailed comparison of sparse coding schemes in terms of base functions for natural, blurred, whitened and normalized images.

3) Implementation of inhibition based sparse coding algorithms and testing the compression results for natural and blurred whitened and normalized images.

CHAPTER 2: BACKGROUND LITERATURE

Barlow's theory states that "The goal of natural vision is to represent the information in natural environment with minimal redundancy"[1].

The biological motivation behind the theory of efficient sensory coding is the assumption of economy in sensory processing [1][3][4]. It has been proved that if any biological organism is confronted with sensory inputs of certain properties, evolution and learning will adjust the organism to these properties such that the efficiency of sensory processing is increased [1][5]. The efficiency in visual neuronal coding means that the neurons becomes sensitive to individual elements that constitute an image. This implies that the receptive fields of the neurons should then correspond to the statistically independent structural primitives of natural images. If the number of structural primitives in a patch of natural images are typically small, this reflects the sparseness structure in neurons that is only few neurons are active at any given instance [3]. Thus the developmental ideas to prove that our human vision has close linkage to sparse coding structures forms the motivational idea behind this thesis.

Over the past four decades a number of attempts have been made to explain the behavior of cortical neurons. The evidence that the cortical neurons are selective to spatial frequency as well as orientation directed a number of researchers to suggest that such neurons must be producing something like Fourier transform and are not performing any feature detection [6][7]. Here features were assumed to be edges, bars and corners of an image [6]. But the notion that the visual cortex performs a global Fourier transform is no longer given serious considerations since the relatively broad spatial-frequency bandwidths and local

spatial properties of cortical neurons make them totally unsuitable for extracting Fourier coefficients. Although this lead to some interesting researches in feature or edge detection due to its effective means to code different images, there is no clear evidence to prove that cortical neurons can be classified as edge detectors. Also the frequency selectivity of the cortical cells are identified to be in opposite to the notion of feature detection since there is less connectivity between frequency selectivity and natural environment.

Field strongly argued that for any code to be efficient, they should be defined only in relation to the class of images that it is most likely to encounter [1]. Since human vision system is most likely to encounter natural images, any coding scheme defined must be related closely to the statistics of natural images. Field also proved in his work that the response properties of cortical neurons are very well suited for the statistics of natural images [1]. Also the frequency selectivity allows the images to be represented by a few active cells. There were arguments developing that the sensors with the tuning of cortical provides the high probability of giving a large response or in worst case no response at all. But these proposals are doubtful to fully claim that these behaviors are related to feature detection. These works did not analyze the statistical behavior of the environment and its relation to the biological significance. So these proposals failed for further developments.

Field [1] proposed a new approach in coding scheme based on Gabor's theory of communication [34] which showed the method to represent time-varying signals in terms of functions that are localized in both time and frequency. These functions referred to as Gabor functions have been used to describe the cortical simple cells behavior that extend in space and time. Field's work suggested that the code he proposed allows on the average the most

information to be represented with small number of active cells. Here information is defined with respect to the variability of the images and not to any specific feature of the image. This description of the information can be directly related to the redundancy of the images.

Barlow [2] in his work provided a thorough discussion of the relations between the redundancy and visual codes. Most of the work developed by Field are based on Barlow's theories of redundancy.

2.1 Barlow's redundancy theories

The researches were directed to determine the efficient code for representing visual information. The analysis was carried out with the base work developed by Barlow's redundancy theory.

The redundancy in the set of images are defined in terms of n-th order conditional probabilities of the coefficients [8]. Consider an array of pixels with a range of possible intensity values. First order statistics relate to the probability of individual pixels. (Eg: pixel i taking on particular intensity level $(m) - p(m_i)$). There is redundancy in the first order statistics when the distribution of intensities is not uniform. A non uniform distribution implies that there is some degree of predictability or order in the intensity values. Second order statistics refer to the conditional probabilities of pairs of pixels. It is a measure of the probability that a pixel will take on a particular value given the values of another pixel, $p(m_i|n_j)$. Most of the redundancies analyzed only consider the second order with power-spectrum and autocorrelation function. However higher order redundancy (eg:third order) may provide significant contribution to the total redundancy of an image. Consider an image

consisting of line segments of random orientation, here if there are two neighboring pixels with same intensity then there will be high probability of a third point with same intensity along the line. This correlation can be well defined only by third order statistics. The power spectrum and autocorrelation will not provide any information regarding the third order statistics. So the aim of the developed code must represent all forms of redundancy [2]. Barlow suggested that the ultimate aim is to reduce the redundancy of the code.

Field [1] discussed in his work the effective ways of coding visual information using Gabor codes and much developed version of log Gabor codes. These models (Gabor and log Gabor) however are not aimed to reduce the redundancy of the code [34][35]. The total number of sensors in representing visual information with these codes remained fixed and the amount of information represented by the code is constant (for a complete code). But these codes provided means to convert the higher-order redundancy (correlation between pair of pixels, triplets of pixels etc) into first order redundancy redundancy (i.e. the response distribution of the sensors) [9]. Theoretically, the redundancy remains unchanged such that the visual information is preserved. Field proved that the most efficient way of coding is the code with most redundant first-order statistics [1]. The next stage of processing can make use of the first-order statistics by coding only the non-redundant elements (i.e. highly active neurons). So it is next stage of processing that has the potential for removing redundancy.

2.1.1 Overcomplete representation and sparse codes

Filed [1] in his inspirational work proposed the Gabor and log Gabor code models to encode the visual information and proved that this way of coding not resembling Fourier

transforms achieves the best efficiency possible for representing the information. The goal of his code is not to discard any type of channel or sensor. The coefficients are here matched to the energy of the image so that on the average any particular coefficient is just as likely to be active as the other and also no particular choice is favored. Hence there can be no chance of reduction in the number of free parameters. For any particular image only a subset of coefficients are active and since the particular active members will vary from one image to other it is not possible to eliminate any particular type of coefficient [4]. So this code developed by him represents a good method for transforming the higher-order redundancy to first order redundancy. Apart from the redundancy discussed that analyzes correlation and statistics of input, there is one more kind of useful redundancy called over-representing the information in the stimulus by using more free parameters than are required [6]. This forms the base idea to represent the visual information using overcomplete basis sets. However Field's work proposed only complete code, where the dimensionality or redundancy in the codes will be equal to the input information. But later developments extended his work towards overcomplete representations using the same set of log Gabor or Gabor kind of codes as developed by Field [1]. He proposed in his work that the redundancy infested by the code can be utilized in later stages to produce a non redundant representation [27]-[29]. If the later stages analyze the outputs of only very active cells (i.e. impose a threshold or inhibition stages) then redundancy can be reduced. This simple-cell behavior model as to represent the visual information with over complete representation set and then use the later stage processing to reduce the induced redundancy forms the core model of research towards sparse coding [7][8]. There are innumerable ways of developing a overcomplete representation and there is still no claims as to define the kind of representations happening in visual system.

However the codes considered such as Gabor and log Gabor have very close resemblance to cortical cell response properties and also forms the base work for most part of the literatures developed that analyzes the ways of reducing the redundancy [4].

This leads to the conclusion that overcomplete representation and sparse coding are the efficient means of coding visual information. In researches so far only two kinds of code representations involving Gabor and log Gabor codes are analyzed because of their close proximities towards cortical cell responses and no other better codes were framed so far [9]. In relevance to the above developed theories many researches were directed towards proving that human vision system is indeed sparse coding. But still there are many issues remain unresolved by this model. The following section takes a brief look on the sparse coding models proposed so far in the literature. To illustrates these model, the coding terminologies must be defined and made common among them and these terminologies are identified in the next section [9].

2.2 Generative model or image coding framework

An image which consists of m pixels can be modeled as a multivariate random signal x ϵR^m. This image can be generated as a linear combination of n independent random sources s ϵR^n according to:

$$x = As + v \tag{2.1}$$

where the matrix A ϵ $R^{m \times n}$ is called the mixing matrix and v ϵ R^m can be considered as iid (independent, identically distributed) Gaussian noise.

The coding formulation can be expressed such as to encode the image **x** into a linear expan-

sion c ϵ R^n such that

$$x = Bc + r \tag{2.2}$$

where the columns of B are called code vectors and the residual r is small relative to the image. The coefficients obtained should be suitable for efficient compression. The coefficients should be statistically independent to achieve efficient compression. The problem then breaks down to just the source estimation if the probability density function of the natural images can be modeled. Even if the statistical properties are properly modeled, there can be infinite possible linear model that has statistically independent sources which could predict or match the observed image density. Now the problem of choosing a model arises. The model that achieves higher compression rates or the model that can show close approximation to human vision system can be chosen. So the criterion of choosing a model depends on the application and research interest of different areas.

The sparse coding algorithms that are applied for the image compression follows the same terminologies. The term coefficients describes the source estimates and residual r describes the noise estimation in the model and the estimation of mixing matrix constitutes learning. There are number of literatures worked on learning as well as source estimation process. This literature review briefly analyzes the existing linear models and also throws light on the constraints and problems of the model assumed.

The sparse coding methods developed falls into two lines of research: sparse code learning and matching pursuit.

2.3 Sparse code learning models

2.3.1 Bayesian formulation

This model [10] forms the base model of sparse code learning that is expanded well in chapter 5. Here the noise and the source are modeled with some prior densities, then the density of the image can be obtained by suing simple Bayes rule:

$$P(s|x) = \frac{P(x|s)P(s)}{P(x)} \tag{2.3}$$

This can be better analyzed by representing this in terms of entropy relation given by:

$$H(x) = -log(P(x)) \tag{2.4}$$

$$H(s|x) = -log(P(s|x)) \tag{2.5}$$

From the above equation the problem now reduces to estimation of criterion MAP (Maximum Aposteriori) estimation by which the posterior negative log-density H becomes the objective function to be minimized. This function can be further simplified by denoting the MAP estimate of s by ŝ, the MAP criterion can be expressed as

$$\widehat{s} = arg_c \max H(c|x) \tag{2.6}$$

The above equation implies that it maximizes the entropy relation by varying the argument.

However the problem now becomes to estimate the entire posterior prior density of the sources should be estimated by Markov-chain Monte Carlo methods which becomes practically difficult. By reducing the relation and by assumption of source densities the Bayesian model is expanded for the spare coding in chapter 5. The next step is to model

the residual or noise term. They are modeled as iid Gaussian with zero mean. By these assumptions and reductions the final equation can be written as:

$$H(c|x) = \frac{R}{2\sigma_v^2} + \sum_i h_i(c_i) + C_o \tag{2.7}$$

where $h_i(c_i)$ is the prior log-density of the source

σ_v^2 is the variance of the noise density

C_o is a constant.

R is the residual error or reconstruction error.

There are lot of works towards this model by assuming different source densities and also noise densities for estimation [9].

2.3.2 Gradient based learning

The algorithm in chapter 5 is based on this gradient learning method [18][20][21]. Here the source densities are smooth such as laplacian and so the objective function can be optimized by gradient-based optimization. The exact form of the prior density is not known and how close the hypothetical density matches this actual density determines the efficiency of these methods. The penalty term used by the algorithm is defined as $\widehat{h}(c)$ to distinguish it from the actual log-density of the sources.

The gradient of the objective function is:

$$\nabla_c H = \frac{\mathbf{g}}{\sigma_v^2} + \nabla_c \sum_i \widehat{h}(c_i) \tag{2.8}$$

where g is the gradient of the reconstruction error.

$$g = \frac{1}{2}\nabla_c R \tag{2.9}$$

The problems with this approach is though the error in the source estimate gets spread over many projections rather than updating the corresponding coefficient.

2.4 Sparse coding methods

2.4.1 Matching Pursuit

Matching pursuit is one of the most influential method on sparse coding which is just an extension of vector quantization based on iterative greedy strategies [25][26]. Chapter 6 is based on matching pursuit applying with the inhibition method one of the special case of matching pursuit algorithms.

2.4.2 Orthogonal Matching Pursuit

This is very similar to the matching pursuit. The variations are as follows:

1) Select the coefficients as in matching pursuit and its state is changed from inactive to active. Truncate the non-selected coefficients to inactive state.

2) The reconstruction error is minimized in the subspace of active coefficients.

Here the iterations are more complex but the objective function is well defined as a weighted sum of active coefficients [26].

2.4.3 Inhibition method

The inhibition method will be further developed and applied in chapter 6. This is a special case of matching Pursuit. The matching pursuit algorithm can be generalized by

updating more than one coefficient per iteration, but by following the constraint that the updated coefficients corresponds to orthogonal or near orthogonal code vectors. However the iterations for the matching pursuit is not much more complex than the matching pursuit. This method provides faster convergence than other matching pursuit algorithms for natural images in both computational cost and also in number of iterations needed, while the number of active coefficients for a given reconstruction error are almost the same. The special note about using this algorithm is that only the inhibition method and the matching pursuit sparse coding methods has been previously applied towards image compression on natural images with the same code book.

2.5 Biological significance of sparse coding

The work on analyzing the human vision system and thereby the working model of the brain from the neural research have imminent proof that responses of simple cells in visual cortex area V1 are projections of the retinal image (pre filtered by retinal neurons) onto Gabor like code vectors [2][5]. The degree of overcompleteness of this Gabor representations is very difficult to estimate since the number of neurons in the V1 exceeds the number of inputs by a factor of about 500 and also these neurons serve purposes of processing not only representation of the input. Considering approximately that 70 percentage of the neurons in V1 are output, it is reasonable enough to believe that V1 representation is overcomplete. The research works in psychophysics and brain research has some long history of works in the idea that sparse coding plays an essential role in sensory perception and particularly visual perception. The evidence that the sparse coding is used in V1 are as follows:

1) Cortical cells are less active than their retinal inputs, i.e. the distribution of their activity is sparse [2].

2) Simple cells in V1 are non linearly tuned to their stimuli and cross oriented inhibition (the nonlinear tuning of simple cells to the orientation of Gabor stimuli). However it cannot be stated that the representation in V1 is not ideally sparse in the sense that the tuning is not so sharp that only one simple cell responds to a Gabor stimulus.

The research is still continuing in these areas and it concentrates on to analyze whether the sparse coding is carried out iteratively. Research has been carried out still in these areas without any solid proof to prove any hypothesis developed. Research on these areas implies that the solution to the problem of sparse coding implies solution to the problem of sparse-code learning but not the vice-versa [9]. There are number of works towards developing learned codebooks for the natural images. Most of the work produces Gabor like code vectors.

CHAPTER 3: STATISTICS OF NATURAL AND BLURRED IMAGES

The visual coding is significantly influenced by the statistical properties of the images that it encounters. So this chapter briefs the statistical properties of the natural and blurred images. Furthermore images that are falling on the retinal regions of the eye are blurred and the high frequency spatial information are recovered by the brain only by further processing on corresponding layers. Hence this chapter also discusses the optical distortions which are primarily modeled using PSF (Point Spread Functions) and whitening filters.

3.1 Statistics of natural images

The images from the natural environment do not consists of random patterns but consists of images with consistent statistical properties. A knowledge of these statistics can lead to a better understanding of the coding of visual system.

Image set for computing the statistics consists of 10 natural images of trees, mountains etc. No attempt was made to select particular type of image, but images was ensured that it has no artificial objects. Though there are various statistical properties that can be analyzed, the primary property is the image's amplitude spectrum. Natural images on the whole appear to be rather complex filled with various objects and shadows and different surfaces. Amid this complexity they can be easily distinguished from variety of other class of images. This difference is best described in terms of the amplitude spectra or power spectra of the images (square root of amplitude spectrum).

The two dimensional spectra of these images shows greatest amplitudes at low frequencies and decreasing amplitude as frequency increases. The amplitude falls off quickly by a factor of $\frac{1}{f}$. Fig.3.6 shows the amplitude spectra averaged across all orientations and plotted on log-log scale.

3.2 Factors impairing the quality of image falling on the retina

Many factors affects the quality of the image that falls into the retina such as optics, photo-pigment spectral sensitivities, effective mean absorbed quanta, internal neuron noises, eye movements, photon Poisson noises etc [30]. This chapter briefs only on the optical distortions neglecting the effects of other factors.

3.2.1 Point Spread Function

This constitutes the major distortion to the input image. When a dot stimulus was presented to the eye, the image of a point source will be distributed or spread on the retina. This effect is mainly due to abberations and diffractions. The optics of the eye at a given eccentricity can be described by a point spread function, $h(x, y)$. Campbell and Gubisch [31] measured the point spread function in terms of the foveal line spread function. From the line spread function a point spread function can be derived and it is related to the pupil's diameter. The measured line spread function can be typically approximated by the sum of two, two dimensional Gaussian function [32][33].

$$h(x) = \frac{a_1}{2\pi a_3} \exp[\frac{-0.5.x^2}{a_3^2}] + \frac{a_2}{2\pi a_4} \exp[\frac{-0.5x^2}{a_4^2}] \tag{3.1}$$

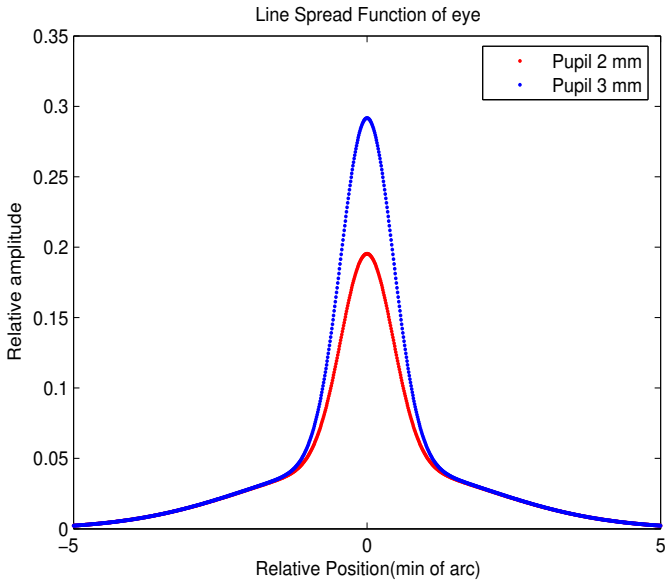

Figure 3.1: Line spread function of eye

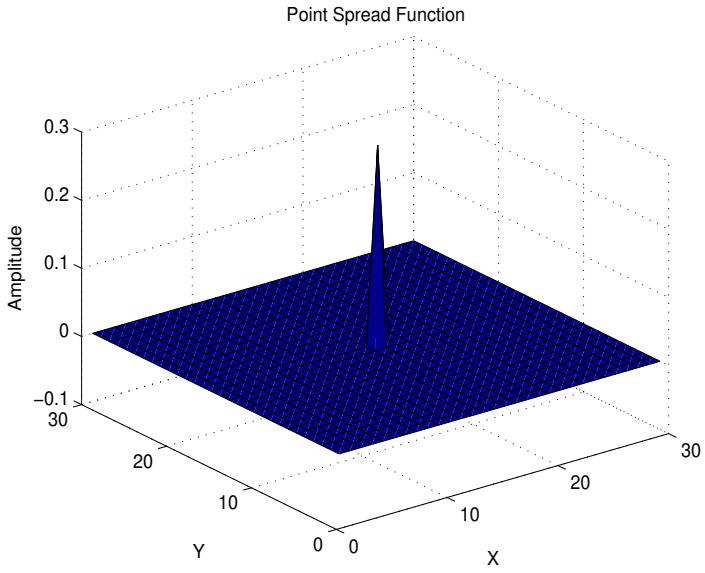

Figure 3.2: Point Spread Function

For a 2 mm pupil the curve produced by Eqn.(3.1) with $a_1 = 0.684$, $a_2 = 0.587$ and $a_3 = 2.305$ mm fits well to the measured data from Campbell and Gubisch [31]. For a 3 mm

pupil the coefficients are $a_1 = 0.417$ min, $a_2 = 0.583$ min, $a_3 = 0.443$ min, and $a_4 = 2.04$ min. Fig.3.1 shows the line spread function for the eye with pupil diameters of 2 mm and 3 mm. This line spread equation can be used to derive the PSF. Fig.3.2 shows the corresponding Point Spread Function derived from Eqn.(3.1).

3.2.2 Whitening filter

The blurred images are further passed through a whitening filter to model our analysis. By applying whitening filter, the image data is sphered by equalizing the variance in all directions [1]-[3]. Since the amplitude spectrum falls as roughly as $1/f$ at all orientations in 2D frequency plane, syphering may be accomplished by filtering with a circularly symmetric "whitening filter" with frequency response $W(f) = f$, thereby attenuating the low frequencies and boosting the high frequencies so as to yield a roughly flat amplitude spectrum at across all spatial frequencies. However it is not wise to boost all the high frequencies indiscriminately for several reasons: First reason is that the highest spatial frequencies will be corrupted by noise and effects of aliasing, second the energy presented in the corners of the 2D frequency domain is an artifact of working on a rectangular sampling lattice, since there is an effectively higher sampling densities across the diagonal than along horizontal or vertical directions. For these reasons it is appropriate to cut out the the energy at the highest spatial frequencies and also in the corners of the 2D Fourier plane by filtering with a circularly symmetric low-pass filter. Hence an exponential filter with the given frequency response is used.

$$R(f) = \exp\left(-\frac{f}{f_0}\right)^n \tag{3.2}$$

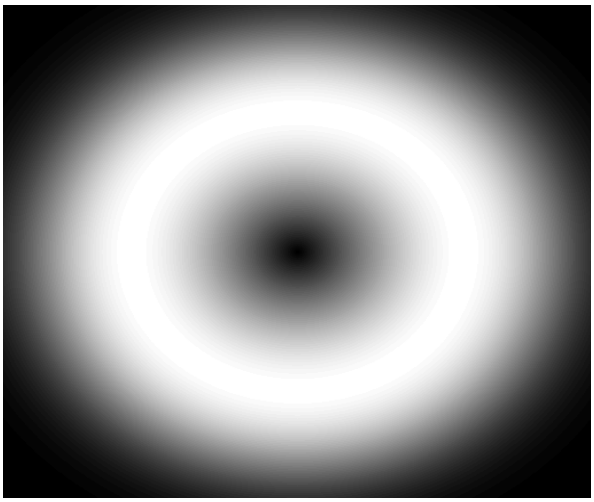

Figure 3.3: Whitening filter in frequency domain

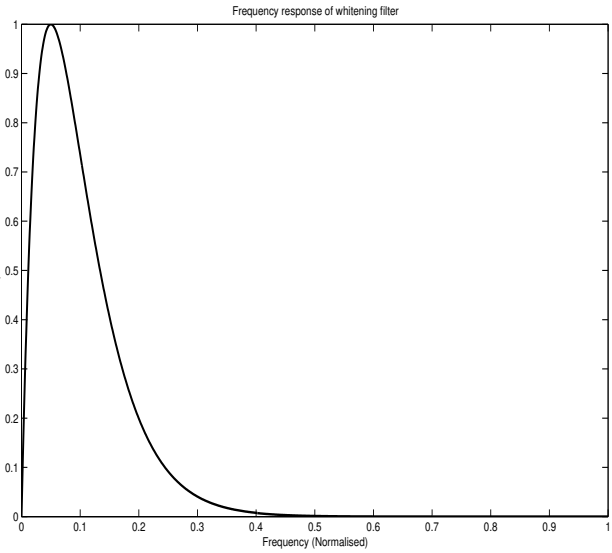

Figure 3.4: Frequency response of whitening filter

where f_0 represents the cut-off frequency (taken as 200 cycles/picture), and n is the steepness parameter (taken as 4). The steepness parameter is chosen to produce a fairly sharp cutoff but without being so sharp as to introduce substantial ringing in the spatial domain. The combined whitening and low pass filter is together used as whitening filter that has the following frequency response.

$$R(f) = W(f)R(f) = f \exp(-\frac{f}{fo})^4 \tag{3.3}$$

The phase of the filter is set to zero at all frequencies. The profile of the filter is shown in Fig.3.3 and Fig.3.4

3.3 Analysis of amplitude spectrum

Fig. 3.5 takes a simple natural image and Fig. 3.6 displays the amplitude spectrum of it. It can be clearly seen that the amplitude drops of with a slope of $\frac{1}{f}$. Fig. 3.7 illustrates the blurring effect as a result of the application of the PSF as per Eqn. (3.1). Fig. 3.8 displays the amplitude spectrum of this image. It can be seen that the very high frequency amplitudes are further dropped down and also the spectrum falls off more smoothly than the natural image spectrum. Fig. 3.9 shows the effect of the application of the whitening filter to the image and Fig. 3.10 shows its corresponding amplitude spectrum. It can be seen that the effect of whitening makes the spectrum much flatter till it reaches the cut off frequency and there is sudden drop off in amplitudes for frequencies greater than cut off frequency. Fig. 3.11 shows the effect of applying PSF as well as whitening filter and normalizing the variance of the images. Fig. 3.12 shows the corresponding amplitude spectrum. The effect of applying PSF as well as whitening filter makes the spectrum much flatter in frequencies

below cut-off and also smooth transition or drop off for frequencies greater than cut off frequency.

Figure 3.5: Natural image

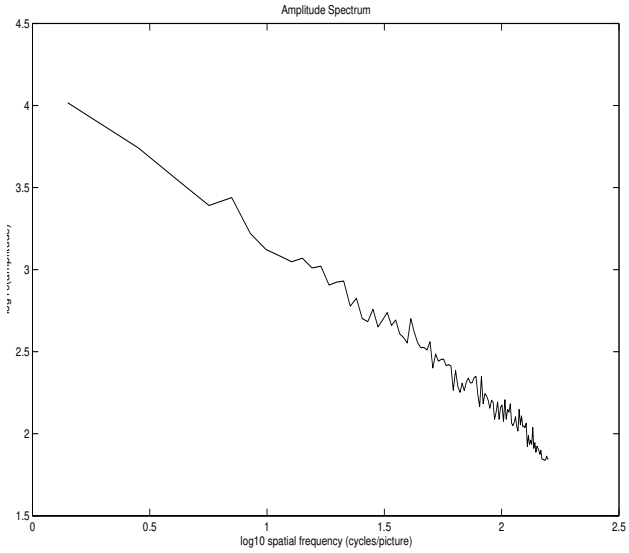

Figure 3.6: Amplitude spectrum of natural images

Figure 3.7: Blurred image by PSF

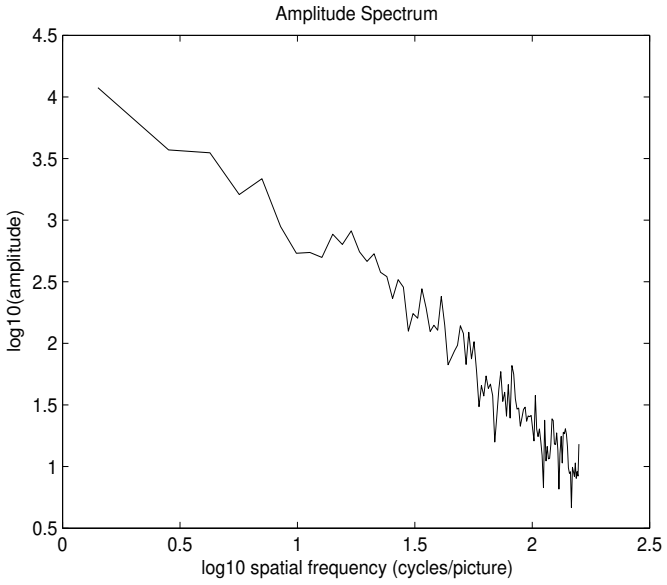

Figure 3.8: Amplitude spectrum of blurred image

Figure 3.9: Whitened and normalized image

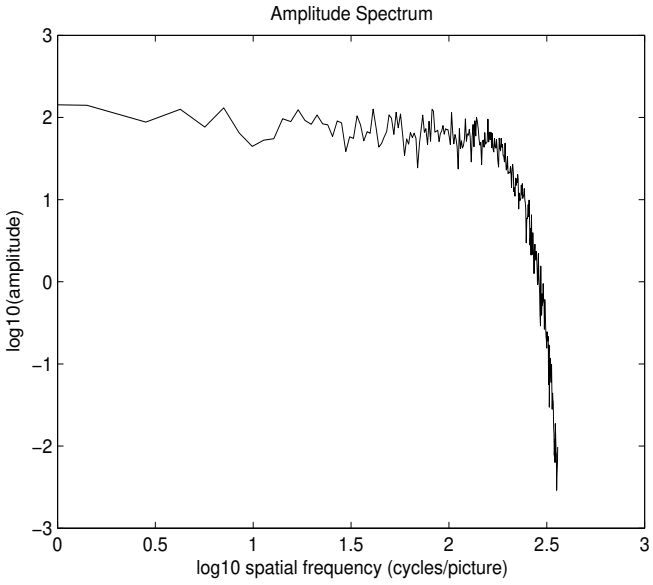

Figure 3.10: Amplitude spectrum of whitened and normalized image

Figure 3.11: Blurred,whitened and normalized image

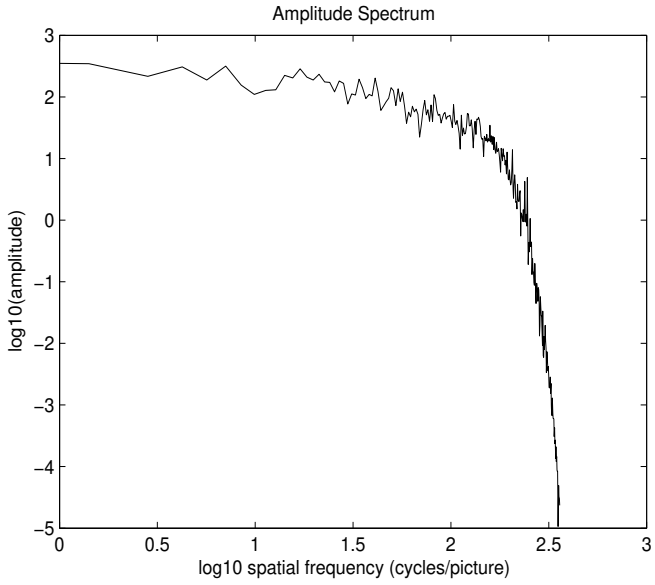

Figure 3.12: Amplitude spectrum of blurred,whitened and normalized image

CHAPTER 4: JPEG AND JPEG 2000 COMPRESSION STANDARDS

This chapter describes the practical JPEG and JPEG 2000 compression standards. JPEG (Joint Photographic Experts Group) a joint ISO/CCITT committee [41] established the first international standard for continuous-tone still images, both grayscale and colour. JPEG 2000 [46]-[49] was later developed by the same group that targeted wide variety of images for different applications under different settings. It is termed as a compression standard with interactive imaging. The ultimate goal of the proposal [43] is to have one unified standard to accomplish different tasks such as lossless compression, low complexity encoding-decoding, scalability and efficiency. JPEG baseline method can be very efficiently implemented in hardware and software. It also performs well in high BPP coding rates but fails to provide flexibility in coding that is very much needed for applications such as internet transmissions. So JPEG 2000 was developed to provide flexibility in coding based on the application.

4.1 Introduction to JPEG

The JPEG standard [41] includes two basic compression methods, each with various modes of operation. A DCT based method is specified for 'lossy' compression and a predictive method for 'lossless' compression. A simple lossy technique called Baseline method is by far the most widely implemented method and the same is implemented in this thesis.

4.2 Processing steps for DCT based coding

Fig.4.1 illustrates the basic block diagram of JPEG.

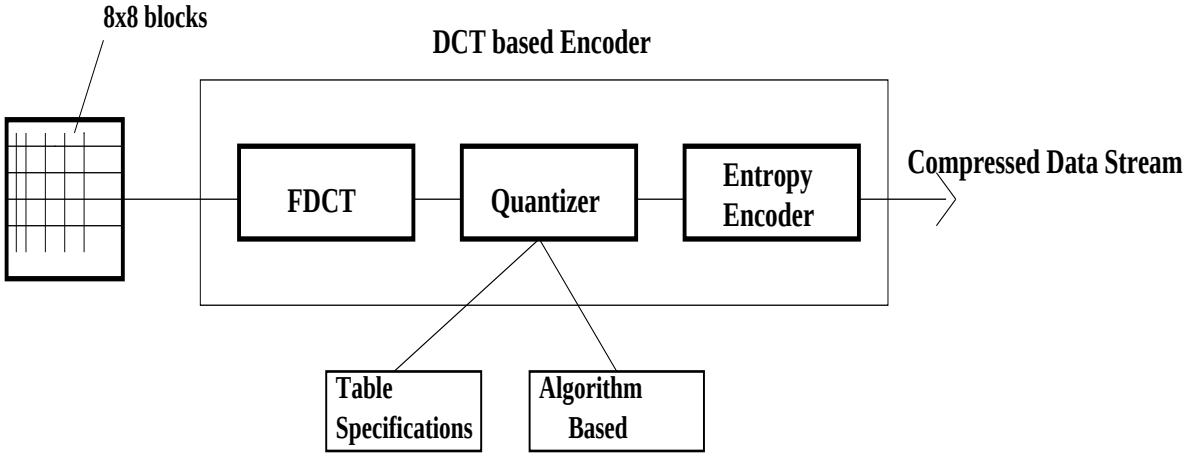

Figure 4.1: JPEG encoder

Each source image undergoes subimage segmentation or gets divided into blocks of 8x8 gray scale sample images. Each 8x8 block goes through identical processing step and yields output in the compressed form as data stream.

FDCT and IDCT:

At the input to the encoder, source image samples are grouped into 8x8 blocks, shifted from unsigned integers with the range of $[0, 2^p - 1]$ to signed integers with range $[-2^{p-1}, 2^{p-1}]$. It then undergoes forward Discrete Cosine Transform given by Eqn.(4.1)

FDCT:

$$F(U,V) = \frac{1}{4}C(U)C(V)[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\frac{cos(2x+1)u\pi)}{16}\frac{cos(2x+1)v\pi)}{16}] \tag{4.1}$$

where $C(u), C(v) = \frac{1}{\sqrt{2}}$ for $u, v = 0$ and $C(u), C(v) = 0$ otherwise

Similarly to convert back to the image spatial domain in the decoder Inverse Discrete Cosine Transform is used given by Eqn.(4.2)

IDCT:

$$f(x,y) = \frac{1}{4}[\sum_{u=0}^{7}\sum_{v=0}^{7} C(u,v)F(u,v)\frac{cos(2x+1)u\pi)}{16}\frac{cos(2x+1)v\pi)}{16}] \tag{4.2}$$

4.2.1 Quantization

The output of FDCT gives 64 DCT coefficients for the given sub image block. The purpose of quantization is to achieve compression representing DCT coefficients with no greater precision than necessary to achieve desired image quality. Quantization is many-to-one mapping and therefore it is the principal source of lossyness in DCT based encoders. When the aim is to compress the image as much as possible without visible artifacts, each step size ideally should be chosen as the perceptual threshold or just noticeable difference. These thresholds are also functions of the source image characteristics, display characteristics and viewing distance. This clearly shows that quantization table or algorithm used or defined for sharp image will be completely different from the blurred whitened images.

For natural images quantization table mentioned in Table.4.1 is used [42][43]. The method of quantization is by dividing the DCT coefficients with the given quantization table and rounding off the resultant values as per Eqn.(4.3)

$$R(u,v) = \frac{F(u,v)}{Q(u,v)} \tag{4.3}$$

where $F(u,v)$ are the DCT coefficients. $Q(u,v)$ is from the quantization table.

Apart from quantization based methods, there are also selectivity based algorithms described in the following sections. This chooses only set of few DCT coefficients and blindly truncates others to zeros. Since for blurred retinal images there are no standard quantization tables proposed, the selectivity bases quantization methods are chosen. After quantization only few DCT coefficients will be left over as most of the coefficients becomes zero which can be encoded efficiently by source coding techniques.

Table 4.1: Quantization table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Other quantization methods are listed as follows:

CASE 1: A single threshold can be applied to all sub image blocks. The coefficients above this threshold are selected and below are truncated to zero. This is a very coarse quantization technique, not able to produce efficient results.

CASE 2: A fixed number of coefficients are selected based on the zig-zag pattern discussed earlier. The remaining coefficients are truncated to zero. Here the coding rate is kept constant which determines the number of coefficients selected.

CASE 3: A fixed number of coefficients are selected based on their position and also the magnitude. However this method produces variable coding rate.

The results section analyzes the RMS coding performance of case 2 and case 3 algorithms.

4.2.2 DC coding and zig-zag sequence

After quantization, the DC coefficients are treated separately from the 63 AC coefficients. The DC coefficient in a given block is a measure of the average value of the 64 image samples. Since there is strong correlation between DC coefficients of adjacent 8x8 blocks, the quantized DC coefficients are encoded by differential encoding method. The DC coefficient contains significant fraction of the total image energy and hence it should be encoded with much protection. The AC coefficients are arranged in zig-zag pattern as mentioned in Fig. 4.2. This ordering facilitates entropy encoding by placing low frequency coefficients which are more likely to be nonzero and high frequency coefficients that are mostly zero in order.

4.2.3 Intermediate entropy coding representation

The coding methods are based on the standards [42] [43] and Huffman codes (given as Appendix) [44].

In the intermediate entropy encoding stage, each AC coefficient is represented in combination with the 'runlength' of zero valued AC coefficients which precedes it in the zigzag sequence. Each such runlength/non-zero combination is represented by a pair of symbols:

symbol 1 symbol 2

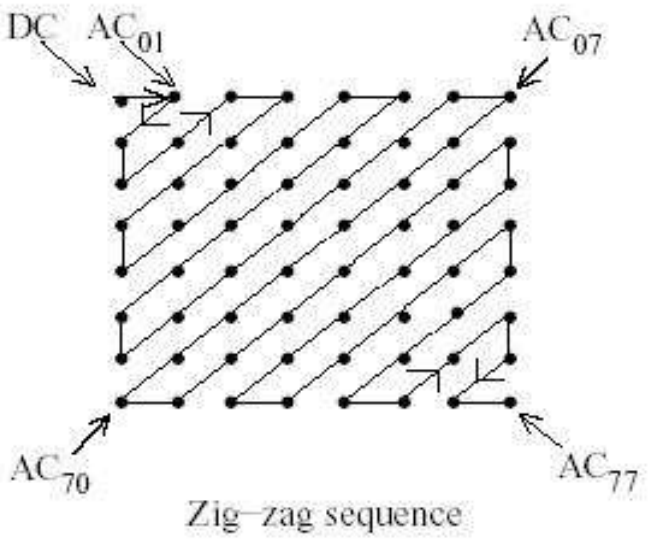

Figure 4.2: DC coefficients and zig-zag pattern of AC coefficients

(RUNLENGTH, SIZE) (AMPLITUDE)

Symbol 1 represents RUNLENGTH and SIZE. Symbol 2 represents the amplitude of the non-zero AC coefficient. RUNLENGTH is the number of consecutive zero valued AC coefficients that precedes the non-zero coefficient. SIZE is the number of bits required to encode the AMPLITUDE in a signed magnitude representation.

RUNLENGTH represents the zero-runs of lengths from 0 to 15. Actual zero runs in the zig-zag sequence can be greater than 15. So the symbol-1 value (15, 0) is interpreted as an extension symbol with run-length=16. There can be up to three consecutive (15, 0) extensions before the terminating symbol-1. But to represent the last run of zeros the special symbol-1 value (0, 0) which means EOB (End Of Block) is used. It can be viewed

as an *Escape Symbol* which terminates the 8x8 sample block. Thus for each 8x8 block of samples the zig-zag sequence of quantized co-efficients are represented as sequence of symbol-1, symbol-2 pairs. In our case AMPLITUDE codes are represented with 1 to 10 bits and RUNLENGTH is represented by values from 0 to 15. The intermediate representation for differential DC coefficient is structured similarly. Symbol 1 here represents SIZE and symbol 2 represents AMPLITUDE. Because the DC coefficients are differentially encoded, so one additional level must be added to the SIZE and AMPLITUDE table ranging from 1 to 11 is used.

Symbol 1	Symbol 2
(SIZE)	(AMPLITUDE)

4.2.4 Huffman coding

The quantized coefficients for an 8x8 blocks are computed and represented in intermediate symbol sequence. For each 8x8 block the DC coefficient's symbol 1 and symbol 2 are coded first. For both DC and AC coefficients, each symbol 1 is encoded with a variable length code from the Huffman table (mentioned in Appendix as AC category codes). Each symbol 2 is assigned with a variable-length integer code whose length is given in the SIZE representation.The length of a Huffman code is not known until it is decoded fully, but the length of a VLI code is stored in its preceding VLC.

4.3 A simple example for JPEG encoder

This section takes a 8x8 block from an image and goes through all the steps involved in JPEG compression. Table 4.2 gives 8x8 pixel values of a subimage . We subtract each pixel value by 128 to create an array of pixels ranging from -128 to 128 (given in Table 4.3). These values are transformed using FDCT which results in Table 4.4. These coefficient values are now quantized as per Eqn.(4.3) which would result in Table 4.5. For example the first FDCT coefficient is 394.1250. This value is divided by 16 and rounded off to gives 25 as the quantized result.

Table 4.2: Source image (8x8 block)

196	196	194	189	184	180	177	180
189	186	182	177	175	168	164	166
195	191	183	178	177	172	167	169
188	188	182	177	175	171	165	165
185	188	183	179	176	169	160	158
191	189	185	181	181	174	164	165
181	177	175	166	167	164	154	152
194	191	187	179	176	173	169	166

Table 4.3: Image sample shifted in scale

68	68	66	61	56	52	49	52
61	58	54	49	47	40	36	38
67	63	55	50	49	44	39	41
60	60	54	49	47	43	37	37
57	60	55	51	48	41	32	30
63	61	57	53	53	46	36	37
53	49	47	38	39	36	26	24
66	63	59	51	48	45	41	38

Table 4.4: FDCT coefficients

394.1250	72.0068	-0.5924	3.8771	1.1250	-7.1492	2.8161	0.6588
21.6568	-4.9030	3.7292	-4.5241	2.8650	-0.0410	-1.5541	-0.9027
11.8748	-2.9182	4.1365	-1.9731	-0.6088	2.6380	1.4812	1.3606
4.8939	-1.1292	-6.8714	-1.7957	0.5513	-1.0840	0.7910	-0.4590
16.8750	-1.0591	-1.0640	-3.5215	-2.6250	1.1884	-1.5888	-1.1737
-4.3393	-2.9301	1.9186	1.0676	-1.0727	-0.1133	1.6931	-0.0256
27.6884	-0.7598	2.2312	-0.2322	1.4699	-0.1273	-0.6365	-0.7127
-6.8293	2.2489	-0.6426	0.4374	-1.0123	-0.1350	0.0260	-0.1880

Table 4.5: Quantized DCT coefficients

25	7	0	0	0	0	0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

By taking the coefficients in zig-zag sequence the data for this 64 element block is reduced to these set of elements $25, 7, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0$.

The DC coefficient 25 is encoded as:

(SIZE) (AMPLITUDE)

(5) **(25)**

The intermediate symbols are then Huffman encoded. DC and AC coefficients follow different tables. The amplitude values are coded using 2's complement representation.

Codes for the data:

DC coefficient VLC **(5)** **(110)**

DC coefficient VLI **(25)** **(11001)**

AC Coefficients codes

(RUNLENGTH, SIZE) VLC code (AMPLITUDE) VLI CODE

(0, 3) **(100)** **(7)** **(111)**

Table 4.6: AC coefficients intermediate Codes

AC Coefficients	**Code (RUNLENGTH, SIZE), (AMPLITUDE)**
7	(0, 3), (7)
2	(0, 2), (2)
1	(0, 1), (1)
1	(6, 1), (1)
1	(10, 1), (1)
0(EOB)	(0, 0), (0)

(0, 2)	**(00)**	**(2)**	**(10)**
(0, 1)	**(00)**	**(1)**	**(1)**
(6, 1)	**(1111011)**	**(1)**	**(1)**
(10, 1)	**(1111111110110111)**	**(1)**	**(1)**
(0, 0)	**(1010 EOB)**	**(0)**	**(0)**

Thus the entropy codes for the given 8x8 block data including EOB code are:

[1101100110011100100011111011111111111011011111010]

It is evident that only 51 bits are required for encoding 64 pixel values.

4.4 JPEG decoder

The entire process of decoding is illustrated in Fig.4.3. The decoder receives bit streams of 0's and 1's. First step is the entropy decoding by using standard Huffman decoders that are supplied with the table specifications. The bit streams are decoded back to the

original values and are arranged into 8x8 blocks. Then IDCT is applied to each block given by Eqn.(4.2). Final stage is to scale shift the image to its original representation of $[0, 2^p - 1]$. The compression achieved and the error performance of the system are analyzed using the following set of equations Eqn.(4.4) - (4.8). The results of the JPEG compression algorithm are tabulated in chapter 7.

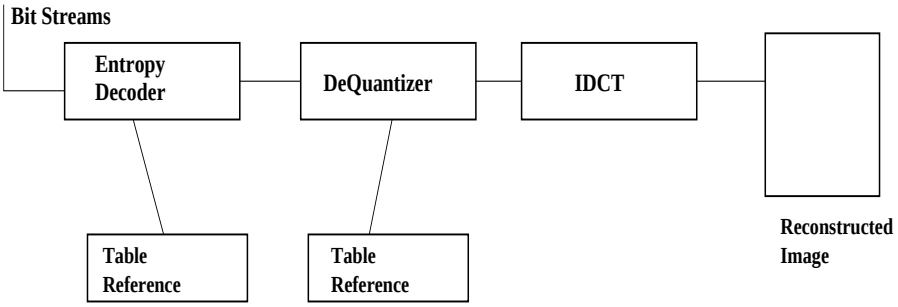

Figure 4.3: JPEG decoder block diagram

$$\text{Compression Ratio} = \frac{\text{No of bits required to represent original image}}{\text{No of bits required for compressed image}} \tag{4.4}$$

$$\text{Bits Per Pixel(BPP)} = \frac{\text{No of bits in the compressed image}}{\text{Total Number of pixels}} \tag{4.5}$$

$$PSNR = 10 \log \frac{\sum_{i=1}^{n} f_i^2}{\sum_{i=1}^{n} (f_i - f_i')^2} \tag{4.6}$$

where f_i is the original image, f_i' is the reconstructed image in spatial domain and F_i and F_i' are in frequency domain.

$$PSNR = 10 \log \frac{\sum_{i=1}^{n} F_i^2}{\sum_{i=1}^{n} (F_i - F_i')^2} \tag{4.7}$$

$$RMS = \sqrt{\frac{\sum_{i=1}^{N}(f_i - f_i')^2}{N}} \tag{4.8}$$

Existing problems of JPEG

1) It is hard to achieve exact BPP coding rates using JPEG encoder.

2) JPEG has poor RMS performance at low BPP coding rates. Low BPP rates are mostly used for internet and transmission based systems. So poor performance makes it less attractive for these applications.

3) It supports only full reconstruction rather than region or resolution based decodings. So all data bits must be recovered for reconstruction.

4.5 Introduction to JPEG 2000

JPEG 2000 was developed and published as an ISO/IEC standard as well as an ITU-T Recommendation [45] - [54].

Major improvements over JPEG are listed as follows:

1)**Superior low BPP performance:**

Many compression algorithms including JPEG, can achieve good RMS performance at low compression ratio or equivalent high BPP rate but deliver poor performance at high compression ratio or low BPP. JPEG 2000 can significantly improve performance at high compression ratios.

2) **Progressive transmission by pixel accuracy:**

Progressive transmission is highly preferred for transmissions over communication channels. JPEG 2000 can support progressive transmission in two ways: progression by image component and progression by spatial location. For example, the user can select any particular image component or all components at a particular location or combination of both to transmit.

3) **Randomness in code-stream access and processing:**

JPEG 2000 offers the code-streams with capabilities to support the spatial random access called 'region of interest access' at varying degrees of granularity. 'Degree of interest' is also supported whereby the quality of the decompressed image can be adjusted for each region of interest.

4) **Robustness to bit-errors:**

JPEG 2000 introduces resynchronization markers and introduce error control coding of smaller data blocks so that the errors inside the block can be corrected.

4.5.1 Processing steps for JPEG 2000

Fig.4.4 illustrates the JPEG 2000 encoder. The implementation provided in the following sections, however is a simple progressive coding without the features of region of interest coding. It follows a discrete wavelet transform and the irreversible scalar quantization and EZW based bit plane coding source encoder. These were chosen for its simplicity in implementation as well as the high performance.

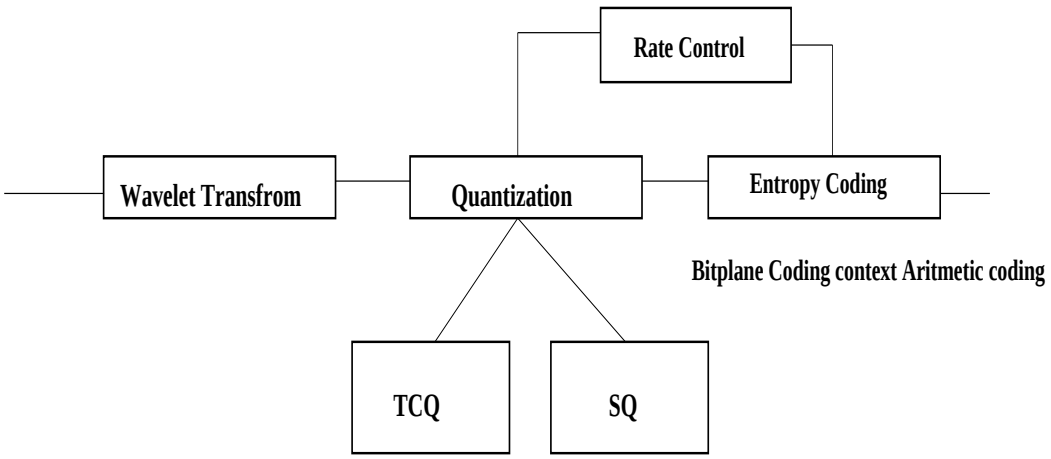

Figure 4.4: JPEG 2000 encoder

To encode an image of size 2^J x 2^J pixels,we first need to tile the image components according to their properties and positions. This step is called tiling process which is essential to achieve progressive transmission and random code stream access. In this thesis since we are only interested in compression of full images, the tiling process is not considered. After the tiling process, the images are shifted in scale from its representation of $[0, 2^p - 1]$ to signed

integers with range $[-2^{p-1}, 2^{p-1}]$. Then the image is transformed using wavelet transform. To perform this transformation the analysing function ψ and minimum decomposition level(D) must be fixed. The wavelet selected as forward and inverse transforms affects directly the computational complexity and indirectly the compression rates and image reconstruction capabilities. The most widely used wavelets are Daubechies and bi-orthogonal filters. In this thesis, we select Daubechies 9-7 filters listed in Table 4.7,4.8. The following sections describe the wavelet transforms, quantization and coding in detail.

4.5.2 Wavelet transform

The scale shifted image of size 2^J x 2^J is passed through the subband coding structure as described in the Fig.4.5. As a first step, this image is convolved with filters $h_0(m)$ and $h_1(m)$ and downsampled along the rows to yield a pair of $2^{\frac{J}{2}}$ x 2^J array. Now this array is fed to the pair of decomposition filters $h_0(n)$ and $h_1(n)$ and they are convolved and downsampled along the columns. The decomposition filter coefficients are shown in Table 4.7. Repeating this procedure will yield a set of four $2^{\frac{J}{2}}$ x $2^{\frac{J}{2}}$ arrays called LL1,HL1,HH1 and LH1. Inside these arrays the maximum energy is packed in the LL1 array called the approximate of the image. The remaining three arrays contain main features of the image in Vertical (LH1), horizontal (HL1) and diagonal (HH1) aspects. These four arrays are arranged into a single 2^J x 2^J matrix as shown in Fig 4.7. This is called the first level decomposition in which the given image is broken down into a set of approximate, horizontal, vertical and diagonal coefficients. Each of them has reduced resolution than the original image. To perform the second level decomposition the approximate array (LL1) is fed back to the same subband

coding structure which results in four $2^{\frac{J}{4}}$ x $2^{\frac{J}{4}}$ arrays called as LL2, HL2, HH2 and LH2. Repeating this process, we can obtain third and fourth levels of decompositions. In practice, we can achieve significant improvement in compression ratio at the second and third levels, but the improvement is diminished after four or more levels and hence we consider only three levels of decomposition in this thesis. Fig.4.7 illustrates the two level decomposition process and Fig.4.6 illustrates the two level decompositions coefficients of the original image Fig.6.8. Fig.4.8 illustrates the iterative reconstruction process to get back the original image from decomposed coefficients.

Table 4.7: Decomposition filters

Filter	Filter Coefficients						
ho(m)	0	0.02674	-0.01686	−0.07822	0.26686	0.60294	0.26686
	−0.07822	-0.01686	0.026748				
h1(m)	0	-0.09127	0.05754	0.59127	-1.11508	0.59127	0.05754
	-0.09127	0	0				

Table 4.8: Synthesis filters

Filter	Filter Coefficients						
ho(m)	0	-0.0912	-0.057543	0.5912	1.1150	0.5913	-0.5913
	−0.0913	0	0				
h1(m)	0	-0.0267	-0.0168	0.0782	0.2668	0.0782	-0.0168
	-0.0267	0	0				

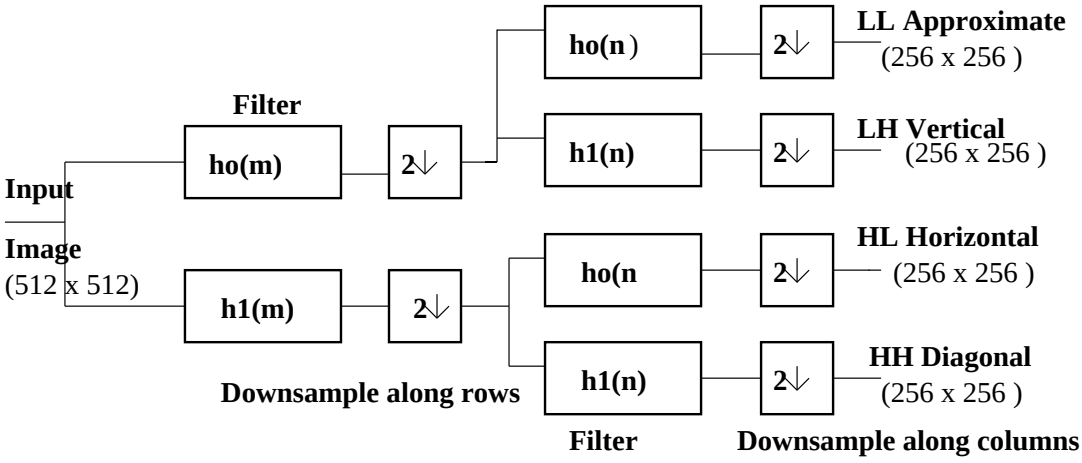

Figure 4.5: Sub band coding method

4.5.3 Quantization

This section discusses the scalar quantization method. It is very important to analyze the genealogy relationship of the coefficients such that they can be grouped efficiently for coding and also to determine the quantization levels.

Genealogy ('parent-child') relationship among the coefficients from sub bands of the same orientation are essential to determine the quantization levels [54] [55]. Fig.4.9 explains

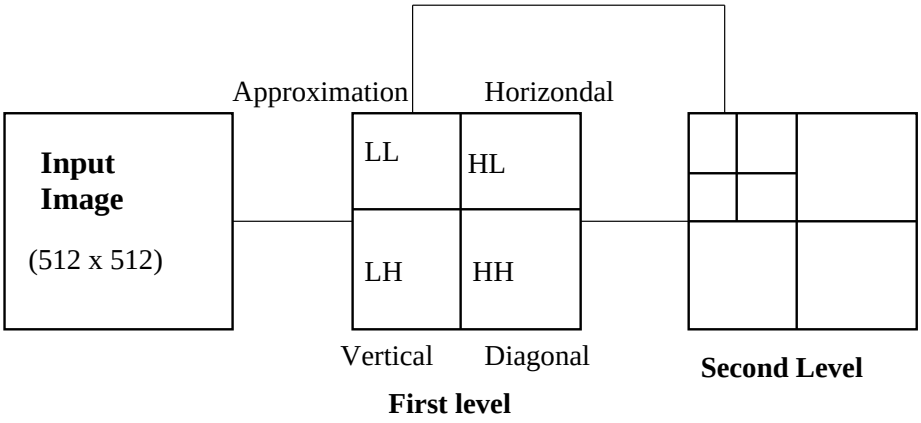

Figure 4.6: Multiresolution decomposition

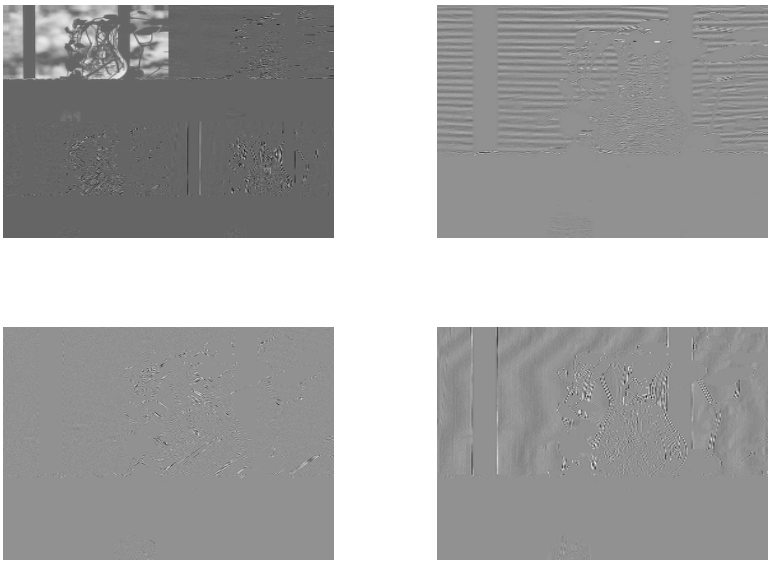

Figure 4.7: Two level decomposition of an image

the genealogy relationship diagrammatically. Each coefficient in HL_d is parent of 4 children in the sub band HL_{d-1}. 'D' is the total number of decomposition levels used in the transform. These relationship are same for the following LH_d, HH_d sub bands.

Similarly this relationship holds for each $d = 2, 3, ...D$. The coefficients in the highest

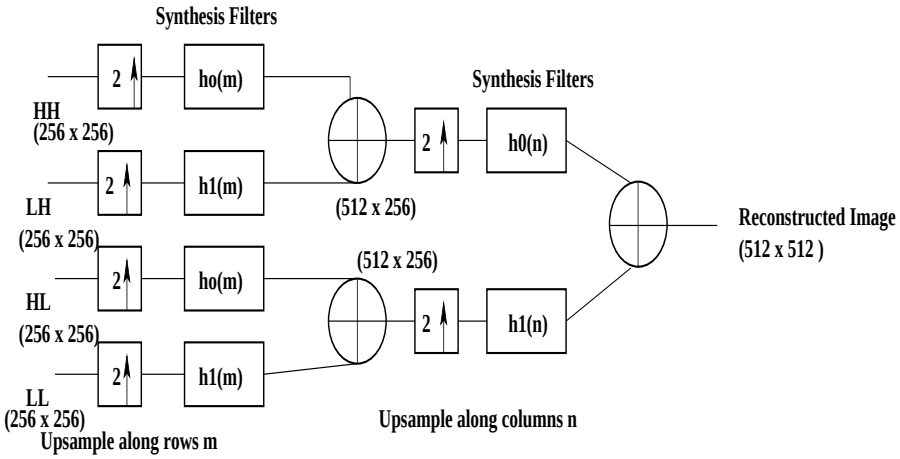

Figure 4.8: Reconstruction of sub band coding

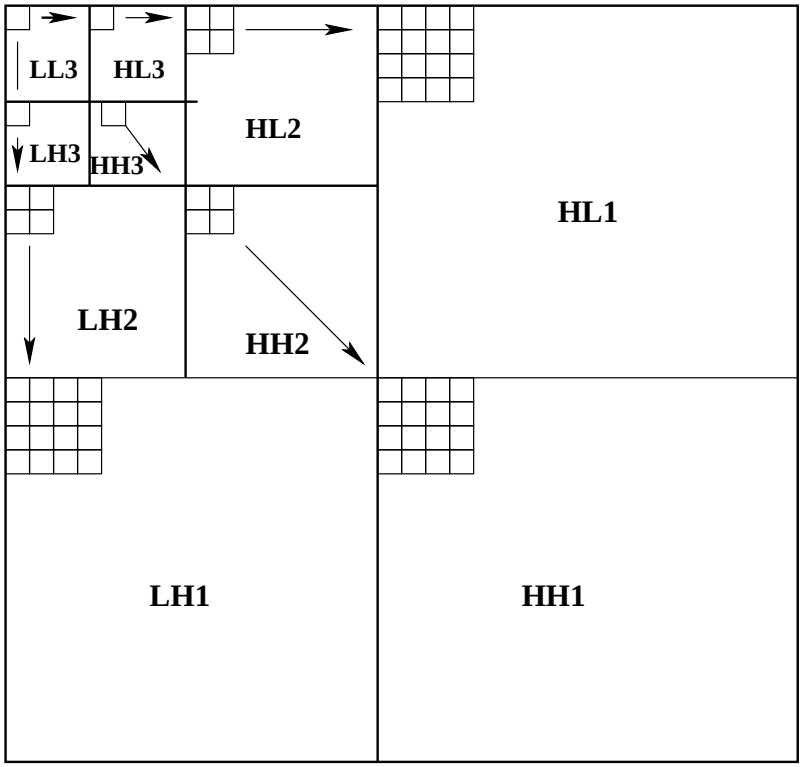

Figure 4.9: Genealogy of sub band coefficients (3 levels of decomposition)

frequency sub bands (d=1) have no children. The coefficients in the lowest frequency sub

band LL_D have only 3 children comprised of single coefficient at the same spatial locations in the HL_D, LH_D and HH_D sub bands.

The 'descendants' of a coefficient consists of all children, grandchildren and great-grandchildren etc. In general any coefficient in the sub band HL_D, LH_D and HH_D sub band has 4 children, 16 grandchildren and 64 great-grand children etc. Thus each coefficient has total number of descendants given by Eqn.(4.9)

$$4 + 4^2 + ... + 4^{d-1} = \frac{4^d - 4}{3} \tag{4.9}$$

Similarly each coefficient in LL_D has 3 children, 12 grandchildren and 48 great-grandchildren for a total of $4^D - 1$ descendants.

This section proceeds on to discuss the quantization level determinations. The basis of quantization comes from the fact that most of the image energy is concentrated into only a few transform coefficients. For wavelet sub band transform the highest energy coefficient resides in the lowest frequency sub bands. In Fig.4.7, it can be clearly seen that the LL_2 sub band which looks similar to the input image has the maximum energy while other sub bands mostly has near zero coefficients. The EZW algorithm also relies on the fact that if a coefficient is small in magnitude then the coefficients corresponding to the same spatial location in the higher sub bands will also tend to be smaller.

Since the descendants of a small coefficients will also be small, they are compared to a series of thresholds, T_k, $k = 0, 1, ..., K-1$. For coding purposes, the thresholds of the form $T_k = T_o 2^{-k}$ are considered where T_o must satisfy the follwing:

$$\frac{|y|_{max}}{2} < T_o \leq |y|_{max} \tag{4.10}$$

$$\frac{|y|_{max}}{3.5} < T_o \leq \frac{|y|_{max}}{2} \tag{4.11}$$

Here $|y|_{max}$ is taken as the maximum magnitude of all the coefficients of all the sub bands. For blurred and whitened images after simulations the quantization level values are determined given by Eqn.(4.11).

A particular coefficient $y = y[i, j]$ is said to be significant with respect to T_k if $|y| \geq T_k$ or equivalent notation can be:

$$\frac{|y|_{max}}{T_{K-1}} \geq \frac{T_k}{T_{K-1}} \tag{4.12}$$

By considering that the $\frac{T_k}{T_{K-1}}$ is an integer, then y is significant top T_k if and only if:

$$|q| \triangleq \frac{|y|}{T_{K-1}} \geq 2^{K-k-1} \tag{4.13}$$

So it can be considered that value of 'q' is the quantized coefficient using a step size value of $T_{K-1} = 2^{-(K-1)T_o}$ and also the 'K' bits are sufficient for coding 'q'. Let the binary representation of 'q' can be given as:

$$q = s, q_0, q_1, ..., q_{K-1} \tag{4.14}$$

From Eqn.(4.12), it can be concluded that the value 'y' is significant if and only if the magnitude of its quantized index is significant with respect to 2^{K-k-1} or equivalently if atleast one value of the $q_0, q_1, ..., q_k$ is 1.

4.5.4 Entropy encoding

This thesis implements bit plane coding based Embedded Zero tree Wavelet coding and Huffman coding of the EZW coded symbols. From the quantized results, it is proved

that the value of 'q' becomes significant in the bit k'. It implies that:

$$q_{k'} = \begin{cases} 1 & \text{if } \forall k > k' \\ 0 & \forall k < k' \end{cases} \tag{4.15}$$

The first bit $q_{k'}$ is called the 'significance bit' of 'q'. All sequences following this are called the 'refinement bits'.

The algorithm of EZW based bit plane coding is the entropy coding used to encode the quantized coefficients. Consider the size of the original image to be of N X N, where $N = 2^n$ and also $n > D$. This image is transformed with the D levels of a dyadic tree-structured sub band transform as on Fig.4.9. This transformation will yield $3D + 1$ subbands of transform coefficients. Each of the coefficient is then replaced by the index (in sign magnitude form) resulting from quantization with step size $T_{K-1} = T_o 2^{-(K-1)}$, where T_o is selected based on Eqn.(4.10). The resulting $N \times N$ array of integers is then 'sliced' into $K + 1$ bit planes. The first such plane consists of sign bit of each index, denoted by $s[i, j]$ where $i = 0, 1, 2..., N - 1$. The next plane consists of all the MSB's of each magnitude denoted by $q_o[i, j]$ and next plane consists of $q_1[i, j]$ and so on. The sign bit of a particular index is coded with the significance bit of that index. This can be explained as the sign bit $s[i, j]$ is coded with the $q_{k'}[i, j]$ where $q_{k'}[i, j] = 1$ and $q_k[i, j] = 0$ for all $k < k'$.

Each of the magnitude bit-planes are coded in two passes. The first passes the codes all refinement bits, while the second pass codes everything else (all the bits that are not coded in the first pass). This can be illustrated as the first pass codes a bit for each coefficient that is significant (due to its significance bit have been coded in a previous bit-pane. This is called 'refinement pass' since all the bits coded in this pass are refinement bits. The second

pass codes a bit for each coefficient that is not yet significant. This is called 'significant pass' since the a 1 coded in this pass indicates a coefficient becoming newly significant. The refinement pass was called the 'dominant pass' since the significance pass is the first to result in coded data and determines which coefficients are visited in the refinement(subordinate) pass of subsequent bit planes.

Significance pass:

In the significance pass each insignificant coefficient is visited in the raster order (left-to-right) and (top-to-bottom), first with LL_D, then with HL_D, LH_D, then HH_D, then HL_{D-1} and so on up to and including HH_1. Coding is done using 4 alphabets:

1) **POS= Significant Positive**. This symbol is equivalent to a 1 followed immediately by a corresponding 'positive' sign bit.

2) **NEG = Significant Negative**. This symbol is equivalent to a 1 followed immediately by a corresponding 'negative' sign bit.

3) **ZTR = Zero Tree Root**. This symbol indicates that the current bit of a particular coefficient is 0 and the corresponding bit in all its descendentats will also be zero.

4) **IZ=Isolated Zero**. This symbol indicates that the bit is 0 but atleast one of its descendent has a corresponding 1 bit.

The three highest frequency subbands HL_1, LH_1, HH_1 have no children. So while coding bits from these subbands, the ZTR and IZ symbols are replaced by single Z symbol. As the scan of insignificant coefficients progresses through subbands, any bit known already to be zero is not coded again. Also for the purpose of determining is a bit is a zero-tree root, only insignificant descendents are examined. Similarly the refinement bits of significant

descendants are treated as if 0 for the purpose of ZTR formation.

Refinement pass:

In this pass, a refinement bit is coded for each significant coefficient. A coefficient is significant if it has been coded POS or NEG in a previous bit plane. Its current refinement bit is simply its corresponding bit in the current bit plane. The order by which significant coefficients are visited during the refinement pass is: First by magnitude, Second by raster within subbands in order LL_D, HL_D, LH_D, HH_D, HL_{D-1},... HH_1.

Now let us go through a simple example to describe the EZW coding process. Here to explain the above mentioned coding procedure small sub image of 8 x 8 sample its taken. It is initially transformed to a 3 level dyadic subband decomposition. It is then quantized as explained in the above section with a series of threshold values each representing a particular sub band. Fig.4.10 represents the quantized coefficients.

Significant pass of q_0:

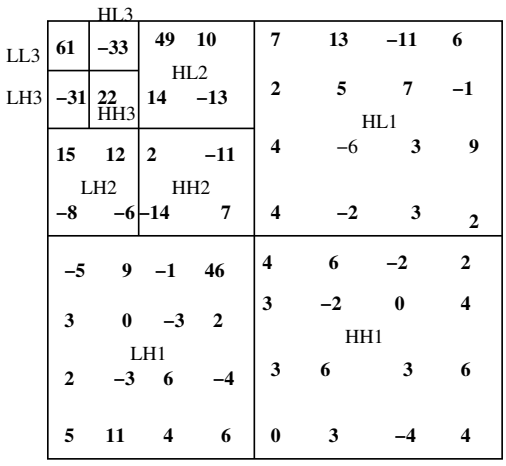

Figure 4.10: Quantised coefficients of a 3 level dyadic decomposition

Coding process begins with the significant pass through q_0. The single value in the LL_3

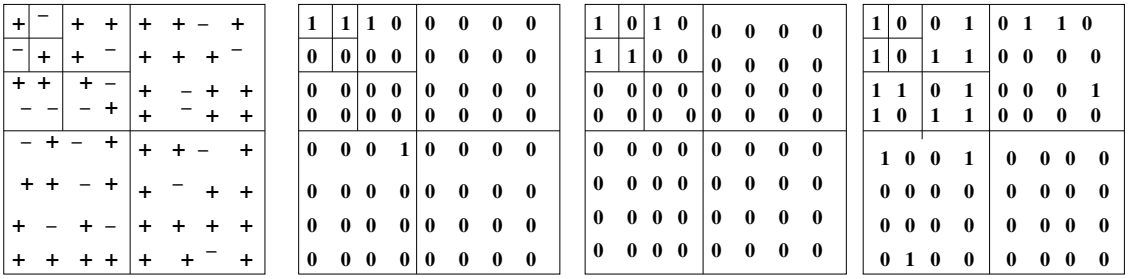

Figure 4.11: Bit plane Slicing

coefficient has a value of 1 and also corresponding sign bit is positive. So the first code symbol is **POS**. Similarly the HL_3 of the q_0 bit is coded as **NEG**. The single bit in LH_3 is coded as **IZ** since the corresponding bit is zero but it has a non-zero descendent in LH_1 of the q_0 bit plane. The bit in HH_3 is coded as **ZTR** indicating it and all its descendants are 0. It can also be noted that no further coding of HH_2, HH_1 will be done for q_0 bit plane. So the code for the first level sub band is:

POS, NEG, IZ, ZTR

Proceeding similar way to the HL_2 the first bit is coded as **POS** while the other three bits are coded with 3 **ZTR** symbols. Similarly LH_2 is coded as **ZTR, IZ, ZTR, ZTR**. HH_2 is already coded with all zeros and so it is skipped. Proceeding similar way to the sub band HL_1 it can be noted that out of 16 coefficients in this sub band, 12 has been already coded as zero by the zero tree from the HL_2. The four remaining bits are coded with **Z, Z, Z, Z**. Similarly there are only 4 uncoded bits in the LH_1, they are coded as **Z, POS, Z, Z**. Thus the code symbol for the 'Significant Pass' of the q_0 bit plane is:

POS, NEG, IZ, ZTR - From level D

POS, ZTR, ZTR, ZTR, ZTR, IZ, ZTR, ZTR- From level D-1

Z, Z, Z, Z, Z, POS, Z, Z - From level 0.

Refinement Pass:

By the Significant pass, there are four coefficients known to be significant.

Specifically $q[0,0], q[0,1], q[0,2], q[4,3]$. The code symbol for this pass is:

$q[0,0], q[0,1], q[0,2], q[4,3] = 1, 0, 1, 0.$

Significant pass of q_1: This codes all the remaining bits left from the previous refinement pass. So the codes are:

NEG, POS

ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR

Z, Z, Z, Z

Refinement pass of q_2:

The q_1 significant pass has added two more significant bits ($q[1,0]$ and $q[1,1]$). Here first the coefficients in the follwing order $|q^4 = 3|$, $|q^4 = 2|$ and $|q^4 = 1|$ are coded. Thus symbols coded in the refinement pass are:

$q2[0,0], q2[0,2], q2[0,1], q2[4,3], q2[1,0], q2[1,1] = 1, 0, 0, 1, 1, 0.$

Thus the coding procedures follows with alternating significant and refinement pass till it reaches the bit plane depth.

Thus the EZW coding codes the given coefficients into series of symbols in the significant pass and group of bit in the refinement pass. These symbols can be coded using a Arithmetic encoder or by simply assigning Huffman codes based on the probability of occurrence of each symbols. Here we use Huffman codes for encoding the symbols.

The compression results by applying the JPEG 2000 encoding methods for natural and blurred images are explained in chapter 7.

CHAPTER 5: SPARSE CODING OF IMAGES

This chapter analyzes the Sparse Code Learning algorithm [20]-[22]. This chapter begins by introducing the mathematical representations needed for sparse coding and explains in detail the sparse code learning algorithm. This chapter is divided into two stages: 1) Training the SCL algorithm for obtaining the base functions. 2) Applying the trained bases for compressing the images.

5.1 Mathematical representation

5.1.1 Image series expansion

An image can be better analyzed [55] as linear combination of expansion functions given by the following Eqn.(5.1):

$$f(x) = \sum_{k} \alpha_k \varphi_k \tag{5.1}$$

Here k is an integer index of the finite or infinite sum, the α_k are real valued expansion coefficients and the φ_k are real valued expansion functions. If the expansion is unique that is there is only one set of α_k for any given $f(x)$, then $\varphi_k(x)$ are called the ***basis functions*** and the expansion set $\varphi_k(x)$ is called a basic for the class of functions that can be so expressed. The expressible functions form a *function space* that is referred to as the closed span of the expansion set denoted by Eqn.(5.2):

$$V = \overline{Span_k(\varphi_k(x))} \tag{5.2}$$

For any function space V and corresponding expansion set $\varphi_k(x)$, there is a set of dual functions denoted by $\varphi'_k(x)$ that can be used to compute the α_k coefficients for any $f(x) \varepsilon V$

by Eqn.(5.3):

$$\alpha_k = \langle \varphi'_k(x), f(x) \rangle = \int \varphi'^*_k(x) f(x) dx \tag{5.3}$$

Here * represents the complex conjugate operation. Depending on the orthogonality of the expansion set, this computation takes on the following three forms.

Case 1:

If the expansion set forms an orthonormal basis for V, then

$$\langle \varphi_j(x), \varphi_k(x) \rangle = \delta_{jk}$$

$$\delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

the basis and its dual are equivalent. That is, $\varphi_k(x) = \varphi'_k(x)$.

Case 2:

If the expansion functions are not orthonormal but are an orthogonal basis for V, then $\langle \varphi_j(x), \varphi_k(x) \rangle = 0$ if $j \neq k$

The basis functions and their duals are called biorthogonal.

$\langle \varphi_j(x), \varphi'_k(x) \rangle =$

$$\delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

Case 3:

If the expansion set is not a basis for V, but supports the expansion, it is a spanning set in which there is more than one set of α_k for any $f(x) \in V$. The expansion sets and their duals are called ***Overcomplete*** or ***redundant***.

They form a frame in which some $A > 0$, $B < \infty$ and $f(x) \in V$.

$$A\|f(x)\|^2 \leq \sum_k |\langle \varphi_k, f(x) \rangle|^2 \leqq B\|f(x)\|^2$$

The overcomplete set explained forms the main coding interest. When the number of code elements is greater than the effective dimensionality of the input space it is called ***over-complete*** representation. Here the basis functions are non-orthogonal and not linearly independent of each other. So coding challenge is to sparsify the code that is to select those basis functions necessary for representing given input. This also implies that the input -output relation is not purely linear.

5.1.2 General coding model

The general form of coding strategy is to apply a linear transform such as DCT or Wavelet transform as already discussed by taking inner product of the image, $I(\vec{x})$ with a set of spatial weighing functions ψ as per Eqn.(5.4):

$$b_i = \sum_{\vec{x_j}} \psi_i(\vec{x_j}) I(\vec{x_j}) \tag{5.4}$$

$\vec{x_j}$ denotes the discrete spatial position within the image. The output of the transformation is represented by b_i. In matrix notation it is represented as Eqn.(5.5):

$$b = WI \tag{5.5}$$

The JPEG and JPEG 2000 based compression are based on this transformation. The matrix W is invertible and is chosen so that the transformation b satisfies certain criterion like sparseness, decorrelation etc.

5.1.3 Generative image model - probabilistic approach

The sparse coding strategy is based on using a probabilistic structure to capture the image structure [21]. Here images are represented as linear superposition of basis functions. The basis functions are adapted so as to best account for the image structure in terms of statically independent events. The probability distribution of these events are sparse meaning that a given image can be represented in terms of small number of basis functions chosen out of a larger set.

The alternate way of coding (different from Eqn.(5.4)) is to represent the image in terms of a linear superposition of basis functions $\phi_i(\vec{x})$ mixed together with the amplitudes a_i given by Eqn.(5.6):

$$I(\vec{x}) = \sum_i a_i \phi_i(\vec{x}) \tag{5.6}$$

The basis functions $\phi_i(\vec{x})$ are chosen such that they form an overcomplete representation of the given input image. The coefficients a_i are calculated for each image. The choice of the basis function ϕ_i becomes very important such that it should best capture the independent structures in images. Fig.5.1 represents this structure.

5.2 Sparse coding algorithm

The sparse coding algorithm is based on the Sparse Code Learning algorithm proposed by Bruno [21]. Based on the generative model [20], the given problem is reduced to determine the basis function that can best account for the structure in images in terms of sparse statistically independent events. The aim is to match, as closely as possible the distribution

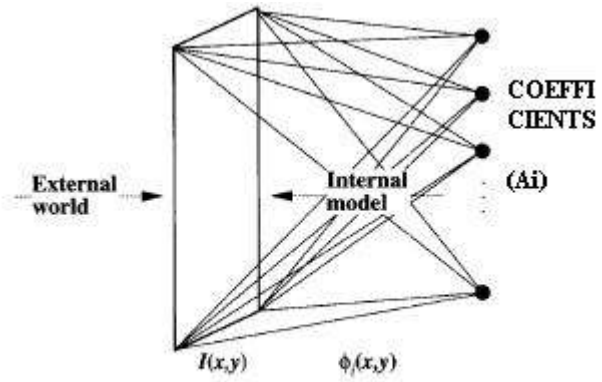

Figure 5.1: Generative image model

of images arising from the linear image model given by $P(I|\phi)$ to the actual distribution of images in nature $P^*(I)$. The coefficients a_i are considered to be sparsely distributed, which means the distribution has more number of zeros with extending tails. The problem now can be interpreted as analyzing the distribution of the image when the coefficients are assumed and how to make it resemble the actual distribution of the images. This is the challenge in sparse coding techniques, where the basis functions are trained and adapted, so that the coefficients needed to represent the image structure follows sparse distribution and also the generated image matches with the real world distribution of the images.

The probability of an image arising from the model is given by Eqn.(5.7)

$$P(I|\phi) = \int P(I|a,\phi)da \tag{5.7}$$

The probability of an image arising out of particular choice of coefficients, $P(I|a,\phi)$ expresses the uncertainty or level of noise in the imaging process. Consider the noise to have Gaussian distribution such that noise added is Additive White Gaussian Noise (AWGN). So the image model can be given by Eqn.(5.8)

$$I(\vec{x}) = \sum_i a_i \phi_i(\vec{x}) + v(\vec{x}) \tag{5.8}$$

The probability of an image arising from particular choice of coefficients a, is given by Eqn.(5.9)

$$P(I|a,\phi) = \frac{1}{Z_{\sigma_N}} e^{-\frac{(|I-a\phi|)^2}{2\sigma_N^2}} \tag{5.9}$$

$$|I - a\phi|^2 = \sum_{\vec{x}} \left[I(\vec{x}) - \sum_i a_i \phi_i(\vec{x})\right]^2$$

where σ_N^2 is the noise variance and Z_{σ_N} is the normalizing constant.

Since the basis function is overcomplete there will be a infinite number of a's to represent the given image. So the distribution $P(I|a,\phi)$ takes the form of Gaussian ridge along the line where $I = a\phi$.

The sparseness is introduced in the distribution of coefficients. Statistical independence is incorporated by specifying P(a) to be factorial distribution in 'a_i'.

$$P(a) = \prod_i P(a_i) \tag{5.10}$$

Thus the probability of any state 'a' of the coefficients is simply given by the product of individual probabilities of each component 'a_i'. The sparseness is incorporated by shaping

the distribution of each 'a_i' to be uni-modal and peaked at zero with heavy tails. This distribution can be given by:

$$P(a_i) = \frac{1}{Z_\beta} e^{-\beta S(a_i)} \tag{5.11}$$

where function S determines the shape of the distribution and β controls the steepness. Z_β is Normalizing constant.

Assume $\beta = 1$, $S(x) = \log(1 + x^2)$. This corresponds to cauchy density for the prior that has the sparse shape.

To determine the matching of the model with the real world distribution of images, log likelihood or **Kullback-Leibler** divergence between the two distributions are taken.

$$KL = \int P^*(I) \log(\frac{P^*(I)}{P(I|\phi)}) dI \tag{5.12}$$

This measures the average information gain per image drawn from $P^*(I)$, for judging in favour of the image being drawn from $P^*(I)$ as opposed from $P(I|\phi)$. The value of KL will be zero if and only if the two distributions are equal. If the difference between the two distribution are more, then the value of KL will be greater. So the aim is minimize the value of KL. Minimizing KL can be interpreted as maximizing $\langle \log(P(I|\phi)) \rangle$ since $P^*(I)$ is fixed and also by:

$$\langle \log(P(I|\phi)) \rangle = \int P^*(I) \log(P(I|\phi)) dI \tag{5.13}$$

Thus the goal of learning is to find a set of ϕ that maximizes the log-likelihood of the images under a sparse, statistically independent prior.

5.2.1 Learning or training

The problem is to determine the learning set of basis functions ϕ for the image model that best accounts for image in terms of sparse, statistically independent components [18]-[20]. The goal as described in probabilistic framework is to find a set of bases, ϕ^*, such that

$$\phi^* = \arg max_{\phi} \langle \log(P(I|\phi)) \rangle \tag{5.14}$$

The evaluation of $P(I|\phi)$ requires integrating over all possible states of 'a' which makes it intractable and difficult to implement. So assume that the function inside the integral $P(I, a|\phi) = P(I|a, \phi)P(a)$ has a peaked maximum in a-space. So this approximates the relation and by evaluating $P(I, a|\phi)$ only at its maximum, the expression reduces to find ϕ^* such that it becomes as Eqn.(5.15).

$$\phi^* = \arg max_{\phi} \langle max_a \log(P(I|a, \phi)P(a)) \rangle \tag{5.15}$$

In order to analyze the effects of this approximation, rewrite the objective equation in energy function framework given by Eqn.(5.16)

$$E(I, a|\phi) = -\log(P(I|a, \phi)P(a)) \tag{5.16}$$

So Eqn.(5.15) can be restated as Eqn.(5.17),Eqn.(5.18)

$$\phi^* = \arg min_{\phi} \langle min_a E(I, a|\phi) \rangle \tag{5.17}$$

$$E(I, a|\phi) = \sum_{\vec{x}} \left[I(\vec{x}) - \sum_i a_i \phi_i(\vec{x})\right]^2 + \lambda \sum_i S(a_i) \tag{5.18}$$

This equation is obtained by using the expressions for $P(I|a, \phi)$ and $P(a)$ from equations Eqn.(5.9),Eqn.(5.10),Eqn.(5.11) and by substituting $\lambda = 2\sigma_N^2\beta$.

The function to be minimized $E(I, a|\phi)$ is now reduced to sum of two terms: the first term computes the reconstruction error that forces the ϕ to span the input space and the second terms handles the coefficient activities that makes it sparse. E is minimized in separate phases, one nested within the other. The inner phase minimizes E with respect to a_i for each image, while keeping ϕ_i constant. The outer phase over many image presentations E is minimized with respect to ϕ_i.

The inner loop minimization over the coefficients a_i is performed by iterating until the derivative of $E(I, a|\phi)$ with respect to a_i Eqn.(5.19) becomes zero. For each image the a_i are determined by solution to the differential equation.

$$a_i = \sum_{\vec{x}} \phi_i(\vec{x}) r(\vec{x}) - \lambda S'(a_i) \tag{5.19}$$

where $r(\vec{x})$ is the residual image given by Eqn.(5.20)

$$r(\vec{x}) = I(\vec{x}) - \sum_i a_i \phi_i(\vec{x}) \tag{5.20}$$

Thus Eqn.(5.19) makes a_i to depend on summation of two terms. The first terms takes a spatially weighted sum of current residual image using the function $\phi_i(\vec{x})$ as weights.

The second term applies a non-linear inhibition on a_i that pushes its activity toward near zero values making it sparse.

The outer loop minimization over the ϕ_i is accomplished by simple gradient descent. This gives rise to the learning rule given by Eqn.(5.21)

$$\Delta\phi_i(\vec{x} = \eta\langle a_i r(\vec{x})\rangle) \tag{5.21}$$

where η is the learning rate. ϕ_i are updated by learning between the ouputs computed for each image a_i and the residual image $r(\vec{x})$. By this method the ϕ_i will grow without bounds. To prevent this, the L2 norm of each basis function given by Eqn.(5.22),Eqn.(5.23) is separately adapted such that the output variance of each a_i is held at appropriate level.

$$l_i^2 = \sum_{\vec{x}} |\phi_i(\vec{x})|^2 \tag{5.22}$$

$$l_i^{\text{new}} = l_i^{\text{old}}[\frac{\langle a_i^2\rangle}{\sigma^2_{\text{goal}}}]^{\alpha} \tag{5.23}$$

where σ^2_{goal} is the desired variance of the coefficients.

Thus the algorithm can be interpreted as that on each image presentation the gradient of 'S' sparsifies the distribution activity by differentially reducing the value of low activity coefficients than the high activity coefficients. The ϕ_i will then learn on the error induced by this sparsification process and adapt itself such that it results in a set of bases that on reconstruction with the sparse coefficients will result in minimum mean square reconstruction error. When the basis set is overcomplete and non-orthogonal, the effect of sparsification is

to choose which bases are most effective for describing a given image structure. Thus the system tries to infer which bases are most appropriate for describing a given image.

5.3 Simulation methods

The datas for training to get the base functions were taken from natural image set. These images are blurred and whitened as already explained. The datas for training were chosen from the image set by randomly selecting a sub image of size 8 x 8. The pixels in the corner of the image are avoided. Initialize the base by taking a random value of 64 x 64 matrix and then normalizing it. The a_i co-efficients are computed using Eqn.(5.24)

$$a_i = \sum_{\overrightarrow{x}} \phi_i(\overrightarrow{x}) I(\overrightarrow{x}) \tag{5.24}$$

Iterate the above equation using conjugate gradient method [22], halting after 10 iterations or whenever there is very slight changes in the a_i coefficients. The set of base functions was updated according to Eqn.(5.21),Eqn.(5.23) based on averages computed every 100 image presentations. The learning rate parameter η was initially set to a higher value and then slowly decreased as the number of iterations increases and then finally reduced to 1.0. The rate parameter for the gain adjustment α was set to 0.01 and target output variance σ^2_{goal} was set to variance of the image pixels. The form of sparseness introduced was $S(x) = log(1 + x^2)$. A stable result will be obtained after iterating for around 300 000 image samples. The basis functions obtained for different imagesets are shown in Fig.5.2, Fig.5.3, Fig.5.4 and Fig.5.5. It can be clearly seen that the basis functions are not properly trained or evolved to definite structures for both the case of natural and blurred images.

For whitened images and blurred whitened cases, we can clearly see that the base functions typically looks like log Gabor functions varying in orientations.

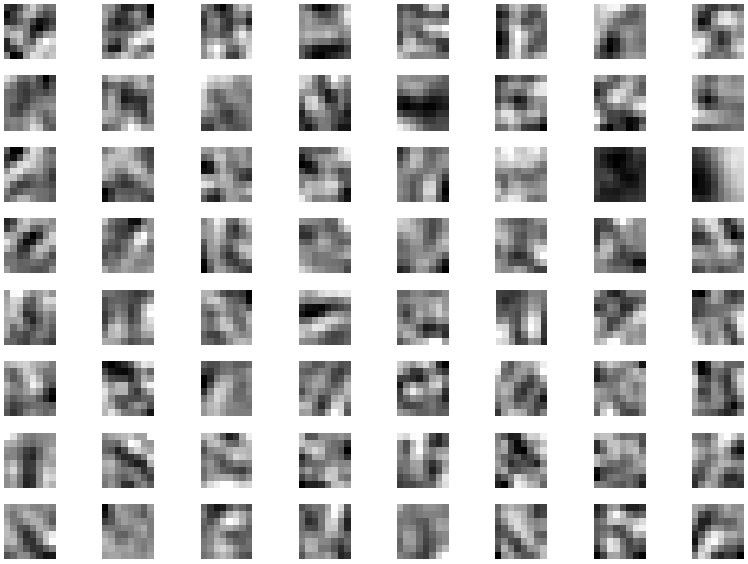

Figure 5.2: Learned bases function for natural images

5.4 Application towards image compression

The initial step of the algorithm discusses the training methods to determine the base functions. Once the base functions are obtained for a given set of images, they can applied to any given image for obtaining the coefficients. The coefficients are obtained for any given image using similar equations as discussed in the training algorithm except for the update step of base function. The advantage of this method is that only coefficients are required to represent any given image and since these coefficients follows a Sparse distribution, they clearly provide an insight towards image compression. The coefficients were determined in

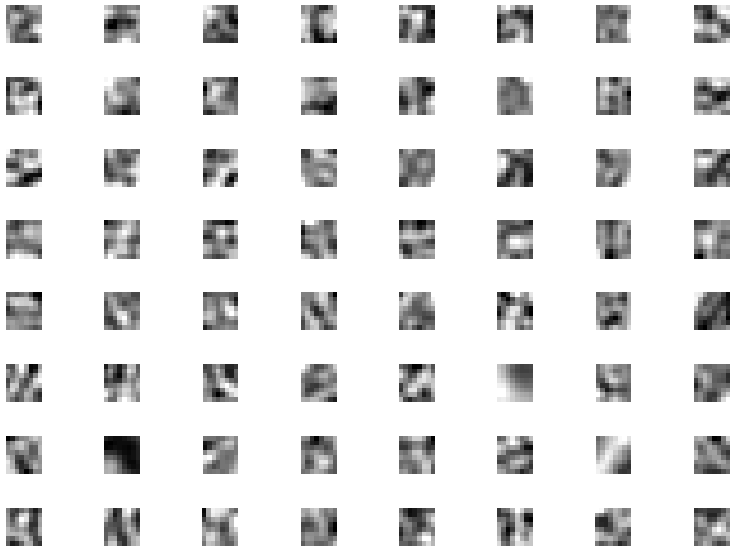

Figure 5.3: Learned basis function for blurred images

a similar way as in the inner loop iterations of the training section and avoiding the outer loop base functions updating steps.

For an example for a given natural image the coefficients were determined for it. Fig.5.6 shows the distribution of the obtained coefficients. Once the coefficients are obtained they can be quantized using either uniform or nonuniform quantization algorithms discussed in detail in next section. These quantized coefficients are then source encoded using Huffman source encoder. For communication or transmission applications it is essential to quantize and transmit the basis function also in a similar manner. But for vision purposes the base functions are not considered for computing the bit rates. This work doesnot include the transmission of base functions.

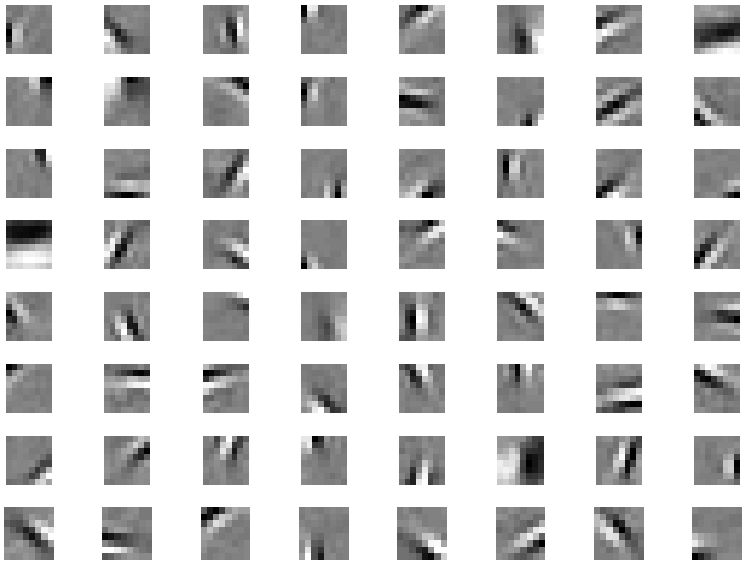

Figure 5.4: Learned basis function for whitened and normalised images

In the decoder it becomes simple as when the coefficients are decoded they can be directly used to reconstruct the image by using the Eqn.(5.24). since the images are linear superposition of the coefficients with the base functions.

5.5 Quantization methods

The quantization methods analyzed for quantizing the coefficients are uniform and non-uniform quantization. Though there are different algorithms proposed in literature to obtain minimum quantization error, the above mentioned algorithms were chosen due to its simplicity in implementation.

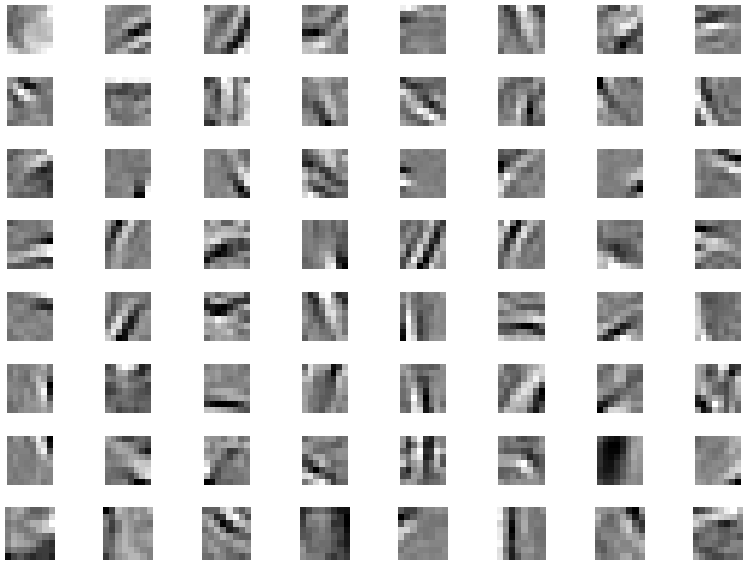

Figure 5.5: Learned basis function for blurred whitened and normalised images

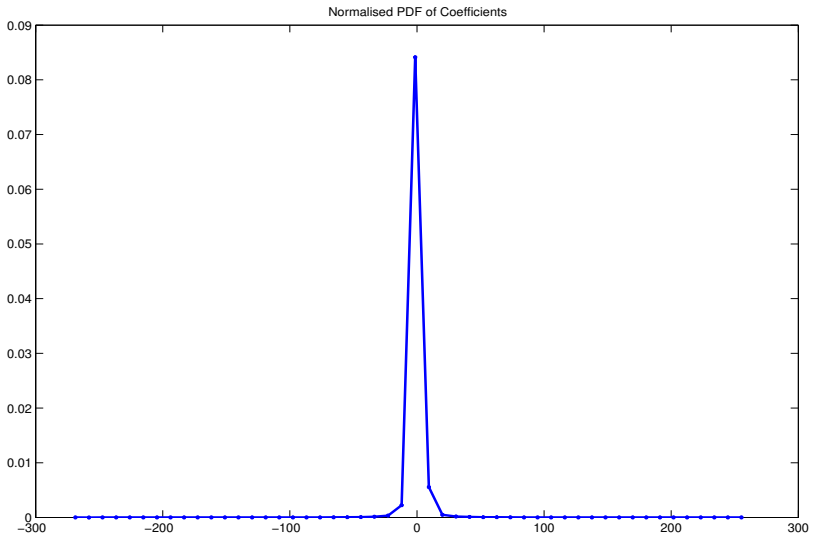

Figure 5.6: PDF of coefficients from simulation

5.5.1 Uniform quantization

In this method [57][58] the range of the values to be quantized is divided into 'Q' intervals of equal length taken as λ (according to the PDF of the data). If any of the data falls between the quantizing intervals x_i and x_{i+1}, then the quantized value is taken as the midpoint value m_i between x_i and x_{i+1}. Fig.5.7 represents this with the help of a Gaussian PDF. Similarly it can be used for the coefficients that has the Sparse PDF. If 'a' and 'b' are the minimum and maximum values of the the coefficients then the step size can be given as

$$\lambda = \frac{(b-a)}{Q} \tag{5.25}$$

The quantized output x_q is generated according to:

$$x_q = \begin{cases} m_i & \text{if } x_{i-1} < X \leq x_i \end{cases}$$

where $i = 1, 2, ..., Q$

$$x_i = a + i\lambda \tag{5.26}$$

$$m_i = \frac{(x_{i-1} + x_i)}{2} \tag{5.27}$$

5.5.2 Non uniform quantization

The RMS error from the uniform quantization is high because the PDF of the data is not uniform and also it is highly distributed towards the region near zero and very few datas in other areas. So the quantization intervals should be varied in accordance to the distribution of the data [57]. Here the quantizer step size should be varied with smaller step sizes near the mode of the PDF of the data and larger tails near the tails of the PDF. Fig.5.8

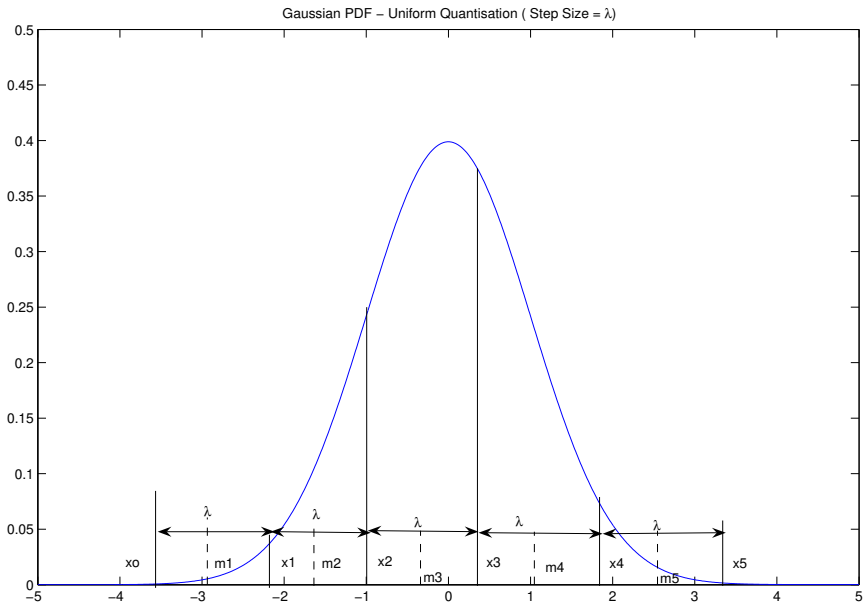

Figure 5.7: Uniform quantisation

explains this method with the help of a Gaussian PDF. This can be similarly applied to the coefficients which follows the sparse PDF.

The input to the quantizer is the data from the sparse PDF and the quantizer output is determined according to:

$$x_q = \begin{cases} m_i & \text{if } x_{i-1} < X \leq x_i \end{cases} \tag{5.28}$$

where $x_0 = -\inf$ $x_Q = \inf$ where $i = 1, 2, ..., Q$

The step size $\lambda_i = x_i - x_{i-1}$ is variable. The design can be theoretically approached as follows. The data X is given with a PDF of $f_x(x)$. This data X has to be approximated by the discrete value X_q in accordance with the Eqn.(5.28) The quantizing intervals and the levels are to be chosen such that the noise value N_Q has to be minimized. The minimizing

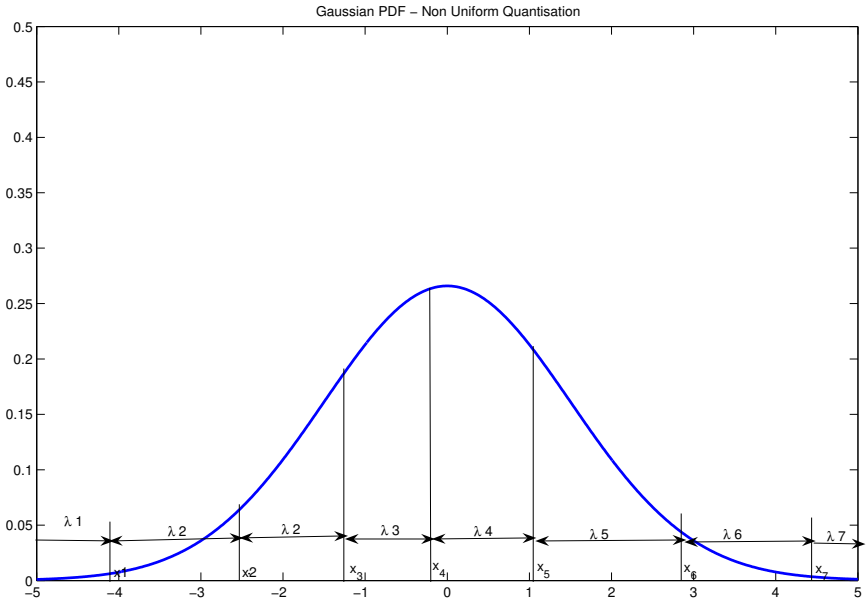

Figure 5.8: Non uniform quantization

can be done as follows:

$$N_Q = \sum_{i=1}^{Q} \int_{x_{i-1}}^{x_i} (x - m_i)^2 f_x(x) dx \tag{5.29}$$

Since N_Q is to be minimized, the necessary conditions to achieve this can be obtained by differentiating N_Q with respect to the x_j and m_j and setting the derivatives to zero:

$$\frac{dN_Q}{dx_j} = (x_j - m_j)^2 f_x(x_j) - (x_j - m_{j+1})^2 f_x(x_j) = 0 \tag{5.30}$$

where $j = 1, 2, ...Q - 1$

$$\frac{dN_Q}{dm_j} = -2 \int_{x_{j-1}}^{x_j} (x - m_j) f_x(x_j) = 0 \tag{5.31}$$

where $j = 1, 2, ..., Q$

$$x_j = \frac{(m_j + m_{j+1})}{2}$$

$$m_j = 2x_{j-1} - m_{j-1} \tag{5.32}$$

where $j = 2, 3, ..., Q$

Eqn.5.31 can be reduced as

$$\int_{x_{j-1}}^{j} (x - m_j) f_x(x) dx = 0 \tag{5.33}$$

where $j = 1, 2, ...Q$

This implies that m_j is the mean of the j^{th} quantizer interval. For a specific PDF the method of solving the above mentioned equations are by picking m_1 and calculate the succeeding x_i and m_i. If m_1 is chosen correctly then at the end of the iteration, m_Q will be the mean of the interval $[x_{Q-1}, \inf]$. If m_Q is not the mean of the Q^{th} interval then a different choice of m_1 is chosen and repeated with the corresponding suitable values of x_i and m_i. Fig.7.2 in the Chapter 7 shows the improvement in performance of using a non-uniform quantizer.

5.6 Results

After quantization each bin level is considered as a symbol for the Huffman source encoder. The quantized values are assigned Huffman codes. In communication based systems it is essential to transmit both the base functions as well as the coefficients. But vision based systems does not need the base functions. We consider only quantizing and encoding the coefficients.

The SCL algorithm is trained using different image sets such as:

1) Natural image set.

2) Blurred image set (blurred by PSF function).

3) Whitened and normalized image set.

4) Blurred whitened and normalized image set.

The compression results for these image sets are tabulated and compared in chapter 7.

CHAPTER 6: GABOR FILTERS AND INHIBITION BASED SPARSE CODING

This chapter begins by introducing the Gabor and log Gabor filters and then analyzes the methods to design a log Gabor filter bank [1][34][35][40]. It explains the methods to construct an overcomplete representation of the given image by applying the constructed filter bank. It also discusses the inhibition based sparse coding algorithm applied for selecting the coefficients from the overcomplete set [25][27].

6.1 Log Gabor filters

It has been shown by Field [1] that Gabor filters are optimal for obtaining localized frequency information. Gabor filters provides very good localization of spatial and frequency information. But the limitations are that the maximum bandwidth of a Gabor filter is limited to approximately one octave [1] and they are not optimal for broad spectral information representation with maximal spatial localization [1]. An alternative to Gabor function is the log-Gabor function proposed by Field [1]. They can be constructed for arbitrary bandwidth and can also be optimized to produce spatial localization [29].

It has been proven that natural images are better coded by filters that have Gaussian transfer functions when viewed on the log frequency scale [1]. The transfer function of log-Gabor function is given by Eqn.(6.1)

$$G(\omega) = \exp(\frac{-log(\frac{\omega}{\omega_o})^2}{2log(\frac{k}{\omega_o})^2}) \tag{6.1}$$

where ω is the frequency scale, ω_o is the filter's center frequency. k is held constant for varying values of $\frac{k}{\omega_o}$. Fig.6.1 shows a simple plot of the log Gabor transfer function for the value of $\frac{k}{\omega_o} = 0.74$. The important aspect of this function is that unlike Gabor functions, the frequency response of the log Gabor is symmetric on a log axis [1]. Moreover the log axis is the standard way of representing the spatial-frequency response of the visual neurons.

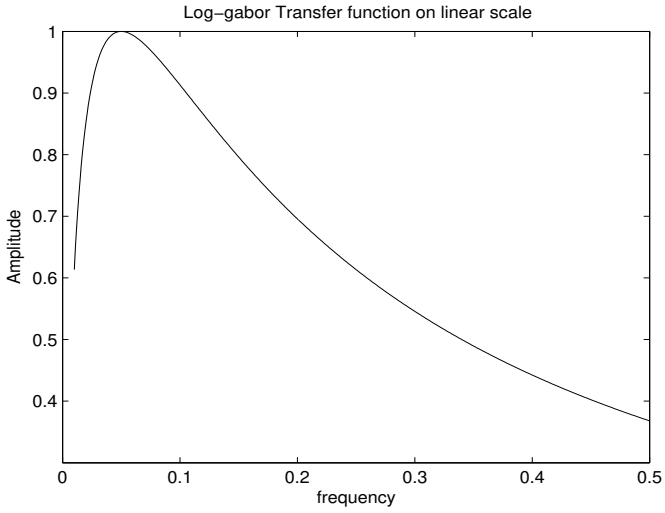

Figure 6.1: Transfer function of log Gabor filter on linear scale

The log Gabor functions has no DC component and also the transfer function has an extended tail at the high frequencies. Field's investigation of natural images have amplitude spectrum that falls approximately $\frac{1}{f}$ [1]. So to encode such images the filters must have similar filter characteristics. Field [1] proved in his work that log Gabor functions having extended tails, should be able to encode natural images more efficiently. The problem with Gabor functions are that they over represent low frequency components and under represent high frequency components which is not an efficient coding method. Also from [2][8], it is

consistent with measurements on mammalian visual systems that indicates cell responses are symmetric on the log frequency scale.

6.1.1 Design of log Gabor filter bank

Due to singularity in the log function at the origin, the shape of the log Gabor function cannot be constructed in spatial domain [34][39][40]. So they are designed in frequency domain and then by IFFT (Inverse Fast Fourier Transform), the equivalent spatial domain can be obtained [39]. Both Gabor and log Gabor filters looks similar for bandwidths less than one octave. As the bandwidth increases the shape of the log Gabor filters becomes much sharper [39].

A Gabor or log Gabor filter bank [35] does not form an orthogonal basis set and hence there is no standard arrangement of the filters. So the aim is to produce a filter bank that provides an even coverage of the spectrum to be represented (the spectrum of the image). This can be achieved by making the overlap of the filter transfer functions sufficiently large such that when one sums the individual transfer functions the net result is the even coverage of the given spectrum [39]. The problem in building this kind of filters is that all points in the spectrum should be covered and also minimum number of filters should be used for efficient coding. The output of each of the filters should be as independent as possible. The filter bank's aim is to obtain maximum information from the given input signal which could be possible only if the outputs of each of the filters are not much correlated to each other [38]. So to achieve the independence of output of filters the filters should have minimum overlap of their transfer functions. So as a conclusion of the issues considered the filter bank

must have independent transfer functions and also must provide even coverage of the given spectrum [38].

The design of the filter bank [39] depends on the following parameters:

1) Minimum and maximum frequencies needed to be covered.

2) Filter bandwidth.

3) Scaling between the center frequencies of successive filters

4) Number of filter scales

5) Number of filter orientations

6) The angular spread of each filter.

The maximum frequency is set by the wavelength of the smallest scale filter. The smallest value one can choose is the Nyquist wavelength of 2 pixels but even at this frequency there is significant amount of aliasing and so minimum value of 3 or above number of pixels are used. The minimum frequency is set by the wavelength of the largest scale filter. This is defined by the number of filter scales, scaling between center frequencies of successive filters and the wavelength of the smallest scale filter. This is given by the following equations Eqn.(6.3),

$$MaximumWavelength = MS^{N-1} \tag{6.2}$$

$$Minimumfrequency = \frac{1}{MaximumWavelength} \tag{6.3}$$

where M - Minimum pixels (taken as 3 pixels)

N - Number of filter scales

S - Scaling between center frequencies of successive filters.

Table 6.1: Bandwidth and scaling of filters

Filter bandwidth	**Scaling between filters**
0.85	1.3
0.75	1.6
0.65	2.1
0.55	3

Filter bandwidth is set by defining the ratio of the standard deviation of the Gaussian describing the log Gabor filter's transfer function in the log-frequency domain to the filter center frequency. This is given by the parameter G and the value taken is 0.65. The scaling between the center frequencies of successive filters plays a major role in deciding the even spectral coverage. So with the trade of between even spectral coverage and independence of filter output this parameter is carefully chosen by simulation trials. The angular overlap of the filter transfer function is controlled by the ratio of the angular interval between the filter orientations and the standard deviation of the angular Gaussian spreading function. This parameter is taken as R and the value is 1.5 for minimal overlap and maximum spectral coverage from standard results [38]. Table.6.1 shows the computed parameter values which provides design of filter banks with minimal overlap necessary to achieve even spectral coverage [38].

Number of filter orientations needed is in conjunction with the angular spread of the filter. It has been shown [35] that six orientations can be used for coverage of image spectrum without much loss in information. The angular spread of each filter has to be determined carefully by considering the trade off between the even spectral coverage and independence

of filter output. The angular interval between the filter orientations is fixed by the number of filter orientations. Fig.6.2 shows the filter bank that provides even coverage of spectrum with minimum number of filters. It uses filters varying in four orientations and four different scales to provide an even coverage of the image spectrum. For better results a very low pass filter and a high pass filter with very high cut off is chosen to extract the left out very high frequncy fine details in the image spectrum.

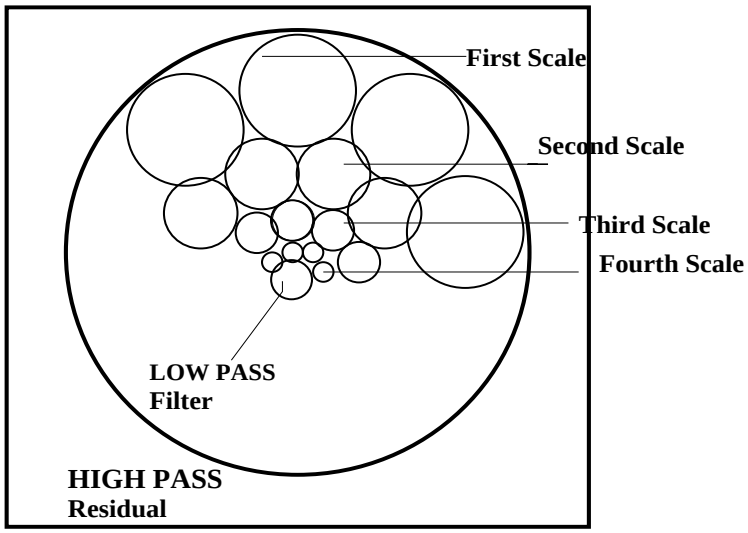

Figure 6.2: Log Gabor filter bank for even coverage of image spectrum

6.1.2 Filter construction

It is already explained that the log Gabor filters cannot be directly designed in spatial domain [39]. So it is designed in frequency domain and then converted to spatial domain by applying IFFT. Filters are constructed same size as of the input image considered.

Filters are constructed in terms of two components:

1) Radial component: This controls the frequency band response of the filter.

2)Angular Component: This controls the orientation response of the filter.

These two components are multiplied together to get the overall response. For small wavelengths, the filters can extend into the higher frequencies in the corners of the frequency domain whereas it is cut off in the vertical and horizontal directions. To overcome this problem and to make uniform coverage in all directions, the filters are multiplied by a low-pass filter with very high cut-off frequency. Fig.6.3 represents the radial component of log Gabor filter designed with $wavelength = 3$ and $centerfrequency = 0.33$ and Fig.6.4 represents the angular component with orientation angle of 0.8 radians. Fig.6.5 shows the frequency domain representation of the log Gabor filter formed by the product of angular and radial component.

Table 6.2: Filter design parameters

Filter Parameters	**Values**
Number of Scales	4
Number of Orientations	4
Minimum Wavelength (In pixels)	3
Scaling Between centers of successive filters (S)	2
Parameter R	1.5
Parameter G	0.65

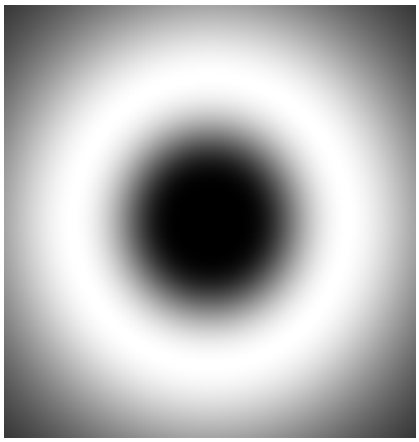

Figure 6.3: Radial component of log Gabor filter

A filter set is constructed by varying the filter in four different scales and four different orientations. This gives the combination of 16 filters varying in scale and orientation that altogether provides an even spectral coverage. Fig.6.6 represents the filter set constructed with four different scales and four different orientations.

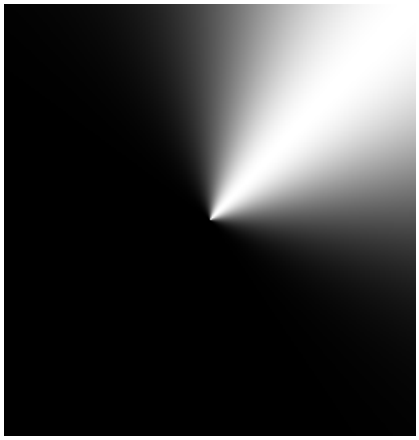

Figure 6.4: Angular component of log Gabor filter

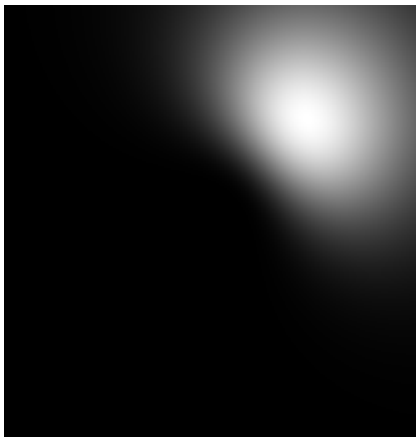

Figure 6.5: Product of angular and radial component log Gabor filter

6.2 Forming overcomplete transform pyramid

This section discusses the construction of an overcomplete pyramid representation of the given image using the filter bank. The method of compression is to filter the given image

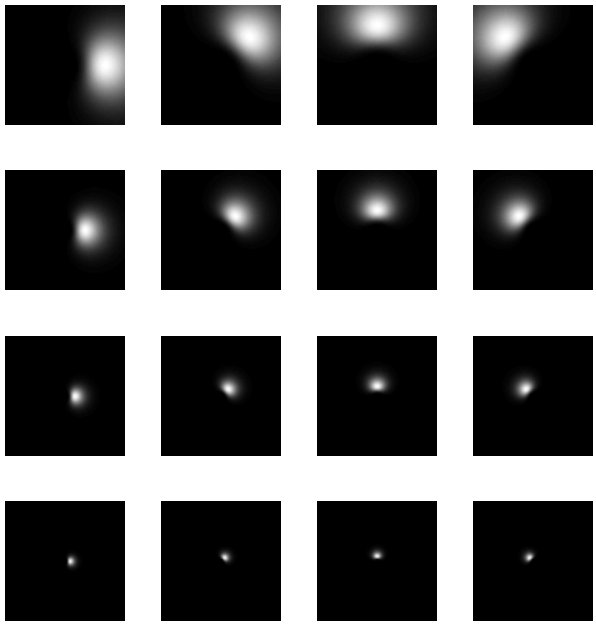

Figure 6.6: Log Gabor filter set varying in 4 scales and 4 orientations

with the filter set generated to form the transform pyramid and then to apply the inhibition algorithm [27]-[29]. The following notations and equations describes the filtering operation. Here I is denoted as the input image, J is its corresponding Fourier transform. Wo denotes its linear log Gabor transform. g_i represents the log Gabor filter in the frequency domain and G_i denotes the log Gabor filter in the spatial domain.

Let C_i denotes different channel outputs obtained by the operation of convolution between image I and the filters and downsampling the results:

$$C_i = \downarrow_i (G_i \otimes I) \tag{6.4}$$

Here $\otimes$ represents the convolution operation. The transformation W_o can be calculated in the frequency domain as:

$$W(I) = (C_i)_i = (F^{-1}(\downarrow_i (g_i J)))i \tag{6.5}$$

The inverse W^{-1} also called reconstruction is given by:

$$W^{-1}(C) = \sum_i (\uparrow C_i) \otimes G_i \tag{6.6}$$

In frequency domain:

$$W^{-1}(C) = F^{-1}(\sum_i (\uparrow c_i) g_i) \tag{6.7}$$

The reconstruction is possible if and only if $W^{-1}(W_o(I)) = I$. So this implies that

$$\sum_i (g_i)^2 = 1 \tag{6.8}$$

Thus the square sum of the filter set must be equal to one. Since the log Gabor filter set is designed in considering these factors, complete reconstruction can be achieved [27]. The method of constructing the transform domain [27], i.e. the pyramid structure can be explained by the following steps:

1) The given image is transformed to frequency domain by FFT (Fast Fourier Transform)

2) The first scale log Gabor filters are shifted (since they are designed keeping center as origin).

3) They are then multiplied by the given image. This operation is equivalent to convolution of the image with the filter.

4) By IIFT (Inverse Fast Fourier Transform) convert the image back to the spatial domain. This forms the first layer of the pyramid structure. This operation is repeated for all the orientations to form the first layer in the pyramid structure.

5) The steps from 1 to 4 are repeated for the second layer in the pyramid structure but now the second scale log Gabor filters are used and also the resulting filtered images are downsampled once.

6) Similarly third and fourth layer of the pyramid structure are formed by correspondingly downsampling twice and thrice the filtered images.

7) Although the log Gabor filters are designed for a even coverage of the image spectrum, yet a low pass filter with very low cut off frequency is used to pass the luminance and a high pass filter is used for encoding very high frequency components. For blurred retinal images this can be neglected.

8) The top layer of the pyramid is formed by filtering the input image with the low pass filter and then downsampling it four times and also by filtering the image with the high pass filter. Fig.6.7 represents this process in detail.

6.2.1 Reconstruction

It has been shown that perfect reconstruction of the filtered image from the transform domain can be possible based on the Eqns.(6.4 - 6.7). The method to reconstruct is explained below and Fig.6.10 details this procedures:

1) The low pass filtered image is upsampled using linear interpolation method to the size of the original image. It is converted to the frequency domain by FFT and filtered by the same low pass filter.

2) The high pass filtered image is converted to the frequency domain by FFT and filtered back by the high pass filter.

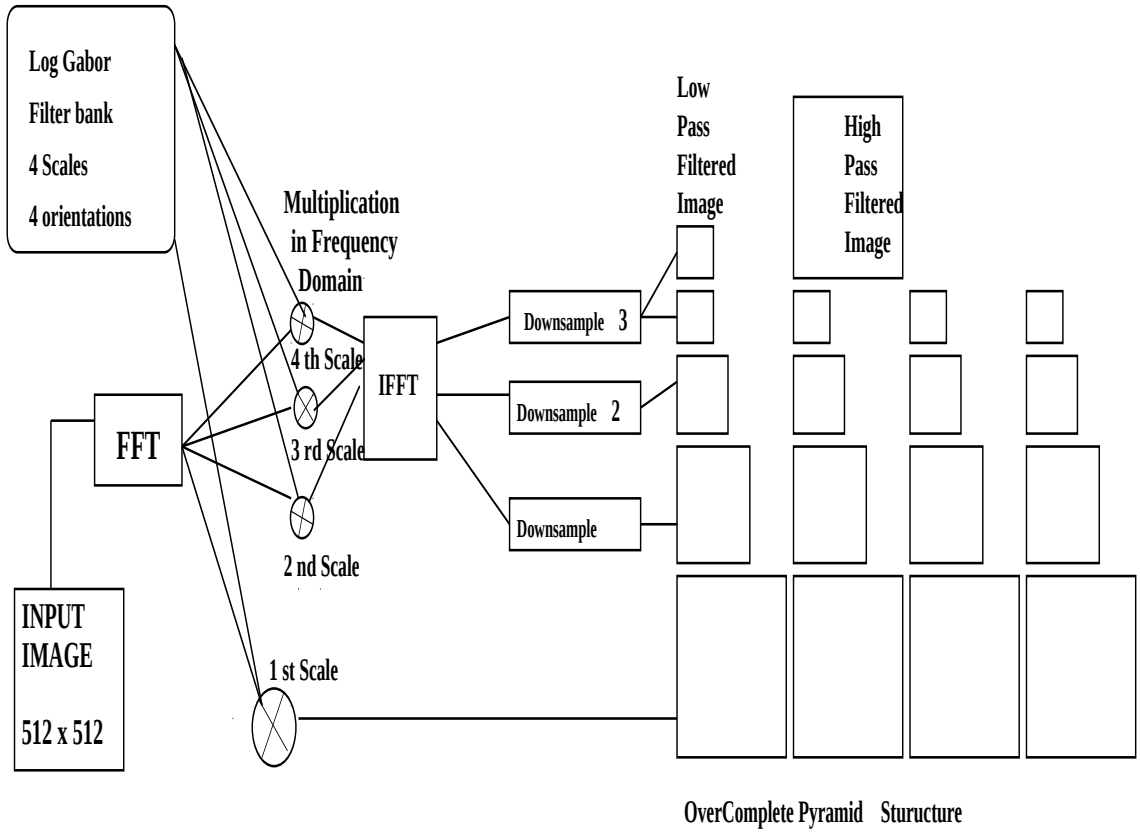

Figure 6.7: Filtering operation using Gabor filter set

3) The first scale filtered images are converted to frequency domain and filtered back with the same first scale filters.

4) The second scale filtered images are upsampled using bilinear interpolation method and then converted to the frequency domain by FFT and they are filtered by the second scale log Gabor filters.

5) Similarly third scale filtered images are upsampled twice and fourth scale filtered images are upsampled thrice and the same operations as carried on step 2 are repeated.

6) The summation of all the above filter outputs are taken and then by IFFT, the image in spatial domain can be reconstructed.

Figure 6.8: Input image for filtering

6.3 Inhibition based sparse coding algorithm

The aim of the inhibition algorithm is to remove the redundancy introduced by the overcomplete log Gabor filtered representation [28][29]. Though the method adopted to compress the data varies from the previous algorithm, it also follows the sparse structure since the distribution of coefficients after compression has sparse distribution [1]. This algorithm aims to keep the expansion of the transform domain to minimum such as to extract a sparse structure out of it. The expansion is kept to minimum because sparse coding algorithm do not permit to recover all the redundancy and also even if the most of the coefficients are zeroed out, still a large population of zeros are needed to be coded and transmitted and this affects the compression rate achieved. Since the transform domain is an overcomplete representation, many sets of the pyramids can provide same reconstruction by the inverse method above mentioned [1]. So the overcompleteness is equivalent to the transforms non-uniqueness. So different iterative algorithms have been proposed in literature [17][25]. But

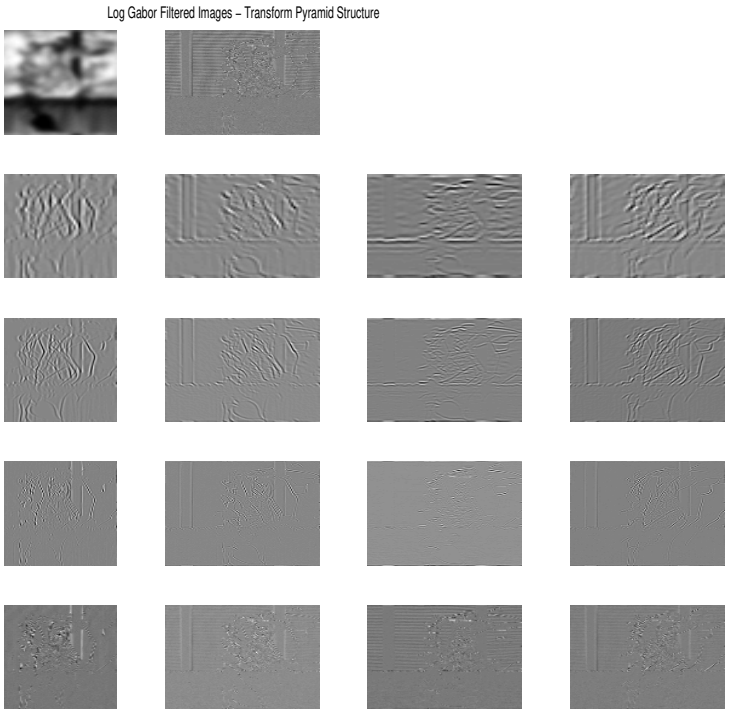

Figure 6.9: Filtered images pyramid structure using log Gabor filters

they have the disadvantage of high complexity and implementation is not practically feasible and also they have the problem of finding a sparse set representation. So these algorithms are seldom applicable for image processing or compression applications.

Here a set of 16 filters varying in 4 different orientations and 4 different scales are considered and also a low pass filter to encode luminance component and a high pass filter to encode very high frequencies. The high pass filter can be avoided for the blurred retinal images. So transform domain for a natural image consists of set of 18 filtered images. Each of this is termed as channel. In this kind of representation the redundancy of data representation is from the neighboring transform coefficients, since the log Gabor functions

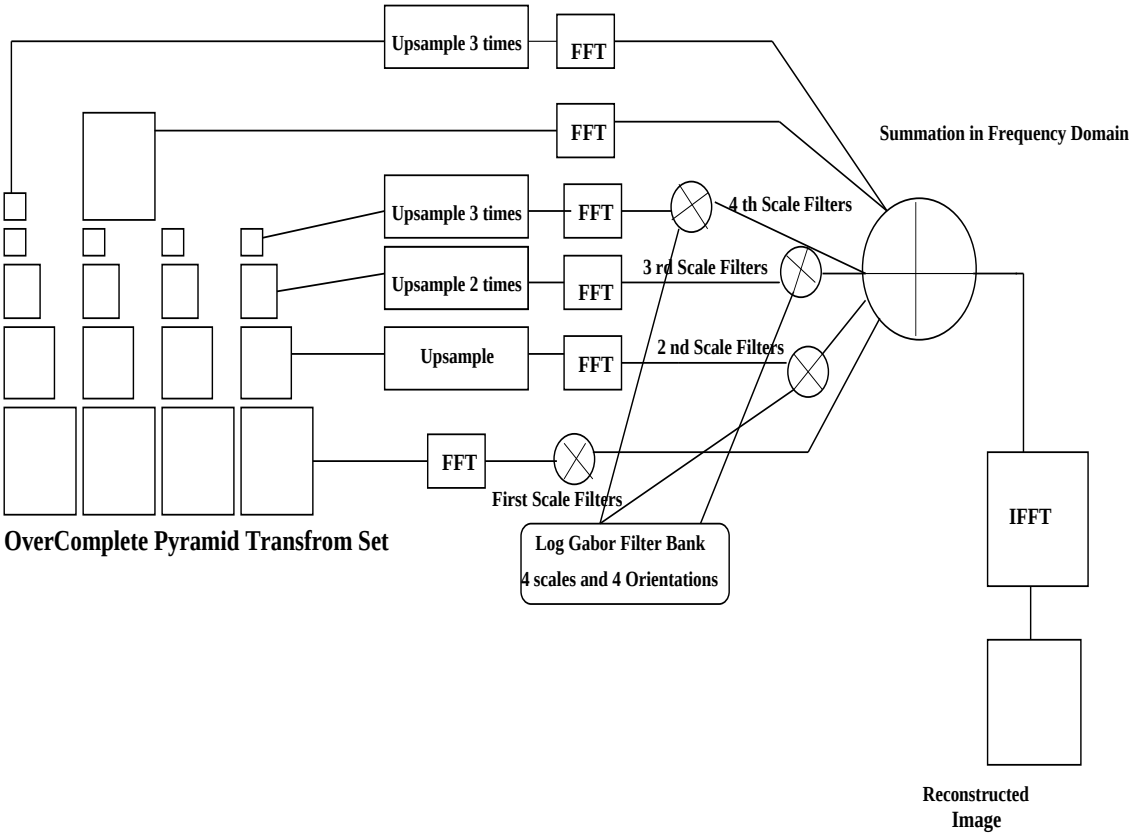

Figure 6.10: Reconstruction of image from filtered pyramid

considered are non orthogonal [28] and so they respond redundantly to the same image feature. So sparse decoding is achieved by removing this redundancy by iterating the the transform to select only few coefficients and zero out the redundant coefficients such that they can reconstruct the original image back [27]. The coefficients obtained have sparse distribution. This is achieved by the iterative nature of the inhibition algorithm. Consider an example image x is encoded to the pyramid g by the linear Gabor transform filtering. This contains large number of coefficients spread out in frequency neighborhood. By applying the iterative algorithm, the same pyramid is reduced to pyramid h which has a sparse distribution of coefficients yet able to represent the image completely as the pyramid g. Both pyramids

g and h can be used for reconstructing the image expect for the fact that the pyramid h is much sparser representation of the information. So the goal of the inhibition algorithm is to obtain the h pyramid from the given g pyramid [27].

6.3.1 Algorithm explanation

To obtain the sparse representation the algorithm implements the competition based method [27] between the neighboring coefficients by zeroing out or inhibiting the lowest coefficients and by concentrating the amplitudes in the higher ones [27]-[29]. The aim as in sparse coding is to obtain as much zeros as possible among the transform coefficients. The algorithm is based on building the pyramid h_n at each iteration with few nonzero coefficients. Each iteration has the following steps:

1) Linear Gabor transform:

The linear transform is computed as already explained by filtering the given image with set of log Gabor filters to form the input pyramid.

$$g = Ax \tag{6.9}$$

where $x \in R^I$ is the source image consisting of I pixels.

$g \in R^J$ is the pyramid representation of the given image as a vector of J real values.

$A \in R^{I \times J}$ is the direct transform.

$\mathring{A}$ is the generalized inverse transform.

$$\mathring{A} = R^{J \times I} \tag{6.10}$$

Since it is overcomplete ($J > I$) representation, $A\mathring{A}$ is not identity.

First iteration is initialized with $h_1 = g$.

2) Selection of coefficients: The coefficients are selected on the basis of: i) They must be local maxima along the normal to the filter direction. ii) They must pass a threshold value ϕ. The value of ϕ is chosen as the maximum absolute value of $g(\phi = max|g|)$. The growing of coefficients is implemented by a sequence of pyramids termed as E_n where $n \in N$. It is increased in each iteration by adding the past value of h_{n-1}. This can be given by:

$$E_n = E_{n-1} + \eta_{n-1} h_{n-1} \tag{6.11}$$

The value of η_{n-1} can be held as constant as 0.02 or can be varied. This helps in faster convergence of the algorithm. h_n coefficients are selected when their corresponding E_n passes the threshold value ϕ in absolute value. Through out this process initially larger coefficients are first selected and then smaller coefficients are added in each step to it.

3) An approximation pyramid a_n is obtained by zeroing out all the non selected coefficients of h_{n-1}. These non selected coefficients are called *residual pyramid* r_n.

$$a_n(k) = \begin{cases} h_{n-1}(k) & \text{if } |E_n(k)| > \phi \\ 0 & \text{otherwise} \end{cases}$$

$$h_{n-1} = a_n + r_n \tag{6.12}$$

4) Coefficient adjustment and iteration:

A new pyramid h_n is defined as for $(n > 1)$

$$h_n = a_n + A\mathring{A}r_n \tag{6.13}$$

$$h_n = h_{n-1} - r_n + A\mathring{A}r_n \tag{6.14}$$

It is straightforward to note that after each iteration h_n has exact reconstruction given by $\mathring{A}h_n = x$

This algorithm is repeated through step 2 for given number of iterations or till it reaches the desired sparse distribution of coefficients.

6.3.2 Example for algorithm iteration

The algorithm is explained in detail for a single iteration. Consider this as the first iteration (n=1).

1) Linear transform: For the given image the linear transform is obtained and the pyramid structure is obtained as explained in previous section. $h_1 = g$

2) $E_1 = 0$. So further proceeding to the algorithm is not possible for n=1. Consider the iteration number n=2.

1) Linear transform: $h_1 = g$

2) Coefficient selection: $E_2 = E_1 + \eta_1 h_1$ the value of η is kept constant 0.02 for all iterations.

$$a_2(k) = \begin{cases} h_1(k) & \text{if } |E_1(k)| > \phi \\ 0 & \text{otherwise} \end{cases}$$

3) Residual pyramid:

$$r_2 = h_1 - a2$$

4) New pyramid construction:

$$h_2 = a_2 - A\mathring{A}r_2$$

This step of calculating $A\mathring{A}r_2$ is by first computing the inverse transform of the error image and then calculating the linear Gabor transform of it. The algorithm can be checked for its operation by reconstructing the image with this newly formed h_2., since at every iteration the pyramid constructed must have exact reconstruction [27].

6.4 Application towards image compression

As the number of iterations of this algorithm is increased, the energy is concentrated in minimum number of coefficients many of the non selected coefficients are truncated to zero in each iterations. Matching Pursuit algorithm [26] can select only one coefficient per iteration, whereas this algorithm chooses few coefficients per iteration. This makes it computationally fast compared to similar algorithms although it requires still significant computation time for efficient practical applications. The coefficients are then quantized based on their distribution by using nonuniform quantization methods and they are encoded using runlength and Huffman as encoder as explained in JPEG section. The compression rates and iteration numbers required are tabulated in chapter 7.

CHAPTER 7: RESULTS AND CONCLUSIONS

7.1 JPEG and JPEG 2000 simulation results

The following figure compares the performance of the quantization algorithms. From Fig.7.1 it can be inferred that by choosing the coefficients based on their magnitudes, provides better performance. So for quantizing blurred and whitened images, this algorithm is chosen.

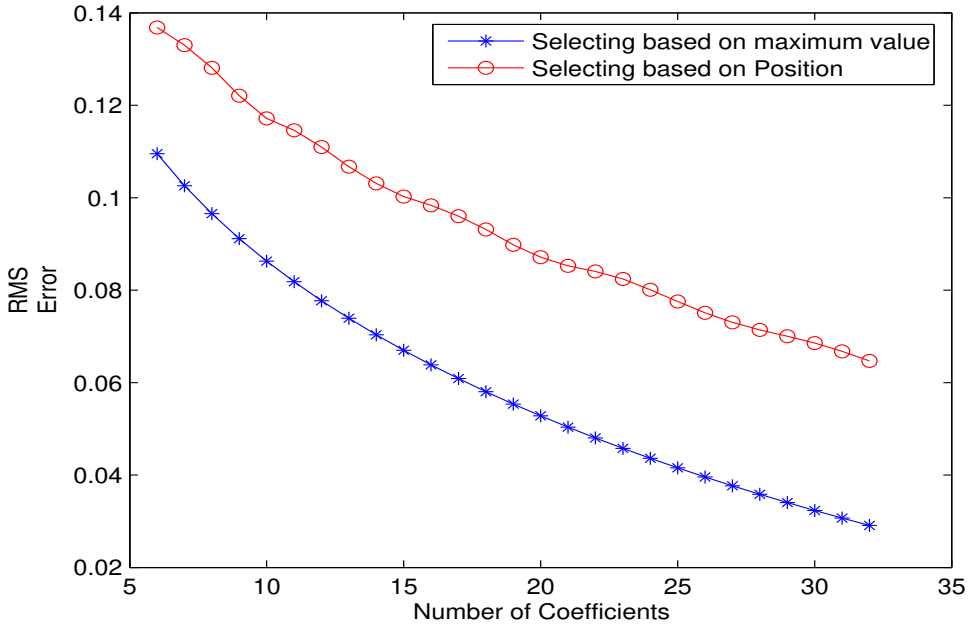

Figure 7.1: Selection of coefficients algorithm comparison

Here N is natural image and B is the blurred, whitened and normalized image.

Table 7.1: BPP Vs RMS using JPEG

BPP	**RMS(N)**	**RMS(B)**
0.57	0.2881	0.3440
1.00	0.1822	0.2912
1.55	0.1044	0.1132
2.00	0.066	0.0882
2.57	0.0558	0.0773

Table 7.2: BPP Vs RMS using JPEG2000

BPP	**RMS(N)**	**RMS(B)**
0.57	0.0898	0.1203
1.00	0.0644	0.0811
1.55	0.0202	0.0218
2.00	0.01338	0.0142
2.57	0.00884	0.009877

7.2 Spare coding results

Fig.7.2 illustrates the improvement in RMS performance by using the non uniform quantization. The following section discusses the result of training the SCL algorithm with different kind of image sets such as natural images (N), blurred images (B), whitened and normalized images (WN) and blurred whitened normalized images (BWN). The improvement in performance for the case of whitened and normalized image and also for blurred, whitened and normalized images from the natural images due to the filtering of high frequency components.

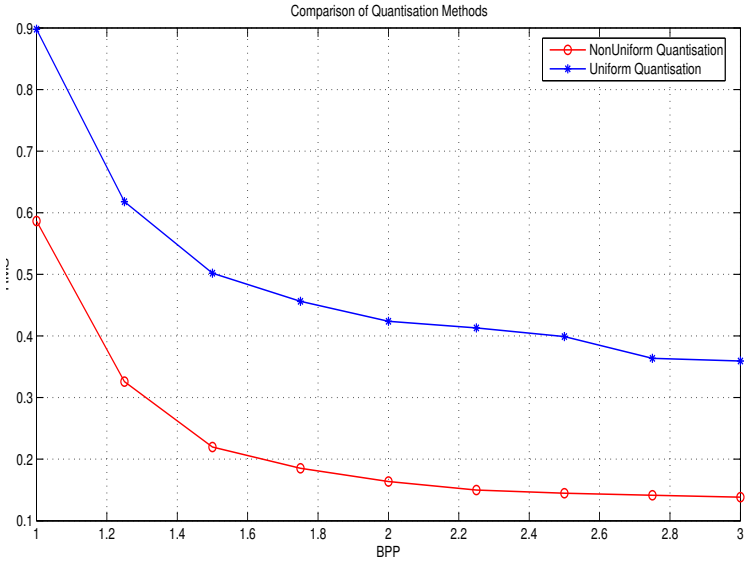

Figure 7.2: Comparison of uniform and non uniform quantization

Table 7.3: BPP Vs RMS using SCL for different imagesets

BPP	RMS(N)	RMS(B)	RMS(WN)	RMS(BWN)
1.00	0.931	0.7822	0.3242	0.4100
1.54	0.522	0.433	0.1011	0.0824
2.00	0.2684	0.2256	0.0727	0.0766
2.57	0.2098	0.1699	0.0629	0.0611
3.00	0.1837	0.1249	0.0614	0.0577

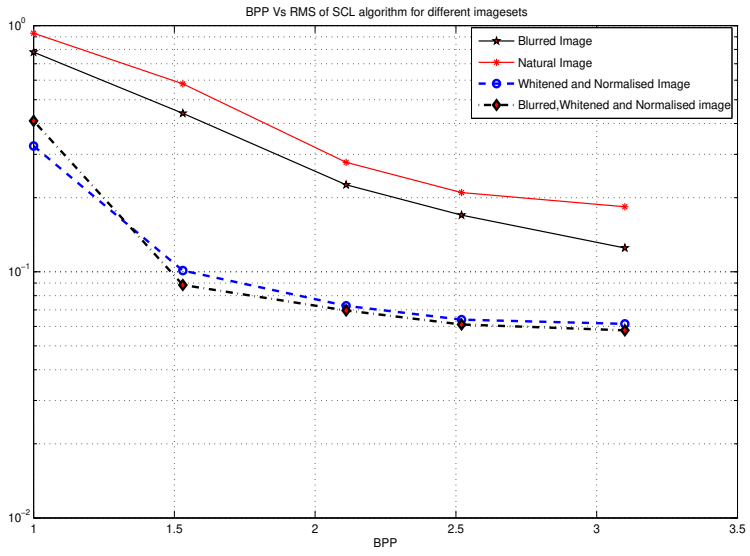

Figure 7.3: Comparison of SCL coding performance for various imagesets

7.3 Inhibition based sparse coding results

Table.7.4 and table.7.5 illustrates the compression results for natural and blurred, whitened and normalized images correspondingly. It can be inferred that there is not much

difference in compression rates but the iteration number required for the blurred images are lesser. The reduction in iteration number is due to the fact that for blurred images, we neglect the high and low pass filters in the pyramids.

Table 7.4: Inhibition results for natural images

BPP	**Iterations**	**RMS)**
0.57	344	0.08820
1.00	287	0.0781
1.55	221	0.0586
2.00	133	0.04332
2.57	102	0.0352

Table 7.5: Inhibition results for blurred and whitened images

BPP	**Iterations**	**RMS)**
0.57	323	0.1022
1.00	231	0.0612
1.55	187	0.0433
2.00	98	0.0388
2.57	68	0.0211

Table 7.6: Comparison graphs for sparse and JPEG codings for natural images

BPP	**JPEG**	**SCL**	**Inhibition**	**JPEG2000**
1.00	0.1822	0.931	0.0781	0.0644
1.55	0.1044	0.522	0.0586	0.0202
2.00	0.066	0.2684	0.04332	0.01338
2.57	0.0558	0.2098	0.0358	0.00844

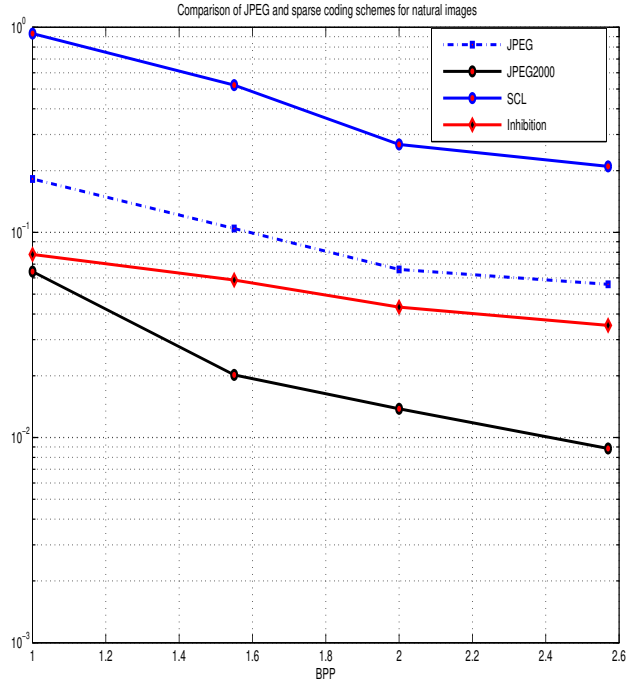

Figure 7.4: Comparison of sparse and JPEG codings for natural images

Table 7.7: Comparison graphs for sparse and JPEG codings for blurred whitened images

BPP	JPEG	SCL	Inhibition	JPEG2000
1.00	0.2912	0.4100	0.1022	0.0811
1.55	0.113	0.0884	0.0612	0.0218
2.00	0.082	0.0716	0.0433	0.0142
2.57	0.077	0.0611	0.0388	0.009877
3.1	0.06111	0.0577	0.0211	0.008433

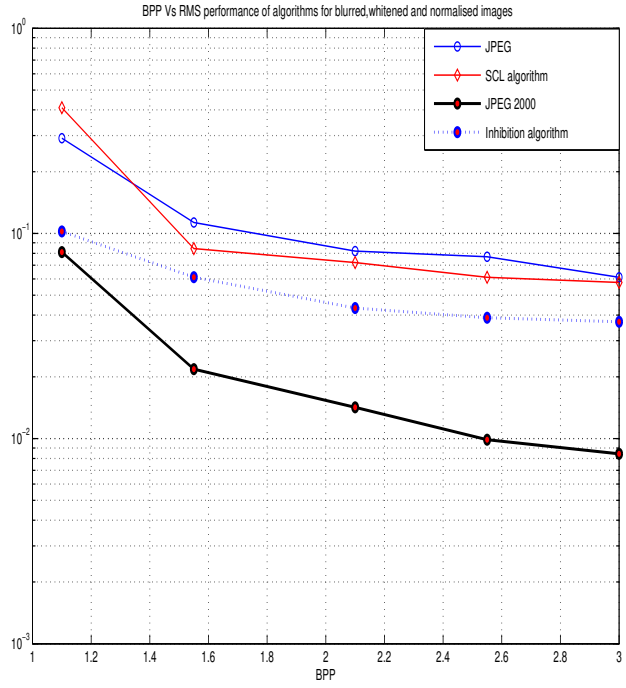

Figure 7.5: Comparison of sparse and JPEG codings for blurred, whitened images

7.4 Conclusions and future works

The sparse coding algorithm discussed produces reasonable compression results when compared with the existing JPEG standards. Our main interest in this thesis is for blurred and whitened images. For these images we can see the SCL coding algorithm slightly betters the JPEG but still cannot better the JPEG 2000 algorithm. The inhibition based algorithm results are very near to JPEG 2000 results. Here the testings are done only for fixed number of natural images. So further testings with broad variety of images are needed.

Significant improvement in performance can be made in the SCL coding algorithm by considering more efficient quantizers such as trellis based quantizers and also efficient source coding strategies can be employed. Most of the RMS error in this algorithm is from the quantization stage. The inhibition algorithm is still computationally very hard to simulate. So investigation towards reduction in complexity of this algorithm is necessary. Although, we cannot prove that sparse coding is the optimal way of encoding the blurred retinal images, this work can be considered as a good beginning for this kind of research rather than final solution for the problem.

7.4.1 Future work

The preceding section contains some of the short comings or existing problems with the algorithms. First progress would be towards correcting on the existing problems. Recent developments in the sparse coding algorithm suggests that soft sparse models considered so far could be replaced by the hard sparse models very recently proposed. Significant advancement would be to adapt these sparse coding algorithms for space time functions to

encode the real world videos rather than stationary images. Recent researches are directed towards proving that these kind of sparse coding schemes may be at work at the human visual system for viewing real world sceneries.

APPENDIX A: HUFFMAN AC CODES

Table A.1: AC category codes

RUN/SIZE	**CODE LENGTH**	**CODE WORD**
0/0(EOB)	2	00
0/1	2	01
0/2	3	100
0/3	4	1010
0/4	5	11000
0/5	5	11001
0/6	6	111000
0/7	7	1111000
0/8	9	111110100
0/9	10	1111110110
0/A	12	111111110100
1/1	4	1011
1/2	6	111001
1/3	8	11110110
1/4	9	111110101
1/5	11	11111110110
1/6	12	111111110101
1/7	16	1111111110001000
1/8	16	1111111110001001
1/9	16	1111111110001010
1/A	16	1111111110001011

2/1	5	11010
2/2	8	11110111
2/3	10	1111110111
2/4	12	111111110110
2/5	15	111111111000010
2/6	16	1111111110001100
2/7	16	1111111110001101
2/8	16	1111111110001110
2/9	16	1111111110001111
2/A	16	1111111110010000
3/1	5	11011
3/2	8	11111000
3/3	10	1111111000
3/4	12	111111110111
3/5	16	1111111110010001
3/6	16	1111111110010010
3/7	16	1111111110010011
3/8	16	1111111110010100
3/9	16	1111111110010101
3/A	16	1111111110010110

4/1	6	111010
4/2	9	111110110
4/3	16	1111111110010111
4/4	16	1111111110011000
4/5	16	1111111110011001
4/6	16	1111111110011010
4/7	16	1111111110011011
4/8	16	1111111110011100
4/9	16	1111111110011101
4/A	16	1111111110011110
5/1	6	111011
5/2	10	1111111001
5/3	16	1111111110011111
5/4	16	1111111110100000
5/5	16	1111111110100001
5/6	16	1111111110100010
5/7	16	1111111110100011
5/8	16	1111111110100100
5/9	16	1111111110100101
5/A	16	1111111110100110

6/1	7	1111001
6/2	11	11111110111
6/3	16	1111111110100111
6/4	16	1111111110101000
6/5	16	1111111110101001
6/6	16	1111111110101010
6/7	16	1111111110101011
6/8	16	1111111110101100
6/9	16	1111111110101101
6/A	16	1111111110101110
7/1	7	1111010
7/2	11	11111111000
7/3	16	1111111110101111
7/4	16	1111111110110000
7/5	16	1111111110110001
7/6	16	1111111110110010
7/7	16	1111111110110011
7/8	16	1111111110110100
7/9	16	1111111110110101
7/A	16	1111111110110110

8/1	8	11111001
8/2	16	1111111110110111
8/3	16	1111111110111000
8/4	16	1111111110111001
8/5	16	1111111110111010
8/6	16	1111111110111011
8/7	16	1111111110111100
8/8	16	1111111110111101
8/9	16	1111111110111110
8/A	16	1111111110111111
9/1	9	111110111
9/2	16	1111111111000000
9/3	16	1111111111000001
9/4	16	1111111111000010
9/5	16	1111111111000011
9/6	16	1111111111000100
9/7	16	1111111111000101
9/8	16	1111111111000110
9/9	16	1111111111000111
9/A	16	1111111111001000

A/1	9	111111000
A/2	16	1111111111001001
A/3	16	1111111111001010
A/4	16	1111111111001011
A/5	16	1111111111001100
A/6	16	1111111111001101
A/7	16	1111111111001110
A/8	16	1111111111001111
A/9	16	1111111111010000
A/A	16	1111111111010001
B/1	9	111111001
B/2	16	1111111111010010
B/3	16	1111111111010011
B/4	16	1111111111010100
B/5	16	1111111111010101
B/6	16	1111111111010110
B/7	16	1111111111010111
B/8	16	1111111111011000
B/9	16	1111111111011001
B/A	16	1111111111011010

C/1	9	111111010
C/2	16	1111111111011011
C/3	16	1111111111011100
C/4	16	1111111111011101
C/5	16	1111111111011110
C/6	16	1111111111011111
C/7	16	1111111111100000
C/8	16	1111111111100001
C/9	16	1111111111100010
C/A	16	1111111111100011
D/1	11	11111111001
D/2	16	1111111111100100
D/3	16	1111111111100101
D/4	16	1111111111100110
D/5	16	1111111111100111
D/6	16	1111111111101000
D/7	16	1111111111101001
D/8	16	1111111111101010
D/9	16	1111111111101011
D/A	16	1111111111101100

E/1	14	11111111100000
E/2	16	1111111111101101
E/3	16	1111111111101110
E/4	16	1111111111101111
E/5	16	1111111111110000
E/6	16	1111111111110001
E/7	16	1111111111110010
E/8	16	1111111111110011
E/9	16	1111111111110100
E/A	16	1111111111110101
F/0 (ZRL)	10	1111111010
F/1	15	111111111000011
F/2	16	1111111111110110
F/3	16	1111111111110111
F/4	16	1111111111111000
F/5	16	1111111111111001
F/6	16	1111111111111010
F/7	16	1111111111111011
F/8	16	1111111111111100
F/9	16	1111111111111101
F/A	16	1111111111111110

APPENDIX B: HUFFMAN DC CODES

Table B.1: DC codes

RANGE	DC difference category	AC difference category
0	0	NA
-1,1	1	1
-3,-2,2,3	2	2
-7,,-4,47	3	3
-15,-8,8,.15	4	4
-31,-16,16,.31	5	5
-63,-32,32,.63	6	6
-127,-64,64,.127	7	7
-255,-128,.128,255	8	8
-511,.,-256,256,.511	9	9
-1023,.,-512,512,.1023	A	A
-2047,-1024,1024,.. 2047	B	B
-4095,-2048,2048,.. 4095	C	C
-8191,-4096,4096,..8191	D	D
-16383..,-8192,8192,.16383	E	E
-32767,16384,16384,32767	F	NA

Table B.2: DC codes (luminance)

Category	**Base Code**
0	010
1	011
2	100
3	00
4	101
5	110
6	1110
7	11110
8	111110
9	1111110
A	11111110
B	111111110

LIST OF REFERENCES

[1] D. J. Field, "Relations between the statistics of natural image and the response properties of cortical cells, " *Journal of Optical Society*, Vol. 4, No. 12, Aug., 1987.

[2] H. B. Barlow, "Understanding natural vision, " *Physical and biological processing of images*, *Springer-Verlag, Berlin* Vol. 11, 1987, pp. 2-14

[3] F. Attneave, "Some informational aspects of visual perception, " *Psychological Review*, Vol. 61, No. 3, 1954.

[4] J. J. Atick, "Could information theory provide an ecological theory of sensory processing?, " *Network:Comput. Neural Syst. 3*, Institute of Physics, Bristol, 1992, Vol. 3, pp. 213-251.

[5] J. J. Gibson, *The perception of the visual world*, Houghton Mifflin, Boston, 1966.

[6] K. K. De Valois, R. L. DeValois and E. W. Yund, "Response of striate cortical cells to grating and checkerboard patterns," *Journal of Physiol*, Vol. 291, pp. 483-505, 1980.

[7] D. G. Albrecht, R. L. DeValois, and L. G. Thorell, "Visual cortical neurons:are bars or gratings the optimal stimuli, " *Science*, Vol. 207, pp. 88-90, 1981.

[8] D. Kersten, "Predictability and redundancy of natural images, " *Journal of Optical Society*, Vol. 4, pp. 2395-2400, 1987.

[9] D. Kersten, "The problem of sparse image coding, " *Journal of Mathematical Imaging and Vision*, Vol. 17, pp. 89-108, 2002.

[10] E. I. George and R. E. McCulloch, "Approaches for Bayesian variable selection, " *Statistica Sinica*, Vol. 7, pp. 339-373, 1997.

[11] G. Golub and C. van Loan, *Matrix Computations*, John Hopkins University press, Baltimore, 1996.

[12] S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes on C,* second edition, Cambridge University press, Cambridge, 1992.

[13] I. Daubechies, "The wavelet transform time-frequency localization and frequency analysis, " *IEEE Trans on Information theory*, Vol. 36, pp. 961-1005, 1990.

[14] A. J. Bell and T. J. Sejnowski, "The independent components of natural scenes are edge filters, " *Vision research*, Vol. 37, pp. 3327-3338, 1997.

[15] A. Hyvarien, J. Karhunen, and E. Oja, *Independent component analysis*, Wiley, Newyork, 2001.

[16] S. Roweis and Z. Ghahramani, "A unifying review of linear gaussian models, " *Neural Computation*, Vol. 11, No. 2, pp. 305-345, 1999.

[17] A. J. Bell and T. J. Sejnowski, "An information-maximazation approach to blind seperation and blind deconvolution, " *Neural Computation*, Vol. 7, pp. 1129-1159, 1995.

[18] P. A. D. F. R. Hojen-Sorensen, O. Winther, and L. K. Hansen, "Mean filed approaches to independent component analysis, " *Neural Computation*, Vol. 14, pp. 889-918, 2002.

[19] A. Hyvarinen and M. Inki, "Estimating cvercomplete independent component bases for image windows, " *Neural Computation*, Vol. 17, No. 2, pp. 139-152, Aug. 14, 1987.

[20] M. Lewicki and B. Olshausen, "A probablistic framework for the adaptation and comparision of image codes, " *Journal of Optical Society*, Vol. 16, No. 7, pp. 1587-1601, 1998.

[21] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplet basis set:A startergy employed by V1, " *Vision Res*, Vol. 37, No. 7, pp. 3311-3325, 1997.

[22] B. A. Olshausen and K. J. Millman, "Learning sparse codes with a mixture of Gaussian priori, " *Advances in Neural Information Processing Systems*, MIT press, Cambridge, pp. 841-847, 2000.

[23] B. A. Olshausen, P. Sallee, and M. S. Lewicki, "Learning sparse codes using wavelet pyramid architecture, " *Advances in Neural Information Processing Systems*, Cambridge MA, pp. 887-893, 2000.

[24] S. Chen and D. L. Donoho, "Examples of basis pursuits, " *Proc of the SPIE*, Vol. 2569, No. 2, pp. 564-574, 1995.

[25] G. Davis, S. Mallat, and M. Avellaneda, "Greedy adaptive approximation, " *J. constructive Approximation*, Vol. 13, No. 7, pp. 57-98, 1997.

[26] S. Mallat and Z. Zhang, "Matching pursuit with time frequency dictionaries, " *IEEE Trsnsactuions Signal Processing*, Vol. 41, No. 12, pp. 3397-3415, 1993.

[27] S. Fisacher and G. Cristobal, "Minimum entrpy transform using Gabor wavelets for image compression, " *CSIC Imaging and Vision*, Serrano 121.28006, Madrid, Spain.

[28] S. Fisacher, G. Cristobal, and R. Redondo, "Sparse edge coding using overcomplete using Gabor wavelets, " *CSIC Imaging and Vision*, Serrano 121.28006, Madrid, Spain.

[29] S. Fisacher, G. Cristobal, and R. Redondo, "Sparse overcomplete Gabor wavelet representation based on local competitions , " *IEEE Transactions on Image Processing*, Vol. 15, No. 2, Feb 2006.

[30] P. E. Hallet, "Some Limitation to human peripheral vision, " *In limits of Vision*, CRC Press Inc., Boston, USA, pp. 44-80, 1991.

[31] F. W. Campbell and R. W. Gubisch, "Optical quality of human eye, " *J. Physiol*, Vol. 186, London, pp. 558-578, 1966.

[32] W. S. Geisler, "Physical limits of Acuity and Hyper-Acuity, " *J. Opt. Soc. Am.* , Vol. 1:2, pp. 775-782, 1984.

[33] W. S. Geisler, "Ideal discrimination in Spatial Vision:two point stimuli, " *J. Opt. Soc. Am.* , Vol. 2:9, London, pp. 1483-1497, 1985.

[34] D. Gabor, "Theory of communication, " *J. Inst. Elec. Engg.* , Sec. 93, pp. 429-457, 1946.

[35] J. H. Van Deemter, "Signal reconstruction with a small set of Gabor filters, " *Institute de Optica(CSIC)*, Serrano 121, 28006, Madrid, Spain.

[36] J. Daugman, "Two dimensional spectral analysis of cortical receptive field profiles, " *Vision Res*, Vol. 20, pp. 847-856, 1980.

[37] J. Daugman, "Uncertainity relation for resolution in space, spatial frequency and orientation optimized by 2-D visual cortical fields., " *J, Opt. Soc. Am. A*, Vol. 2(7), pp. 1160-1169, 1985.

[38] J. Daugman, "Complete discrete 2-D Gabor transform for image analysis and compression, " *IEEE Trans. Acouust. Speech. Signal Proc*, Vol. 36(7), pp. 1169-1179, 1988.

[39] J. R. Movellan, "Tutorial on Gabor Filters, " *Open source documentation*, 1996, 2002.

[40] "Log Gabor filters, " *http://csse. uwa. edu. au*, Obtained and used with permisssion open source documentation

[41] G. K. Wallace, "The JPEG picture compression standard, " *IEEE Transaction on Consumer Elecronics*, Multimedia Engineering, Digital Equipment Corporation, Maynard, Massachussetts. Vol. 38, No. 1, Feb. 1992.

[42] "Digital Compression and Coding of Continous tone Still Images, Part 1 requirements and guidelines, " *ISO/IEC JTC1 Draft International Standard* 10918-1, Nov. 1991.

[43] "Digital Compression and Coding of Continous tone Still Images, Part 2 requirements and guidelines, " *ISO/IEC JTC1 Draft International Standard* 10918-1, Dec. 1991.

[44] D. A. Huffman, "A method for the construction of minimum redundancy codes, " *In Proceedings IRE*, Vol 40, 1962, pp. 1098-1101.

[45] O. K. Al-Shaykh, I. Moccagatta, and H. Chen, "Rockwell Science Center, " *IEEE Proceedings of Image Processing*,

[46] "JPEG2000 Image coding scheme, " *ISO/IEC*, 15444-1, 2000.

[47] "JPEG 2000: Image coding system, " *Technical Report N390*, ISO/IEC JTC1/SC29/WG1, June 1996.

[48] "JPEG 2000: Image coding system, " *Technical Report N505*, ISO/IEC JTC1/SC29/WG1, March 1997.

[49] "Lossless and near-loseless compression of continous tone still images, " *ISO/IEC and ITU-T Recommendation T. 87* 114495-1, Dec. 1999.

[50] J. M. Shapiro, "Embedded image coding using zero trees of wavelet coefficients, " *IEEE Trans on Signal Processing*, Vol. 41, pp. 3445-3462, Dec. 1993.

[51] C. E. Shannon, "A Mathematical Theory Of Communication," *Bell System Technical Journal*, Vol. 27, pp.379-423 (Part I), 623-656(Part II), Jul. 1948.

[52] R. G. Gallager, *Low Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.

[53] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58-60, Feb 2001.

[54] S. D. Taubman and W. M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publisher, 2002.

[55] C. R. Gonzalez and E. R. Woods, *Digital Image Processing*, Pearson Education, 2002.

[56] C. R. Gonzalez and E. R. Woods, *Digital Image Processing using Matlab*, Pearson Education, 2002.

[57] K. S. Shanmugan and A. M. Breipohl, *Random Signals Detection, Estimation and Data Analysis*, John Wiley and Sons, 1988.

[58] P. Athanasios and P. S. Unnikrishnan, *Probablity, Random variables and Stochastic Processes*, McGraw Hill, Fourth Edition, 2002.