1996

Detectability model for the evaluation of lossy compression methods on radiographic images

Vivek Ramaswami
Iowa State University

Detectability model for the evaluation of lossy compression methods on radiographic images

by

Vivek Ramaswami

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Major Professors: Satish S. Udpa and Joseph N. Gray

Iowa State University

Ames, Iowa

1996

Graduate College
Iowa State University

This is to certify that the Master's thesis of

Vivek Ramaswami

has met the thesis requirements of Iowa State University

Signature redacted for privacy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. Joseph N. Gray and Dr. Satish S. Udpa for their guidance and support in overseeing the work presented here. I would like to thank Dr. Lalitha Udpa and Richard Wallingford for serving as members on my committee.

1 INTRODUCTION

The purpose of image data compression is to represent data efficiently without loss of information. This involves identification and removal of unnecessary information. Uncompressed image data is typically represented in such a way so that it is highly redundant. Need for data reduction arises due to limitation on storage space or transmission time. Although the storage capacities of magnetic media increases, the demand for data compression has been growing steadily. The Nuclear Regulatory Commission requires that the radiographs be stored for 100 years. The film radiograph degrades due to aging. To avoid this generally the radiograph is digitized between 35 and 100 micron spatial resolution and 12 bits. For a 11x14 inch radiograph this requires on the order of 30 Mbytes for storage. Data compression is necessary to increase the number of images that can be stored. Various factors used in the evaluation of compression are the amount of compression provided, speed of compression and decompression, memory requirements and the mean square error (MSE). Since the radiographs are viewed by the human eye, it is very important that the compression does not introduce any artifacts that are visible. It is necessary to evaluate the visual impact of the error due to compression. In this thesis, a method is presented which calculates the visual distortion of the compressed image as compared to the original image. This method is based on a model of the human eye.

The compression techniques implemented in this thesis are lossy compression methods. Chapter 2 is concerned with the discussion of the various lossy compression algorithms including vector quantization, fractal image compression and compression using

Discrete Cosine Transform (DCT). Chapter 3 discusses the idea of how the mathematical differences between the original and the compressed image are converted to visual differences and it gives a method of estimating the visible artifacts introduced by the compression method. Chapter 4 provides a description of the detectability measure used and how to calculate intensity of the pixel from the cathode ray tube characteristics. Chapter 5 summarizes the work done and possible directions for future work in the areas discussed in the previous chapters.

In lossless compression there is no data loss and the reproduction of data is exact. This is useful to compress computer programs and data which requires perfect reconstruction. Current lossless compression techniques offer only modest amount of compression (2 to 4). Since the image is viewed by the human eye, some degree of loss can be tolerated. The advantage of lossy representations is that they generally achieve compression ratios many times greater than lossless methods.

Vector Quantization (VQ) is one of the lossy compression methods that is gaining widespread use. VQ offers high compression ratios (8 to 10) while still maintaining acceptable subjective image quality. One salient feature of VQ is that the decoder is very easy to implement, making it attractive for applications where the data is compressed once and decompressed many times, as in archiving. One of the disadvantages is the large amount of memory required for the codebook. VQ is not suitable for compressing a few images and broadcasting the information since the codebook also has to be distributed. VQ provides the best possible performance of any block coding technique for a given block size [7]. VQ is often limited to small block sizes since the computational complexities are greater and artifacts are far more visible when larger blocks are employed.

Arithmetic coding is used to further compress the indices of VQ and other files like mean and differential pulse code modulation (dpcm) values in a lossless manner. Arithmetic coding removes redundancies caused by non-uniform distribution of message

symbols. The basic idea behind this lossless compression scheme is to assign a variable number of bits to symbols depending on their probability of occurrence. Symbols that occur frequently are assigned a fewer number of bits while symbols that are less probable are assigned more bits. This has the overall effect of reducing the average data rate.

Arithmetic coding has several advantages over other lossless coding methods like Huffman coding, since it is capable of providing near optimal performance and does not require blocking of the data stream [7]. Huffman coding requires that each symbol be mapped to an integral number of bits. Arithmetic coding does not require the use of a blocking scheme and allows fractional bit allocations and consequently each bit in the arithmetic code is fully utilized. Huffman coding requires that a new set of codes be generated every time the symbol probabilities change. This must be done both at the encoder and the decoder and is time consuming. Arithmetic coding enables the separation of the process of coding and source modeling. Source modeling attempts to model data in a statistical sense, the way in which an information source produces data.

The digitized images used in thesis are drawn from several sources:

- Digitized film radiographs
- Real time images obtained using phosphor screen and CCD camera
- Images obtained using the simulation tool XRSIM

XRSIM is a simulation package used to obtain a radiograph of a part whose computer aided design (CAD) model is input to the package. XRSIM has options for simulating different types of film. It is capable of calculating the optimum exposure times for a given set of generator voltage and current values. It also allows the introduction of noise. Various types of flaws like ellipse can be inserted into the CAD model and its effect on the radiograph can be observed. It should be noted that XRSIM can produce images that have more that 8 bits. Most of the images used in this thesis are 8-bit images. The

compression methods can also be extended to images using more gray scales through suitable modifications. Some of the images were digitized from radiographs using a digitizer. Other images were obtained using a real time x-ray imaging system.

In image compression the location of the image error is as important as the magnitude of the error. Small errors around very bright regions are not visible. The mean square error measure does not provide a good measure of the visibility of the error. Subjective image quality as evaluated by human observers is used to classify the distortion in the compressed images as perceptible, imperceptible, annoying and very annoying. This method of comparing compression methods can be used for some classes of images but cannot be used for all cases, as these results are very subjective. Two compressed images might have the same mean square error but in one of the compressed images the error may be less annoying or visible when compared to the other. A new technique capable of quantifying the visual difference between two images from a mathematical perspective is the visual difference predictor (VDP) [13]. It can be used to test whether the flaw or any artifact is visible when compared to a reference image. This is important because most of the radiographs are viewed by human eyes. The predicted visible difference image can be used to optimize parameters such as viewing distance and background illumination so that the regions of interest can be seen by the human eye. The VDP can be used to see how the compressed image differs from the original image and it can be used to select the compression routines that minimize the visible difference between the compressed and original images.

2 IMAGE COMPRESSION METHODS

2.1 Introduction

This chapter discusses the various image compression methods that were used to compress digitized radiographic images. The techniques explored in this chapter are Vector Quantization, Fractal Image Compression and JPEG which is a Discrete Cosine Transform (DCT) based method. Fractal image compression is a new concept which has been explored widely in the past four years. JPEG is a standard compression method which is widely used to compare the performance of other image compression routines. The following sections provide an in depth description of each of the above methods together with the results of their application to radiographic images.

2.2 Vector quantization

Vector Quantization (VQ) methods represent a generalization of scalar schemes to the quantization of an ordered set of real numbers (vector). Scalar quantization is used for analog to digital conversion. In contrast VQ is used when the input signal already has a digital representation. A vector can be used to describe any pattern, such as a segment of speech or a segment of an image. VQ can be viewed as a form of pattern recognition where an input pattern is approximated by one of a predetermined set of patterns. This involves matching the input pattern to a stored set of codewords [7].

2.2.1 Classified vector quantizer

The vector quantization algorithm implemented in this thesis is described in the following sections. The mean is removed from the shade blocks since this yields a substantial reduction in computation and storage by separately coding the vector mean and its mean removed residual vector. Let N_m and N_r be the size of the mean codebook and residual codebook, there are $N_r N_m$ possible reproduction vectors. Since the mean and residual codebooks are searched individually, the computational complexity is reduced. Computation of the mean is simple (scalar computation) when compared to the residual comparisons. For the edge blocks differential pulse code modulation (DPCM) is used where the current pixel is estimated from the previous neighboring pixels. The prediction error is quantized.

The classified vector quantizer (CVQ) classifies blocks of images as either shade or edge blocks depending on the gradient and line thresholds. CVQ reduces the complexity of coding, allowing classes to be coded based on their perceptual significance. The classification of each block is based on a perceptual model of edges. It is assumed that edge perception of the eye is proportional to normalized gradient and not the actual gradient [12]. The gradient between two adjacent pixels is normalized by the average gray level of the two pixels. The normalized gradient between pixel $x(i,j)$ and its neighbor $x(i,j+1)$ is

$$d_h = \frac{2[x(i,j) - x(i,j+1)]}{x(i,j) + x(i,j+1)} \tag{2.1}$$

Similarly the gradient between $x(i,j)$ and $x(i+1,j)$ is

$$d_v = \frac{2[x(i,j) - x(i+1,j)]}{x(i,j) + x(i+1,j)} \tag{2.2}$$

These normalized gradients are compared to two thresholds, T_l and T_e to determine if the pixel transition is a shade or edge. For 8 bit images the edge threshold T_e and the line threshold T_l are parameters that can be varied. In this thesis the edge threshold is

set as follows:

$$\begin{aligned} T_e &= \frac{8.0}{d_{av}} \qquad if\ d_{av} < 30 \\ &= \frac{43.0}{d_{av}} \qquad if\ d_{av} > 215 \\ &= T_e \qquad otherwise \end{aligned} \tag{2.3}$$

where d_{av} is the average intensity of the two pixels under consideration. Four counters are used to determine the number of edge transitions in the horizontal and vertical directions. The counters are:

H_p= Positive horizontal gradient counter

H_n= Negative horizontal gradient counter

V_p= Positive vertical gradient counter

V_n= Negative vertical gradient counter

At each pixel, if d_h exceeds T_e then H_p is incremented. If d_h is less than $-T_e$ then H_n is incremented. If for a block, any one of the gradient counters exceeds or equals the line threshold T_l then that block is classified as an edge block otherwise it is a shade block. Further details regarding CVQ can be found in [12].

2.2.2 Coding of shade blocks

The mean of the shade blocks is predicted from the pixels along the left edge and top edge of the adjacent blocks as shown in Table 2.1. The prediction error of the mean is uniformly quantized into 64 levels. The predicted mean is just the average of the pixels along the left and top edges. By predicting the mean and storing the error in the mean, the storage requirements for the mean decreases from 8 bits to 6 bits for a quantizer of size 64. The actual mean is removed from the shade blocks and the mean removed shade block is coded using the codebook. The codebook is designed using the Linde-Buzo-Gray (LBG) algorithm.

Table 2.1 Predicted mean

m1	m2	m3	m4	m5
m6				
m7				
m8				
m9				

Since the image is scanned from top to bottom and from left to right, the predicted mean considers the pixels along the left and top sides of the 4 x 4 block. The predicted mean is given by:

$$\hat{m} = \frac{1}{9}\sum_{i=1}^{9} m_i \tag{2.4}$$

where m_i is the coded vertical and horizontal pixels bordering the current block.

2.2.3 Design of codebook

The design of the codebook is an important part of vector quantization scheme. The purpose is to find an optimal codebook which minimizes the average distortion. The conditions for optimality are:

1) The quantizer employs a nearest neighbor selection rule. Let X be a vector to be quantized, into one of the m cells ($S_i, 1 \leq i \leq m$). Then

$$Q(X) = Y_i \quad i.e\ (X \epsilon S_i)\ if\ and\ only\ if \quad d(X, Y_i) \leq d(X, Y_j)\ \forall i \neq j, 1 \leq j \leq m \tag{2.5}$$

where $Q(X)$ denotes the closest neighbor in the codebook, Y_i are the individual vectors in the codebook and d is the distance between the vector X and the codevalue Y_i.

2) Each output vector Y_i is chosen to minimize the average distortion D_i in cell S_i.

$$Y_i = 1/N_i \sum_{X \epsilon S_i} X \tag{2.6}$$

$$D_i = 1/N_i \sum_{k=1}^{N_i} d(X_k, Y_i) \tag{2.7}$$

where N_i is the number of vectors that belong to cell S_i. The average distortion of all the codevectors is given by

$$D = 1/N \sum_{i=1}^{m} \sum_{X \epsilon Y_i} \|X - Y_i\|^2 \tag{2.8}$$

The two optimization processes lead to the well known LBG algorithm, which is outlined in the following steps.

- Step 1: Randomly choose an initial codebook C_1
- Step 2: Given a codebook $C_m = Y_i$, partition the training set into clusters using the Nearest Neighbor (NN) rule:
  $S_i = X \epsilon T : d(X, Y_i) \leq d(X, Y_j), \forall j \neq i$
- Step 3: Compute the centroids of the clusters $C_{m+1} = centroid(S_i)$
  $Y_i = 1/N_i \sum_{X \epsilon S_i} X$
- Step 4: Compute the average distortion D_{m+1} for C_{m+1}. If the average distortion has no significant change, $(D_m - D_{m+1})/D_m) < \epsilon$ then stop, otherwise go back to step 2.

2.2.4 Coding of edge blocks

The pixels of the edge block are predicted using the formula shown below.The predicted error is quantized using a non-uniform quantizer of size 32. The predicted value est, at position $(i+1, j+1)$ is given by

$$est = x(i+1, j) + x(i, j+1) - x(i, j) \tag{2.9}$$

where $x(i, j)$ is the gray scale value at pixel location (i,j). The prediction error is given by

$$err = x(i+1, j+1) - est \tag{2.10}$$

The original image is of size 256x160 as shown in Figure 2.1. $T_l = 1$ was chosen so that even the occurrence of one line classifies it as an edge block, since edges are very important in X-ray images. The parameters used are for coding are :

Edge threshold T_e =0.1

Line threshold $T_l = 1$

Block size =4

The indices of the vector quantization, the mean and the dpcm values are further compressed using lossless compression routines. The compressed image is shown in Figure 2.2. The statistics for the image are shown below:

Total no of blocks=2560

No of shade blocks=497

No of edge blocks=2063

Compression ratio=3.508.

The compressed image shown is Figure 2.3 has the following parameters

Edge threshold T_e=0.15

Line threshold T_l=0.1

Block size=4

The statistics for the image are shown below:

Total no of blocks=2560

No of shade blocks=937

No of edge blocks=1623

Compression ration=3.996

By comparing Figure 2.2 with Figure 2.3 it is evident that as the edge threshold is increased the number of shade blocks increases and hence the compression ratio increases. Similarly by keeping T_e constant and increasing T_l will have the same effect on the compression ratio.

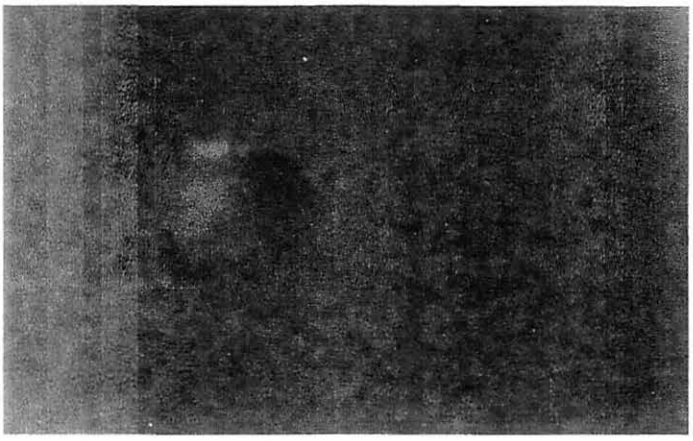

Figure 2.1 Original image

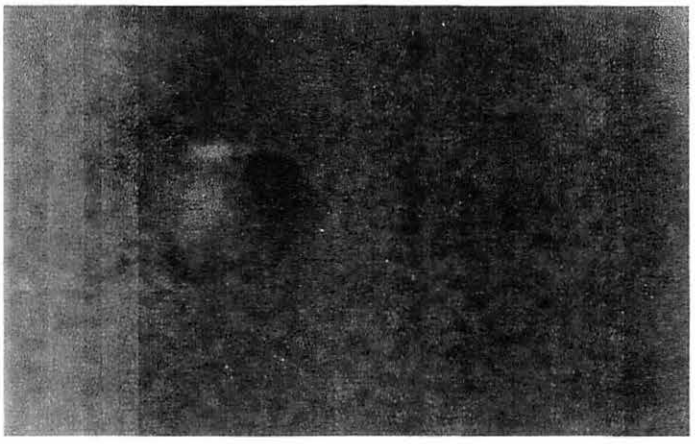

Figure 2.2 Vector quantized image with $T_e = 0.1$ *and* $T_l = 1$

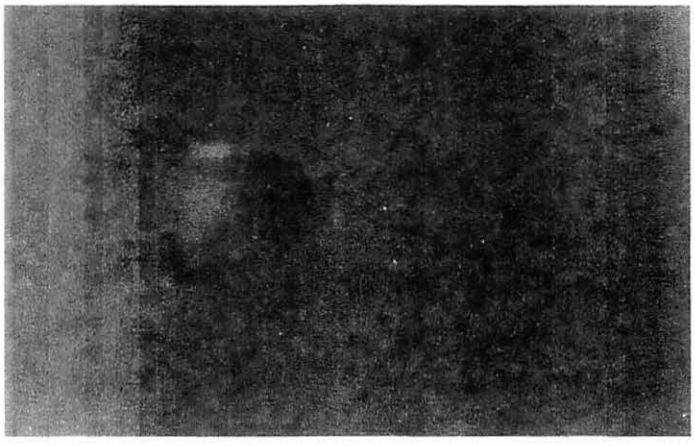

Figure 2.3 Vector quantized image with $T_e = 0.15$ *and* $T_l = 1$

2.3 Fractal image compression

2.3.1 Introduction

Arnaud E. Jacquin proposed a novel approach to image coding based on fractal theory of iterated transformations [9]. We assume that the image redundancy can be exploited by modeling it as fractal objects. The technique is based on iterated contractive transforms in metric spaces. The salient features of this coding method are

- Image redundancy can be exploited through self-transformability
- The original image is approximated by a fractal image

A metric space (X,d) is a space **X** together with a real-valued function d:XxX→R, which measures the distance d between pairs of points x and y in X. The function d should satisfy the following axioms:

$$d(x,y) = d(y,x) \quad \forall x,y \epsilon X \tag{2.11}$$

$$0 < d(x,y) < \infty \quad \forall x,y \epsilon X, x \neq y \tag{2.12}$$

$$d(x,x) = 0 \quad \forall x \epsilon X \tag{2.13}$$

$$d(x,y) \leq d(x,z) + d(z,y) \quad \forall x,y,z \epsilon X \quad (\textit{triangle inequality}) \tag{2.14}$$

Fractal objects have high visual complexity and low information content. They are made of transformed copies of either themselves or part of themselves [2]. Images can be coded by using the fractal concept by considering blocks in the images. This section covers the partitioning of the image, distortion measure and contractive transformations used.

2.3.2 Iterated transform theory

Let (M, d) denote a metric space of digital images. Let μ_{orig} be the image we want to compress or encode. The inverse problem is to construct a contractive image transformation τ for which μ_{orig} is an approximate fixed point. The requirements for the code τ are:

$$\exists s < 1 \quad such\ that \quad \forall \mu, \nu \in M, d(\tau(\mu), \tau(\nu)) \leq sd(\mu, \nu) \quad (contractivity) \tag{2.15}$$

$$d(\mu_{orig}, \tau(\mu_{orig})) \quad is\ as\ small\ as\ possible \quad (approximate fixed point) \tag{2.16}$$

where μ, ν are images and s is called the contractivity of τ. If τ has lower memory requirements for storage, then τ is a lossy code for μ_{orig}. For any image μ_0 and any positive integer n:

$$d(\mu_{orig}, T^n(\mu_0)) \leq \frac{1}{1-s} d(\mu_{orig}, \tau(\mu_{orig}) + s^n d(\mu_{orig}, \mu_0) \tag{2.17}$$

Equation 2.17 gives the error bounds for image compression as a function of the contractivity s. The forward iterates of τ is given by $\tau^2(\mu) = \tau(\tau(\mu))$ where $\tau^2(\mu)$ is the second iteration of the transformation. μ is a fixed point of a transformation if $\tau(\mu) = \mu$.

2.3.3 Construction of the fractal code

An original image μ is partitioned into non-overlapping blocks called range cells. Fractal compression tries to utilize the self-similarity that is found in certain regions of the image. The image will be formed of copies of properly transformed parts of itself. Each transformation selects a portions of the original image, which we denoted by D_i and copies that part with a brightness (o) and contrast (s) transformation to a part of the produced copy which is denoted by R_i. We call D_i domains and R_i ranges. We denoted the transformations by τ_i. For example let us consider a 128x128 image with 256 gray levels. Let $R_1, R_2, ..., R_{256}$ be the 4x4 non-overlapping sub-squares of the image and

let D be the collection of all 8x8 (overlapping) sub-squares of the image. The collection D contains 121x121=14641 squares. For each R_i search through all of D to find $D_i \in D$ which minimizes equation 2.23; that is find the part of the image that most looks like the image above R_i. This domain is said to cover the range. There are 8 ways to map one square to another (massic transformations H_i), this means comparing 8x14641=117128 squares with each of the 1024 ranges.

A square in D has 4 times as many pixels as in R_i, so we must sub-sample D_i and this is done by the geometric transformation G_i. A choice of D_i, along with s_i and o_i determine a map τ_i. This is the basic fractal compression algorithm. Different variations of this algorithm can be used. The different variations of the algorithm help to reduce the domain space D and also the number of computations.

The image transformations are defined blockwise. They are of the form

$$\forall \mu \in M, \tau(\mu) = \sum_{0 \leq i < N} (\tau\mu)|R_i = \sum_{0 \leq i < N} \tau_i(\mu|D_i) \tag{2.18}$$

where $R_i, 0 \leq i < N$ denotes a nonoverlapping partition of the image into N range cells. $\mu \mid R_i$ denotes the restriction of the image μ to the cell R_i. τ_i is the composition of two transformations G_i and H_i. $\tau_i = H_i \circ G_i$, where G_i is the geometric transformation and H_i is the massic transformation of τ_i. The fractal block coding of the μ_{orig} amounts to finding for every R_i a transformation from the domain D_i such that the transformed domain block $\tau(\mu_{orig} \mid D_i)$ is close to the original range block in the image($\mu_{orig} \mid R_i$). The main issues are:

- Image dependent partitioning of the image
- Specification of a class of discrete contractive transformations
- Quantization of the parameters of the transform

2.3.4 Image partition

The original image μ_{orig} is partitioned into nonoverlapping square range cells. Using a quadtree partitioning scheme, a square image is repeatedly partitioned into 4 equally sized sub-squares, until the subsquare covers the domain of interest fully. This process can be repeated until the squares are covered within a specified error limit. For a 128x128 image the square of size 128x128 is level 0 of the partition and square of size 64x64 is level of the partition. As the level of the partition increases the range block size decreases. We can also specify a minimum and maximum levels of the quadtree partition. Depending on the image statistics and block similarity, the size of the blocks that can be coded vary. Small squares can be covered better than larger ones. This partitioning allows the coder

- to use large blocks in smoothly varying areas
- to use small blocks in complex areas (edges and textures)

2.3.5 Discrete image transformations

We describe contractive transformations that operate on blocks.

1) Geometric Part G_i:

The transformation G_i maps image blocks from a domain cell $D_i = S(i_d, j_d, D)$ to a range cell $R_i = S(i_r, j_r, B)$, where $S(i_r, j_r, B)$ is a image block of size BxB starting at pixel location (i_r, j_r). In our case we use D=2B. The domain considered is twice the size of the range. To get the transformed domain of the same size as the range we decimate the domain. The pixels values of the contracted image on the range block $S(i_r, j_r, B)$ are the average of the four pixels in the domain block:

$$(G_i\mu)_{i_r+i,j_r+j} = (\mu_{I(i),J(j)} + \mu_{I(i)+1,J(j)} + \mu_{I(i),J(j)+1} + \mu_{I(i)+1,J(j)+1})/4, \qquad (2.19)$$
$$\forall\, i,j \in 0,1,...,B-1$$

$$I(i) = i_d + 2i \quad and \quad J(j) = j_d + 2j \tag{2.20}$$

2) Massic Part H_i:

Massic transformations process image blocks supported on a range cell R_i. These transformations affect the gray scale values of the pixels of the transformed domain block whose size is the same as that of the range block. The general massic transformation is as shown below

$$(\tau\mu)_{i,j} = s\mu_{i,j} + o \tag{2.21}$$

where s is the scale factor and o is the offset. Given two squares containing n pixel gray values $a_1, a_2, ..., a_n$ (from the domain) and $b_1, b_2, .., b_n$ from the range we can seek the scale factor s and offset o to minimize

$$R = \sum_{i=1}^{n} (s.a_i + o - b_i)^2 \tag{2.22}$$

This gives s and o which minimize the rms error between the range and the transformed domain. The minimum occurs when the partial derivates with respect to s and o are zero, which occurs when [9]

$$s = \frac{[n^2(\sum_{i=1}^{n} a_i b_i) - (\sum_{i=1}^{n} a_i)(\sum_{i=1}^{n} b_i)]}{[n^2 \sum_{i=1}^{n} {a_i}^2 - (\sum_{i=1}^{n} a_i)^2]} \tag{2.23}$$

$$o = \frac{[\sum_{i=1}^{n} b_i - s \sum_{i=1}^{n} a_i]}{n^2} \tag{2.24}$$

The parameters s and o are quantized uniformly to reduce storage, depending on the number of bit allocated to store the scaling and offset values.

We also use some isometries to simply shuffle pixels within a range block. A list of eight canonical isometries follows:

1) Identity:

$$(\ell_0\mu)_{i,j} = \mu_{i,j} \tag{2.25}$$

2) Orthogonal reflection about mid-vertical axis:

$$(\ell_1\mu)_{i,j} = \mu_{i,B-1-j} \tag{2.26}$$

3) Orthogonal reflection about mid-horizontal axis:

$$(\ell_2\mu)_{i,j} = \mu_{B-1-i,j} \tag{2.27}$$

4) Orthogonal reflection about first diagonal(i=j):

$$(\ell_3\mu)_{i,j} = \mu_{j,i} \tag{2.28}$$

5) Orthogonal reflection about second diagonal (i+j=B-1):

$$(\ell_4\mu)_{i,j} = \mu_{B-1-j,B-1-i} \tag{2.29}$$

6) Rotation through 90°:

$$(\ell_5\mu)_{i,j} = \mu_{j,B-1-i} \tag{2.30}$$

7) Rotation through 180°:

$$(\ell_6\mu)_{i,j} = \mu_{B-1-i,B-1-j} \tag{2.31}$$

8) Rotation through -90°:

$$(\ell_7\mu)_{i,j} = \mu_{B-1-i,j} \tag{2.32}$$

Massic transforms generate a family of blocks which can be used for coding.

2.3.6 Image reconstruction from a fractal code

The natural decoding scheme simply consists of iterating a code τ on any initial image μ_0, until convergence to a stable decoded image is obtained. The sequence of images $\mu_n = \tau^n(\mu_0)$ is called the reconstruction sequence for the code τ. For each cell index i, the transformation τ_i is applied to the current image block over the domain cell D_i, and mapped onto the range cell R_i.

The transformation in τ is affine and has the form

$$\tau(\mu) = L(\mu) + \nu_0 \tag{2.33}$$

where L is a contractive linear image transformation, and ν_0 is an image which is uniform in the block sense. The Nth iteration in the reconstruction process is given by [9]

$$\mu_N = \sum_{i=0}^{i=N-1} L^i(\nu_o) + L^N(\mu_0) \tag{2.34}$$

The term $L^i(\nu_0)$ is the ith order term of the expansion. The finite sum

$$\sum_{i=0}^{i=N-1} L^i(\nu_0) \tag{2.35}$$

is the partial sum of order N-1, and the residual term $L^N(\mu_0)$ represents the remainder of the expansion. From Banach theory, we can show that

$$\|L^N(\mu_0)\| \leq s^N \|\mu_0\| \tag{2.36}$$

which, since s < 1, is small for large N. The original image is shown in Figure 2.4. The absolute values of the expansion terms $L(\mu_0)$ and $L^2(\mu_0)$ are shown in Figure 2.5 and Figure 2.6 respectively.

2.3.7 Suggested improvements

A weakness of the quadtree partitioning is that it does not select the domain D based on its content. A way to remedy this, while increasing range partition is to use a Horizontal Vertical (HV) partition. In HV partitioning, a rectangular image is recursively is partitioned either horizontally or vertically to form two new rectangles instead of four in the quadtree partitioning scheme. The process repeats until the desired tolerance level is achieved.

Another way to partition an image is based on triangulation. In triangular partitioning, a rectangle is divided into two triangles. Each of these triangles is subdivided into 4 triangles by segmenting the triangle along lines that join three partitioning points along the sides of the triangles. The artifacts arising from the process do not run horizontally or vertically, which is therefore less distracting.

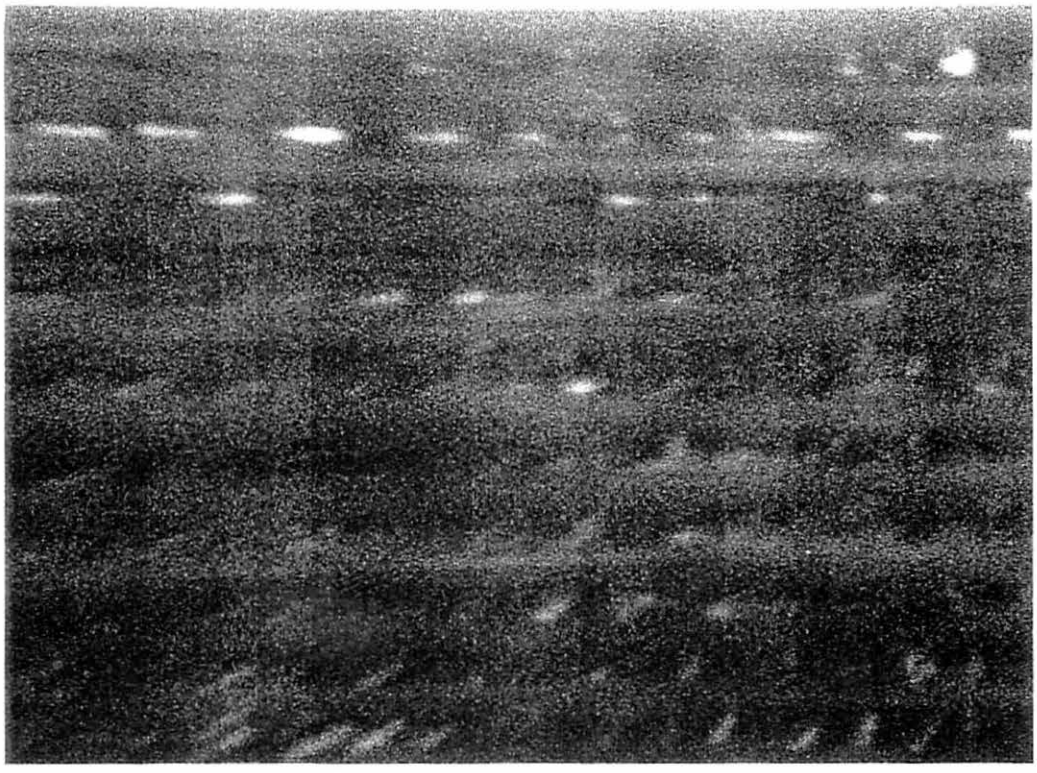

Figure 2.4 Original image of composite 'post12'

2.3.8 Comparison of fractal block coding and vector quantization

The bit rates that can be obtained using fractal block coding schemes depend both on the system parameters and also on the complexity of the image to be encoded. The similarities between fractal coding and VQ is given in Table 2.2. Fractal coding is a variable rate coder. Fractal coding is time consuming when compared to Vector Quantization (VQ). The salient features of the fractal coder are:

- Domain blocks are not needed at the decoder and hence it is called a "virtual codebook" scheme
- The encoding does not use a codebook but exploits the redundancy present in the image

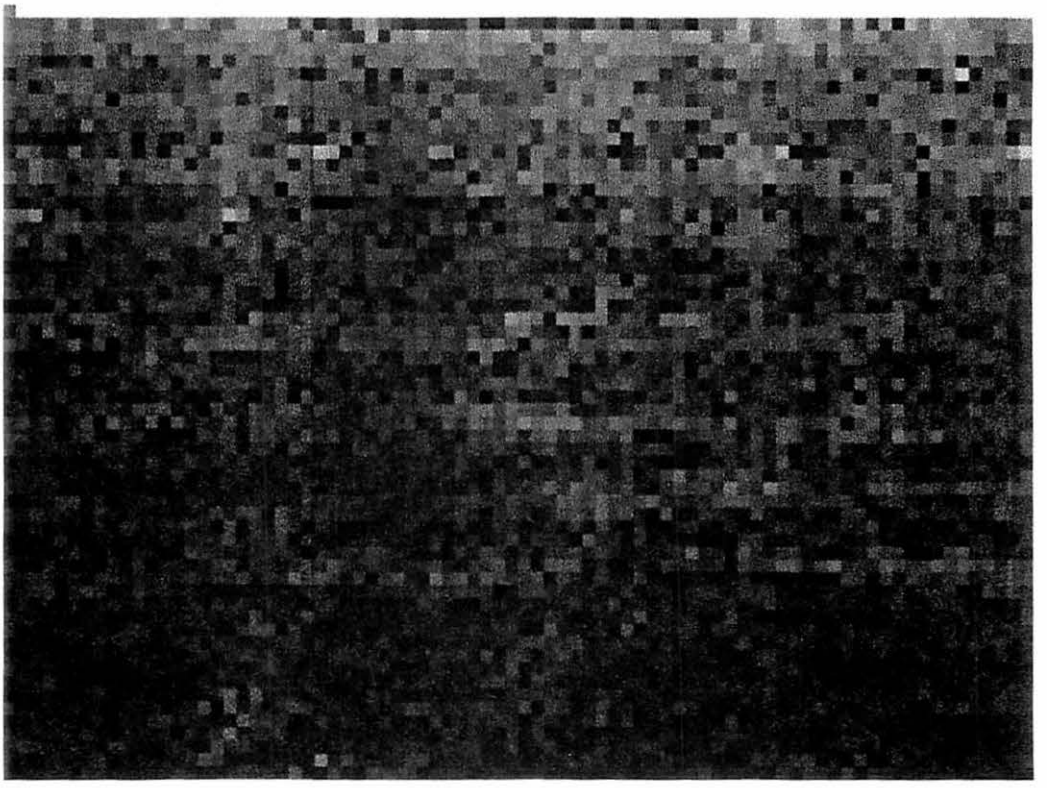

Figure 2.5 Absolute value of the first term of the expansion for 'post12'

2.4 DCT based coding

The JPEG (Joint Photographic Experts Group) compression method computes the Discrete Cosine Transform (DCT) of the image blocks. The DCT coefficients are then quantized depending on the quality factor and the statistics of the coefficients from the different blocks. JPEG is a transform domain based compression method. The advantage of transform compression methods is that the transformed coefficients are better correlated and hence better suited for compression when compared to the original image. One of the features of the JPEG coding method is that it is an asymmetrical coding method. The time for compressing and the decompressing an image are equal.

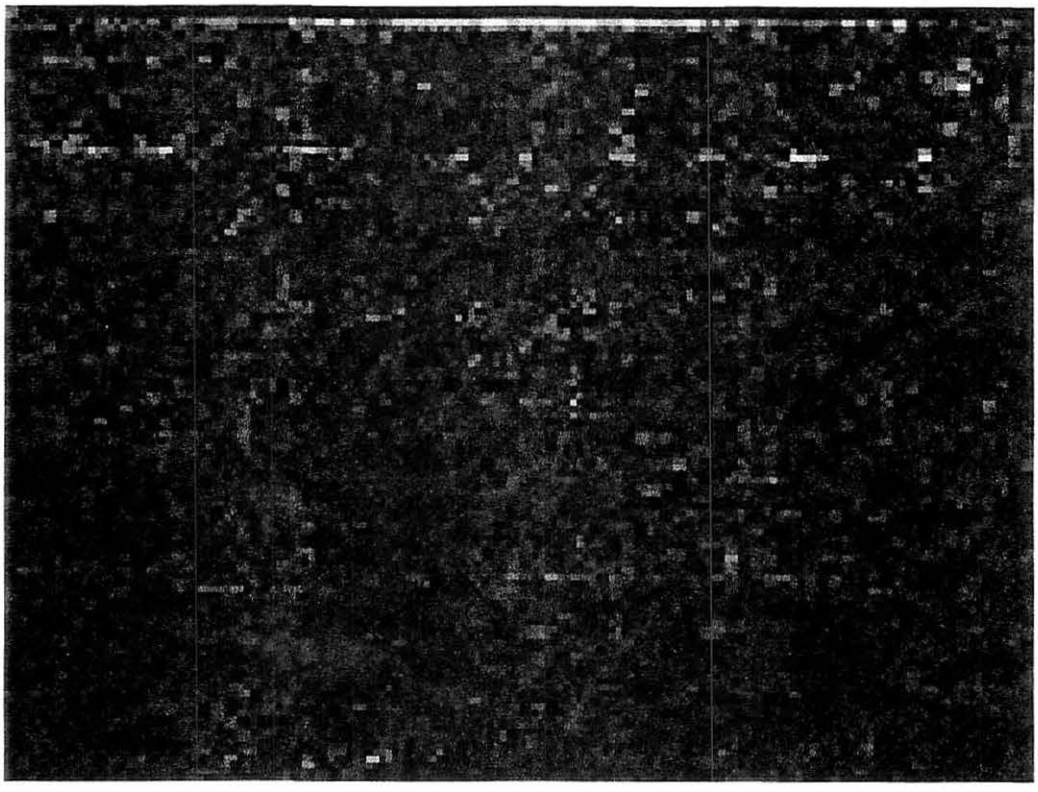

Figure 2.6 Absolute value of the second term of the expansion for 'post12'

2.5 Results

In vector quantization $T_e = 0.1$ was chosen, since at this value the compression did not introduces visible artifacts for the classes of images considered. It is observed that it takes at least eight iterations before the decompressed image stabilizes. Figures 5.5 through 5.8 are decompressed images for 'post12' for one, two, four and eight iterations where the initial image was 'post9'. It can be inferred from Figure 5.1 and Figure 5.8 that after eight iterations the decompressed images do not differ much, even though the starting images are different. All the images compressed using fractal technique used minimum quadtree level=3, maximum quadtree level=5 and tolerance 1 except for the coding in Table 5.1 where tolerance=0.2, minimum quadtree level=4 and maximum quadtree level=6 for images 'post9' and 'post12'. For 'flat5mil' minimum quadtree level=3, maximum quadtree level=5 and tolerance=0.2. The image 'flat5mil' is the

Table 2.2 Features common to VQ and fractal coding

VQ	Fractal Coding
Partition	Partition
Partitioning of image into square blocks	Partitioning of image into range blocks
Different block sizes possible	Different block sizes possible
Codebook	**Virtual Codebook**
Training images needed	Decimated and processed domain blocks extracted from the original image
Codebook design using LBG	Reducing the search space of the domain pool is feasible
Off-line transmission of codebook	No codebook needed at decoder
Block Matching	**Encoding of range blocks**
Selection of a block distortion measure	Same type of blocks are used for coding
Block classification to retain the geometric nature of a block	
Decoding	Decoding
Image is represented by the list of addresses of blocks in the codebook	Fractal code consists of block transformations on an image partition
Reconstruction by lookup table	Iterative reconstruction
Execution Time	Execution Time
Equal coding and decoding time	Slow encoding time and fast decoding

image of flat plate of thickness 1 inch and the flaw is 5 mil thick. The Figure 2.15 shows the post12 image compressed using JPEG with the quality factor Q=0.7.

From the Table 5.1 it can been inferred that the compression ratio achieved by VQ is close to that of JPEG. The lossless compression ratio is the theoretical maximum amount of compression possible based on the entropy of the image. Fractal coding does better than VQ and JPEG for image like post12 where there is much self similarity among the blocks. Fractal coding achieves good coding of the flat5mil whereas other methods have poor compression ratios. The number of scaling and offset bits used in fractal coding are 5 and 7 respectively.

Let a_i be the gray scale values of the original image and b_i be the gray scale values

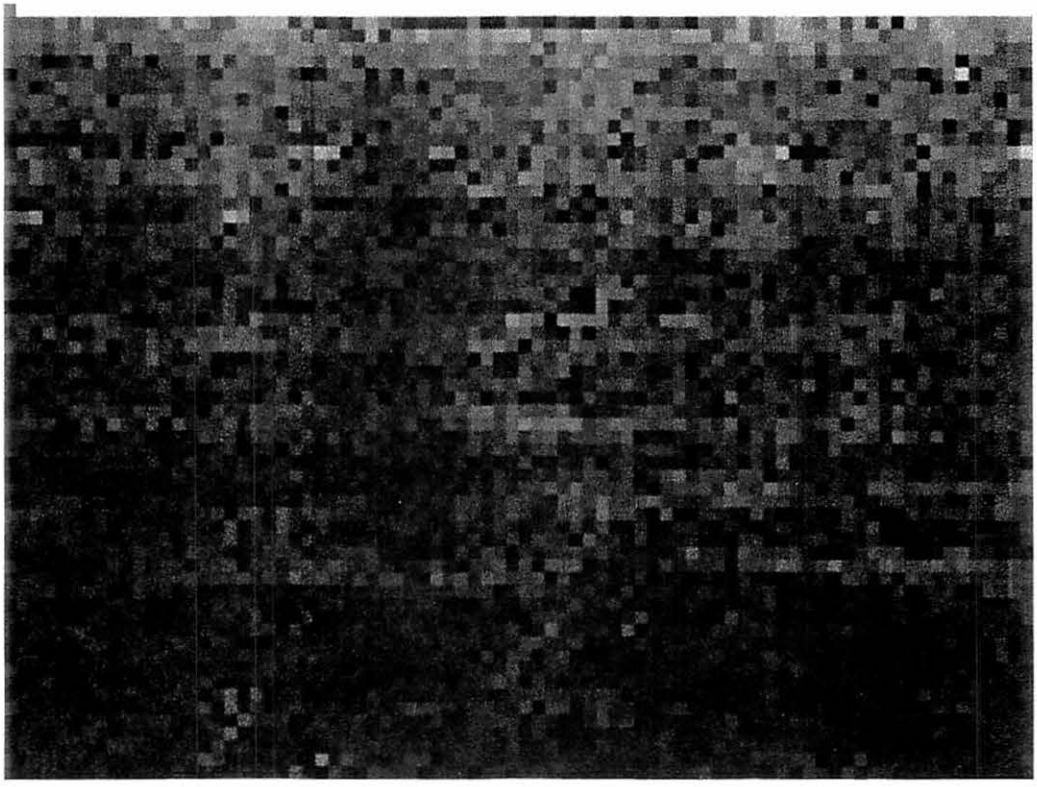

Figure 2.7 Decompressed image for 'post12' after one iteration of the code

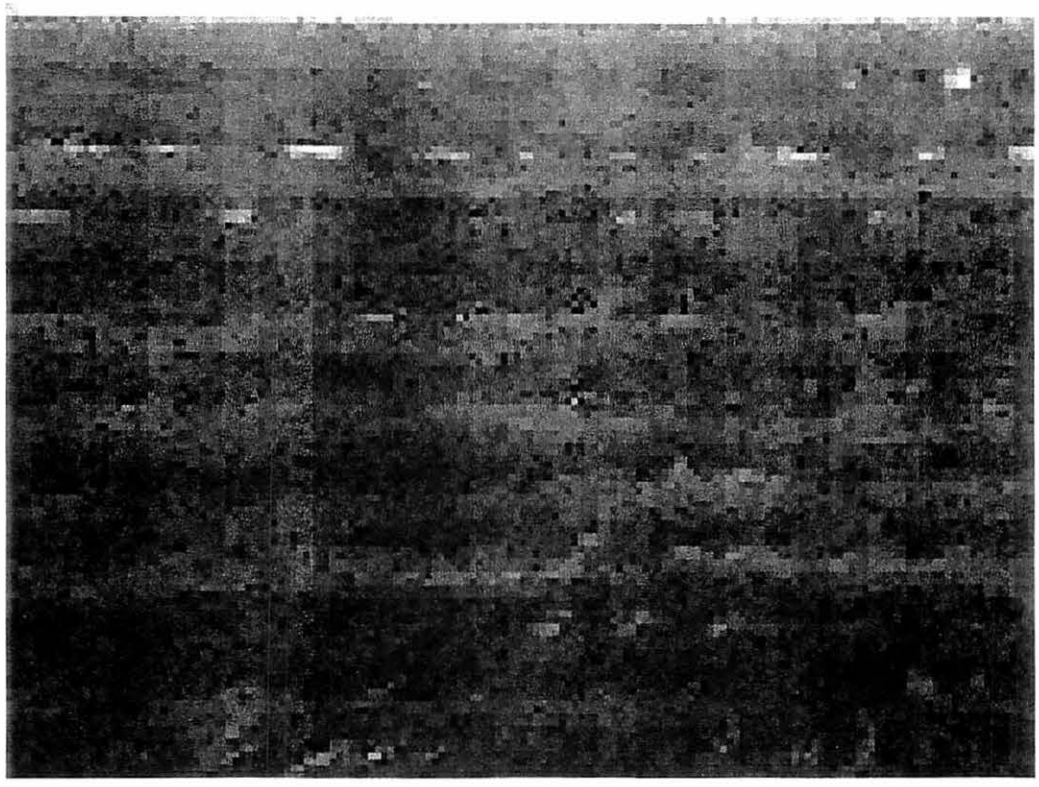

Figure 2.8 Decompressed image for 'post12' after two iterations of the code

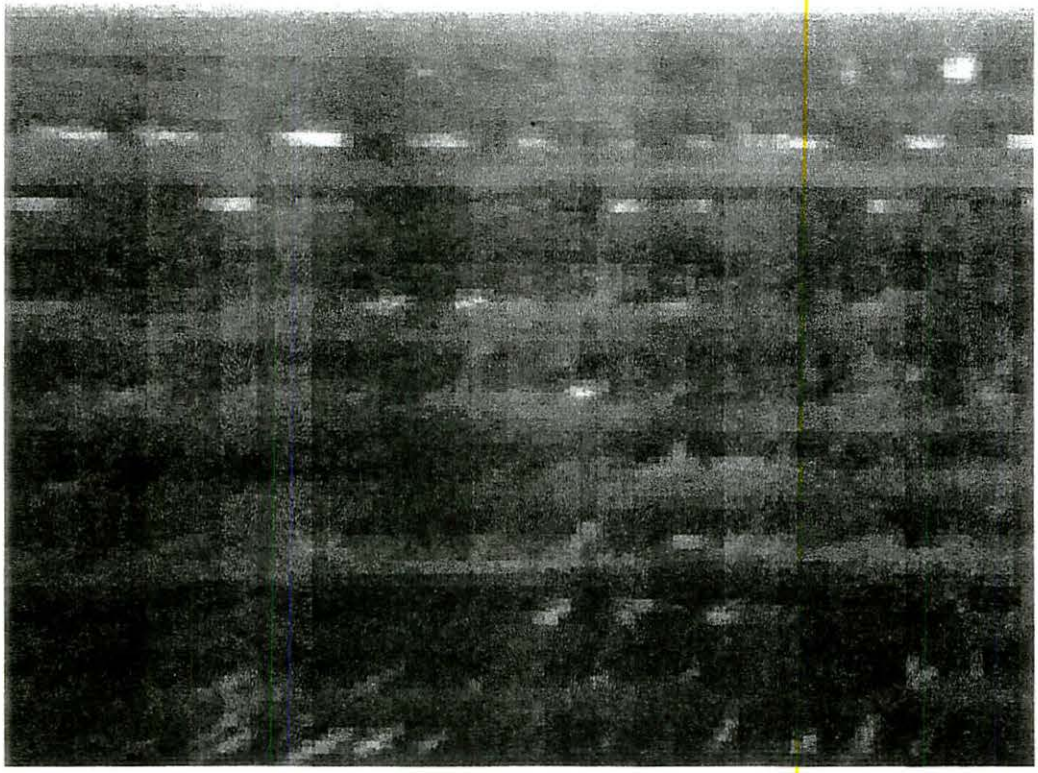

Figure 2.9 Decompressed image for 'post12' after four iterations of the code

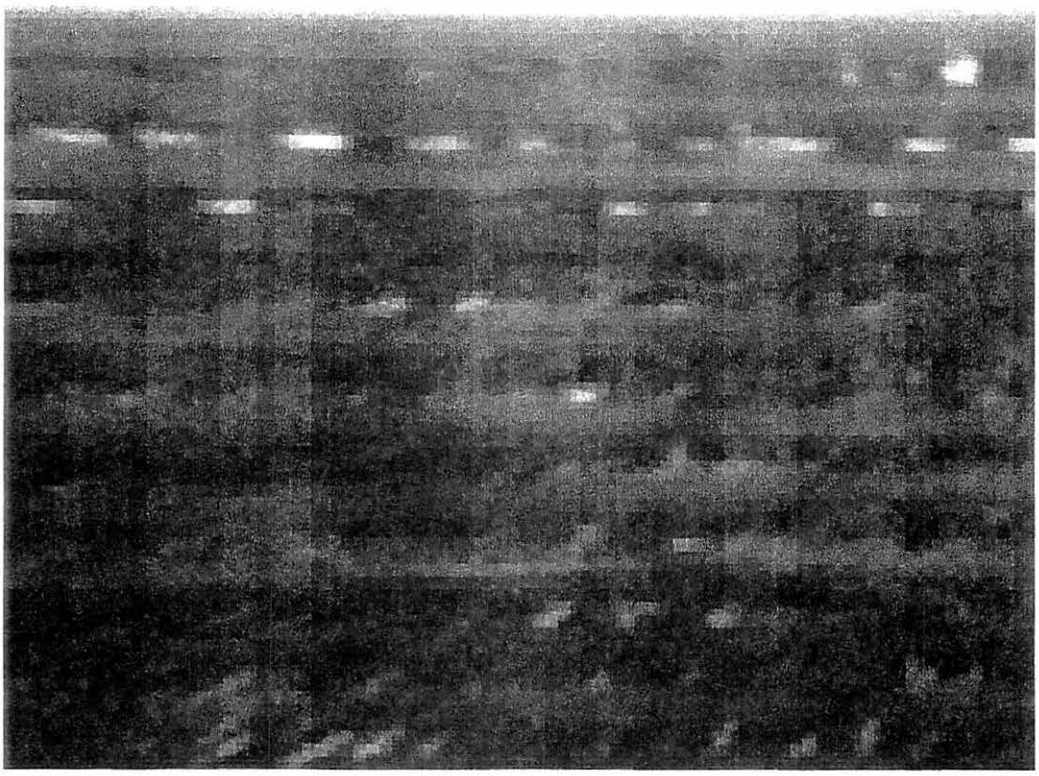

Figure 2.10 Decompressed image for 'post12' after eight iterations of the code

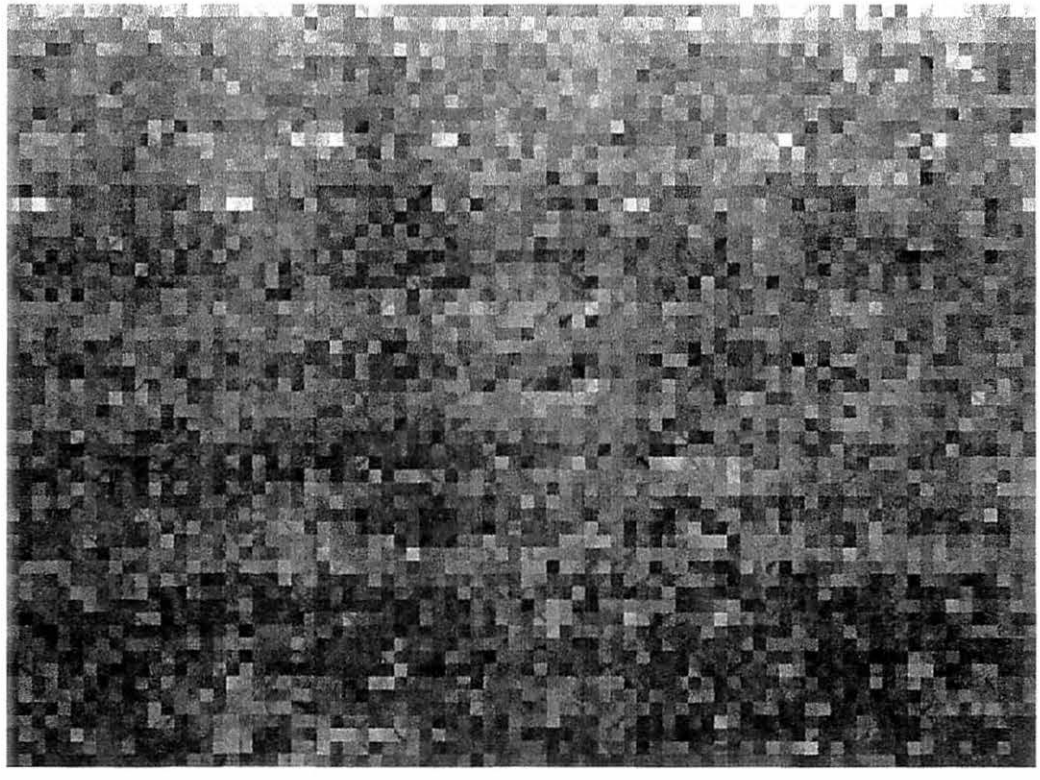

Figure 2.11 Decompressed image for 'post12' after one iteration with 'post9' as the initial image

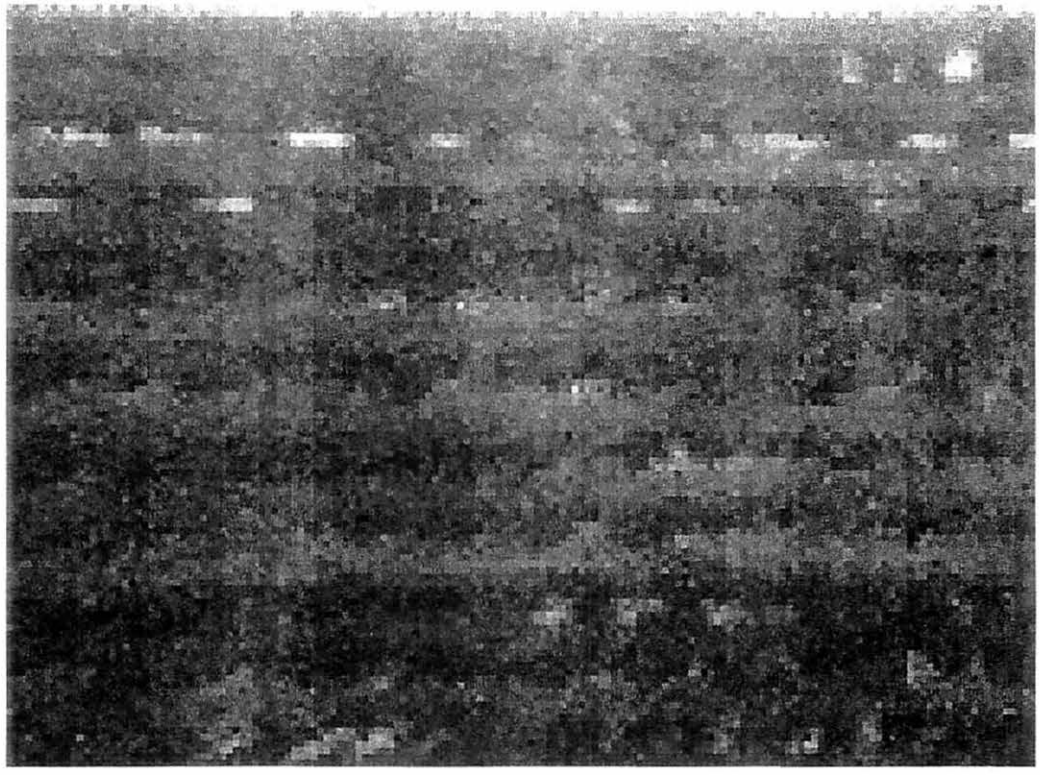

Figure 2.12 Decompressed image for 'post12' after two iterations with 'post9' as the initial image

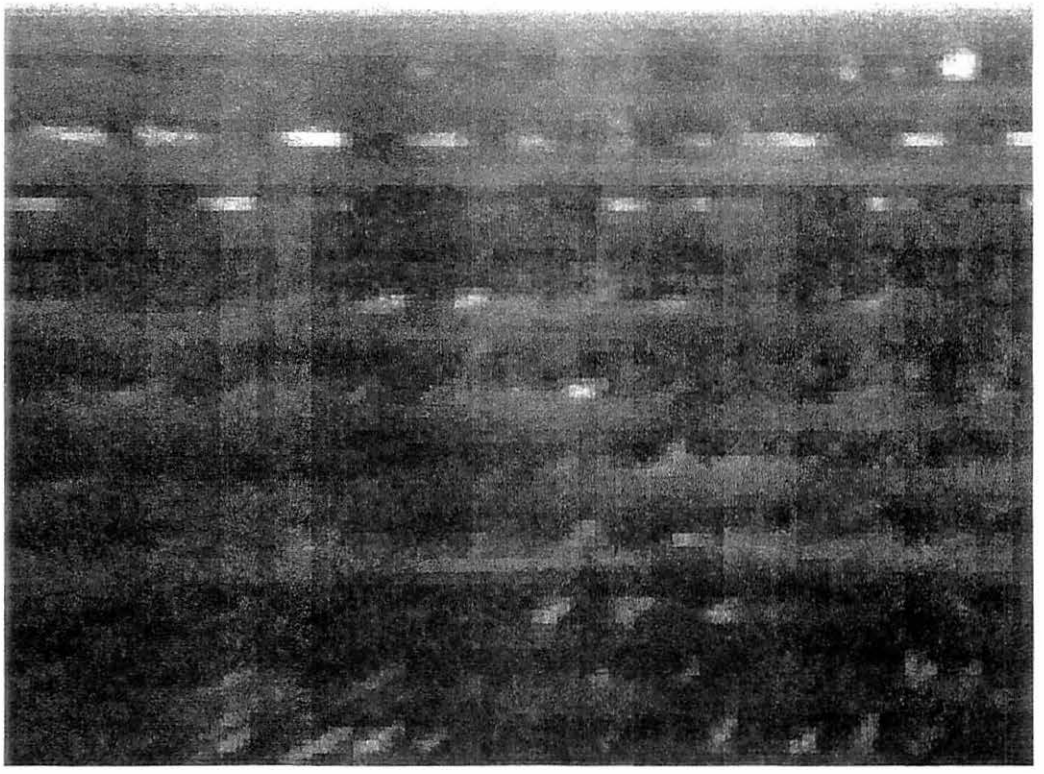

Figure 2.13 Decompressed image for 'post12' after four iterations with 'post9' as the initial image

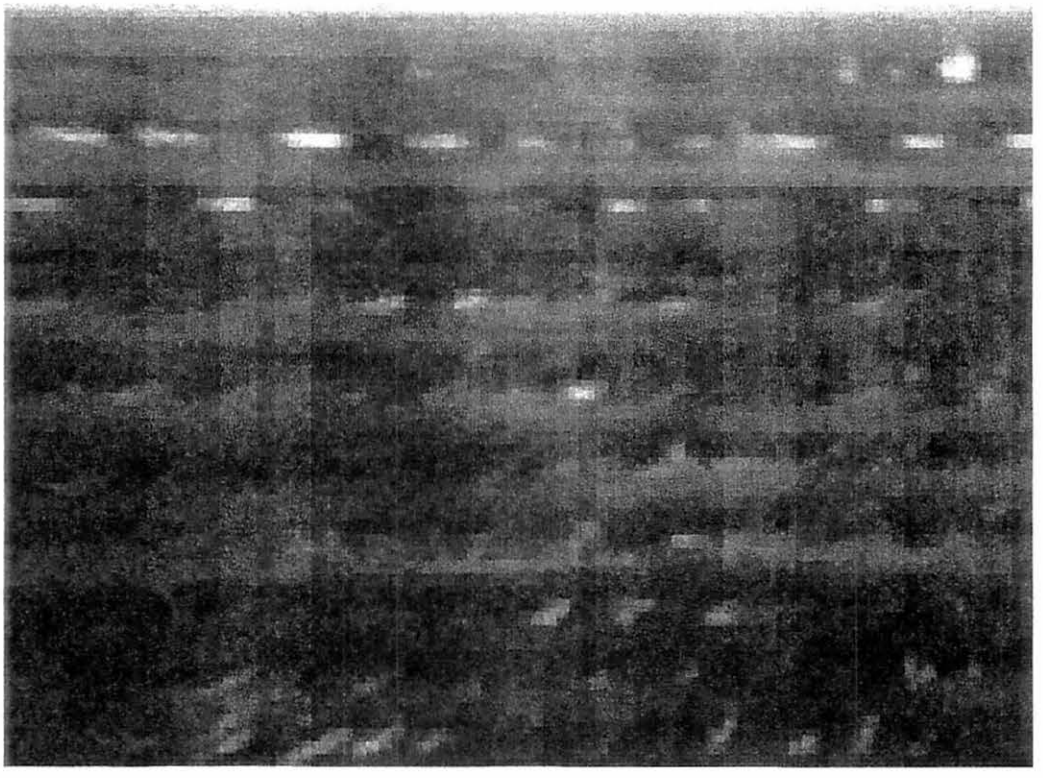

Figure 2.14 Decompressed image for 'post12' after eight iterations with 'post9' as the initial image

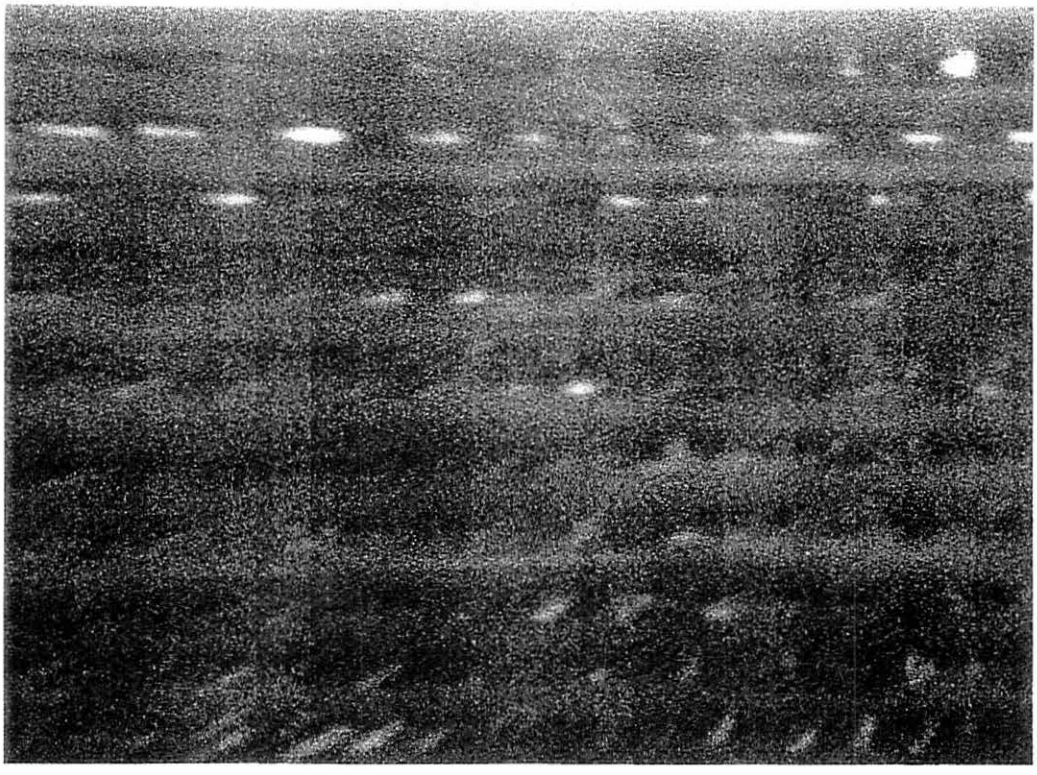

Figure 2.15 Decompressed image for 'post12' using JPEG

of the decompressed image. The signal to noise ratio (SNR) is defined as:

$$SNR = 10log_{10}\frac{\sum(a_i)^2}{\sum(a_i - b_i)^2} \tag{2.37}$$

The SNR values for different values of the number of sbits (scale factor) and number of obits (offset) are shown in Table 2.4. All the images were compressed using a quadtree scheme with minimum depth 3 and maximum depth 5 and tolerance 1. From Table 2.4 it can be concluded that we need at least 5 scaling bits and 5 offset bits to represent

Table 2.3 Compression ratio for various compression methods

Image	Lossless	VQ THL=1 THS=0.1	VQ THL=2 THS=0.15	JPEG Q=0.70	Fractal
autof	1.654	8.100	11.520	11.060	5.281
post9	1.410	7.465	7.717	10.450	4.520
post12	1.015	1.927	1.926	4.067	4.596
flat5mil	1.110	1.613	1.607	2.530	5.258

the image better using the SNR criterion. The parameter sbits and obits refer to the number of bits used to represent the contractivity (scale) factor and the offset o. From Table 2.5 it is evident that as the number of minimum levels in the quadtree partition increases the compression ratio decreases as the size of the range blocks get smaller and the number of transformations used to represent the image increases. Similarly for a fixed minimum level in the quadtree partition as the number of the maximum level in the quadtree partition increases the compression ratio decreases.

Table 2.4 Values of SNR between the original 'post12' image and successive terms of the reconstruction sequence with initial black image

Iteration Number	sbits=5 obits=5	sbits=5 obits=6	sbits=5 obits=7	sbits=4 obits=6	sbits=6 obits=6
1	6.2540	6.0533	6.0529	6.1081	6.0438
2	10.9536	10.7844	10.7723	10.837	10.7372
3	13.2665	13.2319	13.2239	13.2503	13.1956
4	14.0842	14.1221	14.1351	14.1107	14.1064
5	14.2985	14.3699	14.3951	14.3500	14.3673
6	14.3395	14.4185	14.4478	14.3965	14.4200
7	14.3475	14.4279	14.4580	14.4063	14.4309
8	14.3488	14.4297	14.4601	14.4085	14.4329

Table 2.5 Compression ratio for 'post12'

Compression Ratio	Tolerance	Minimum level	Maximum level	sbits	obits
21.6032	1	3	5	5	6
21.5968	1	3	5	5	5
21.4926	1	3	5	4	6
19.8012	1	3	5	6	6
19.7859	1	3	5	5	7
4.5973	1	4	6	5	7
19.8334	1	4	5	5	7
19.7849	0.2	3	5	5	7
19.7849	0.01	3	5	5	7

3 VISUAL DIFFERENCE PREDICTION

3.1 Introduction

The term image fidelity is related to the visual equivalence of two images. Subjective rating scales do not provide an accurate or repeatable measure to measure image fidelity. We need an objective measure which measures the fidelity of images. A visual model is used to predict the visual differences between the compressed and the original images. There are many possible models yielding different estimates of image fidelity. The visual difference predictor (VDP) is used to calculate how the mathematical differences in gray scales between images translate into visual differences [13]. The commonly used methods like MTF and noise power spectra calculate a single number for describing image quality. The VDP addresses the visibility of differences between two images rather than the visibility of a single image. The input to the algorithm includes two images and parameters for viewing conditions and calibration. The output is an image representing the visible differences between the two images.

The response of the eye is nonlinear with respect to luminance and this is modeled by the amplitude nonlinearity. A image once captured by the receptors in the eye, pass through numerous layers of cells, each of which, by means of its input/output relation, imposes some transformation of the data. Each neuron in the eye has a receptive field (the region of the retina within which light will cause some change in response of the cell). The visual cortex receives all of the cortex projection from the retina via lateral geniculate nucleus (LGN). A portion of the visual cortex called V1 has a type of cell called

simple cells. The simple cells will respond to a range of frequencies and orientation about its center values, and this range is characterized by spatial frequency and orientation bandwidths of the cortex. The receptive field (RF) of each cell occupies only a portion of the total frequency region to which the collection of cells is sensitive. The ideal RF is a 2D Gabor function (the product of 2D sinusoid and 2D Gaussian).

To predict the differences between two images the initial step is to transform the image into a representation as seen by the eye. The cortex filter described in the following section is a digital approximation of the ideal RF.

The VDP consists of the following components

- Calibration for input images
- A Human Visual System (HVS) Model
- Method for displaying the HVS predictions of visible differences

3.2 HVS model

The HVS model attempts to capture the lower order processing of the visual systems, such as the optics, retina, lateral semiculate nucleus and striate cortex. The approach is to model the visual system as a number of processes that limit the sensitivity of the eye. Two images having the same Mean Square Error (MSE) may differ in the degree of visibility of the error. This fact necessitates the use of the HVS model rather than a physical model of the displayed image. Physical models using parameters like mean squared error often fail to determine the visibility of the error. The HVS model address sensitivity variations with respect to the following issues

- Light intensity
- Spatial Frequency

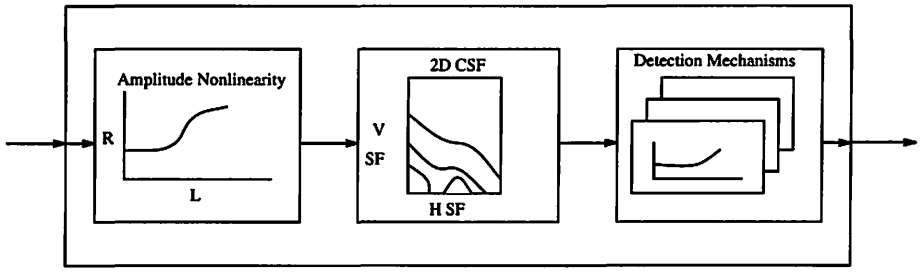

Figure 3.1 HVS model

- Signal Content

Sensitivity is defined as the inverse of the contrast required to produce a threshold response,

$$S = \frac{1}{C_T} \tag{3.1}$$

where C_T is referred to as the threshold. The Michelson definition of contrast,

$$C = \frac{L_{max} - L_{mean}}{L_{mean}} \tag{3.2}$$

where L_{max} and L_{mean} are the maximum and mean luminance of the waveform. The variation in sensitivity as a function of the light intensity is due to the nonlinear response of the eye and is called the amplitude nonlinearity of the HVS and is implemented as a point process; variation in sensitivity as a function of spatial frequency are due to the neural response of the eyes and optics and is referred to as the contrast sensitivity function (CSF), implemented as a filtering process; variations in sensitivity as a function of signal content are due to the postreceptoral neural circuits and these effects are known as masking. Masking is implemented as a combination of filters and nonlinearities (detection mechanisms). These three components are modeled separately and are cascaded as shown in Figure 3.1.

3.2.1 Amplitude nonlinearity

Visual sensitivity is a nonlinear function of luminance. The amplitude nonlinearity describes variations in sensitivity as function of gray scale and background illumination. It is assumed that the observer is able to focus on a small image area. The amplitude nonlinearity as a function of pixel luminance is given by [15]

$$R/R_{max} = \frac{L}{(L + (c_1 L)^b)} \tag{3.3}$$

where, R/R_{max} is the normalized response. L is the luminance in cd/m^2, b and c_1 are constants equal to 0.63 and 12.6 respectively. The sensitivity of each pixel is determined from that pixel only. Although the visual system cannot adapt to very small areas of an image, we assume that the observer may view the image at any close distance. This removes any frequency attributes from the amplitude nonlinearity component and facilitates modeling the CSF as a separate block. A logarithmic response for the amplitude nonlinearity overestimates the detectability of differences in the low gray scale range.

3.2.2 Contrast sensitivity function

The CSF models the variation in sensitivity of the eye as a function of spatial frequency. The variations are due to the sampling aperture of the cone photoreceptor, and both passive and active neural connections [17]. The CSF changes as a function of light adaptation, noise, color, accommodations, eccentricity and image size. Equation 3.2 models the sensitivity S, as a function of radial frequency ρ in cy/deg, orientation θ in degrees, light adaptation level l in cd/m^2, image size i^2 in visual degrees, lens accommodation due to distance d in meters, and eccentricity e in degrees, [13]

$$S(\rho, \theta, l, i^2, d, e) = P.min[S(\frac{\rho}{bw_a, bw_e, bw_\theta}, l, i^2), S(\rho, l, i^2)] \tag{3.4}$$

$bw_a = \ 0.856\ d^{0.14}$ where d is the distance in meters

$bw_e = \frac{1}{1+ke}$ where e is the eccentricity in visual degrees, k=0.24

$bw_\theta = ((1 - ob)/2)\cos(4\theta) + (1 + ob)/2$ where ob=0.7

P is the absolute peak sensitivity whose value is 250 [17]. The parameters bw_a, bw_e, bw_θ model the changes in bandwidth due to the accommodation level, eccentricity and orientation. Eccentricity is the location in the visual field. The fovea, or center is at 0°. Orientation is the orientation of the spatial frequency modulation in the image. Vertical stripes are modulated horizontally and have $\theta = 0$. For each digital frequency (i,j) the corresponding (ρ, θ) is calculated using cartesian to polar conversion. The frequency ρ in cy/deg is calculated as follows

$\rho(cy/deg) = \pi * d * (cy/mm)/180$

where, cy/mm is the physical spacing of the pixels in the displayed image. For a 256x256 image and for a viewing distance of 91.2 cm and a display resolution of 20 pixels/cm the highest image frequency is 16 cycles/deg. The sensitivity as function of image size and light adaptation level is given by [13]

$$S(\rho, l, i^2) = ((3.23(\rho^2 i^2)^{-0.3})^5 + 1)^{-\frac{1}{5}} A_l \epsilon \rho e^{-(B_l \epsilon \rho)} \sqrt[2]{1 + 0.06 e^{B_l \epsilon \rho}} \tag{3.5}$$

$$A_l = 0.801(1 + 0.7/l)^{-0.2} \tag{3.6}$$

$$B_l = 0.3(1 + 100/l)^{0.15} \tag{3.7}$$

where ϵ is a frequency scaling constant whose value is 0.9. Equation 3.3 models sensitivity due to image size, the second part models sensitivity and bandwidth as a function of light level. The CSF is modeled in the units of cy/deg and then mapped to the digital frequency domain. The calibration parameters are viewing distance (d), horizontal and vertical pixel spacings, image size and light level (l). In models based on spatial frequency, the approach is to weight the sensitivities of the different channels to get an aim CSF. The problem with this technique is that it is difficult to include the adaptive behaviors of the CSF since it is not decoupled from the multiple channels. The VDP technique of preceding the cortex filter (Section 3.3) with the CSF causes a distortion in the shape of the frequency response from that of the cortex filters.

3.2.3 Detection mechanisms

The final component of the HVS is comprised of multiple detection mechanisms. The components are

- Spatial frequency hierarchy that models the frequency selectivity of the eye
- Masking function that models the masking magnitude
- Psychometric function that deals with the threshold levels of detection
- Probability summation which combines the outputs of all detection mechanisms into a single response

3.2.3.1 Spatial frequency hierarchy

The spatial frequency selectivity of the visual system is characterized by a radial spatial frequency and orientation. The visual cells in the cortex of the eye respond to a range of frequencies and orientations about the center values, which can be characterized by frequency and orientation bandwidths. Studies based on psychovisual methods have found that the eye has a radial frequency selectivity that is symmetric on a log axis with bandwidths of one octave [6]. Further, the orientation selectivity is symmetric about a center peak angle and bandwidths varying as a function of radial frequency from 30° for high frequencies to 60° for low frequencies [11]. These effects can be modeled using Gabor and Wavelet approaches [16]. In this thesis the frequency sensitivity is modeled using a hierarchy of filters. The hierarchy of filters have properties of multiresolution decomposition with localization in frequency and space. The Cortex Transform [19] was selected to implement the frequency selectivity over Wavelet and Gabor transforms. The cortex transform introduced by Watson, is a non-orthogonal transform which is reversible. One of the disadvantages of the the cortex transform is high computational complexity. The radial frequency selectivity and the orientation selectivity are modeled

with separate classes of filters and are cascaded. The combined filter is referred to as the cortex filter. The implementation here is different from the original cortex transform in certain aspects, and they are

- Filter transitions use Hanning function instead of a Gaussian distribution
- Gaussian baseband characteristics to reduce ringing
- All radial frequencies above 0.5 cy/pixel (high frequency residual) are discarded

The radial frequency selectivity is represented by dom filters (difference of low-pass mesa filters). The mesa filters are characterized by a flat passband, a transition region, and a flat stop-band region. The transition region is modeled with hanning window and the mesa filter is described by its half-amplitude frequency $\rho_{\frac{1}{2}}$ and transition width tw. The mesa filter is described below [14]

$$\begin{aligned} mesa(\rho) &= 1.0 && for \quad \rho < \rho_{\frac{1}{2}} - \frac{tw}{2} \\ &= \frac{1}{2}(1 + cos(\frac{\pi(\rho - \rho_{\frac{1}{2}} + \frac{tw}{2})}{tw})) && for \quad \rho_{\frac{1}{2}} - \frac{tw/2}{<}\rho < \rho_{\frac{1}{2}} + \frac{tw}{2} \\ &= 0.0 && for \quad \rho < \rho_{\frac{1}{2}} + \frac{tw}{2} \end{aligned} \tag{3.8}$$

The dom filters shown in Figure 3.2 are formed as the difference of two mesa filters with different half amplitude frequencies. The kth dom filter is given by [14]

$$\begin{aligned} dom_k(\rho) &= mesa(\rho)|_{\rho_{1/2}=2^{-(k-1)}} - mesa(\rho)|_{\rho_{1/2}=2^{-k}} \quad for\ k = 1, K-2 \\ &= mesa(\rho)|_{\rho_{1/2}=2^{-(k-1)}} - base(\rho)|_{1/2} = 2^{-k} \ for\ k = K-1 \end{aligned} \tag{3.9}$$

where the | symbol means the mesa filter is to be calculated with the indicated half-amplitude frequency $\rho_{1/2}$. The lowest frequency filter is referred to as the baseband. A truncated Gaussian function is used to avoid ringing. The baseband function is given by

$$\begin{aligned} base(\rho) &= e^{-\frac{\rho^2}{2\sigma^2}} \ for\ \rho < \rho_{1/2} + \frac{tw}{2} \\ &= 0 \ for\ \rho \geq \rho_{1/2} + \frac{tw}{2} \end{aligned} \tag{3.10}$$

where $\sigma = \frac{1}{3}(\rho_{1/2} + \frac{tw}{2})$; $\rho_{1/2} = 2^{-K}$. K is the total number of radial filters. The frequency is expressed in cy/pixel. Higher values of k correspond to higher level of the hierarchical pyramid and lower frequency bands. The transition width of each filter is a function of its half-amplitude frequency

$$tw = \frac{2}{3}\rho_{\frac{1}{2}} \tag{3.11}$$

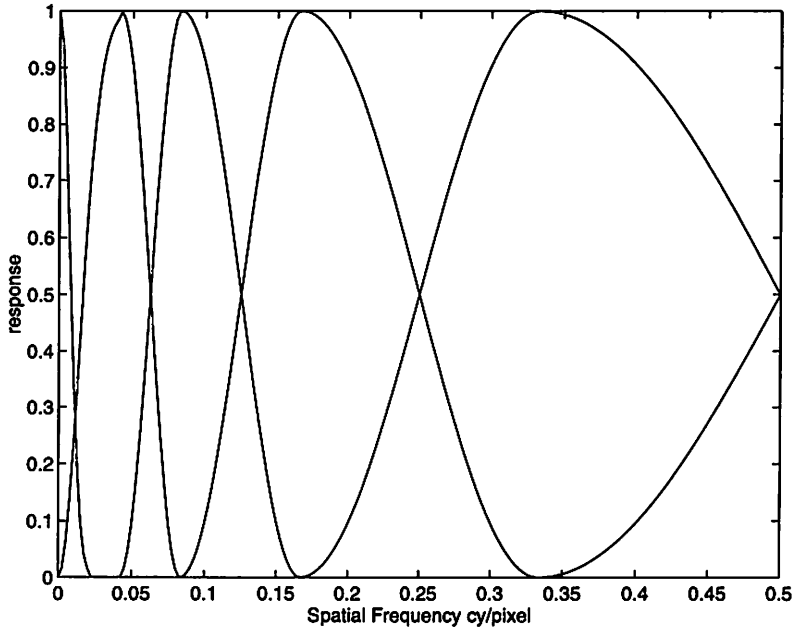

Figure 3.2 Domain filters

The orientation selectivity is modeled with fan filters as shown in Figure 3.3. An integer number of fan filters are used to approximate the nearly continuous orientation sensitivity of the visual system. Hanning window is used to model fan filters. The equation for fan_l as a function of orientation is [14]

$$\begin{aligned} fan_l(\theta) &= \frac{1}{2}(1 + \cos(\frac{\pi|\theta - \theta_c(l)|}{\theta_{tw}})) && for \quad |\theta - \theta_c(l)| \leq \theta_{tw} \\ &= 0.0 && for \quad |\theta - \theta_c(l)| > \theta_{tw} \end{aligned} \tag{3.12}$$

where ,

θ_{tw} = Angular transition width

$\theta_c(l)$ = Peak of the fan filter l

$\theta_{tw} = \frac{180}{L}$

$\theta_c(l) = (l-1)\theta_{tw} - 90$

L= Total number of fan filters

In the current implementation K=6 and L=6 and the orientation bandwidth is 30 degrees. Since the orientation bandwidth is 40° we use L = 6.

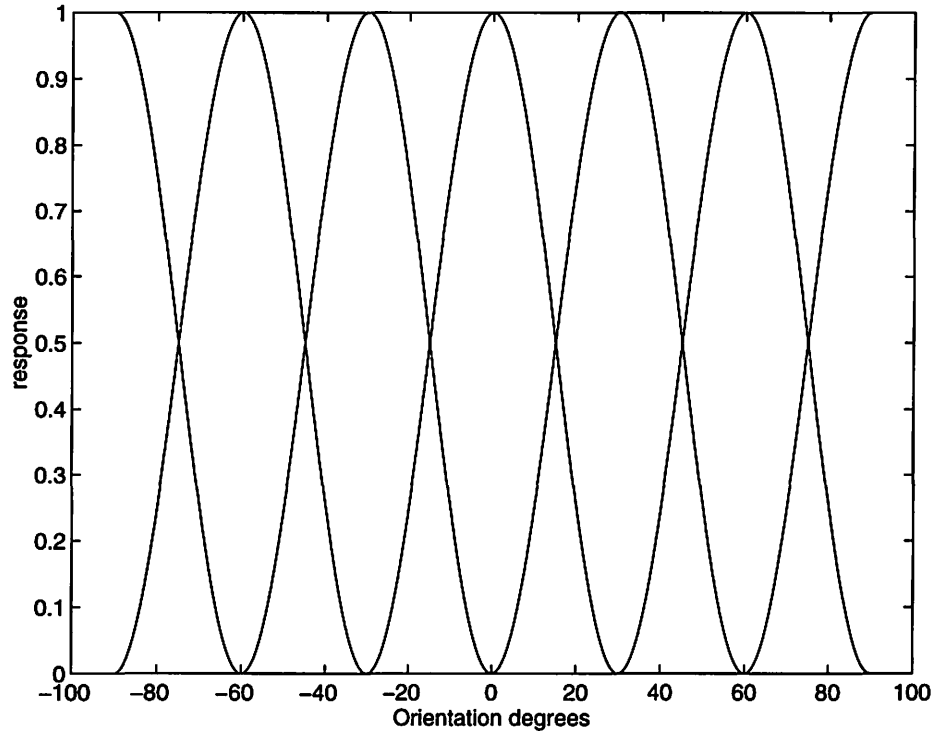

Figure 3.3 Fan filters

The cortex filters are formed as the product of dom and fan filters as,

$$\begin{aligned} cortex_{k,l}(\rho,\theta) &= dom_k(\rho)\, fan_l(\theta) \quad for \;\; k = 1, K-1; l = 1, L \\ &= base(\rho) \qquad\qquad\qquad\qquad for \;\; k = K \end{aligned} \tag{3.13}$$

where the indices k and l are for the dom and fan filters respectively. No orientation selectivity is necessary for the baseband. The total number of cortex filters is $((k-1)*L+1)$, which is 31 for K=6 and L=6. Cortex filters are reversible as shown below

$$\sum_{k=1,Kl=1,L} cortex_{k,l}(\rho,\theta) = 1 \quad \forall \rho,\theta \tag{3.14}$$

3.2.3.2 Contrast units in the cortex filtered images

The two images whose visual differences is to be estimated are filtered by the cortex filter after modification by the amplitude nonlinearity and CSF. The contrast in the cortex bands is given by

$$C_{k,l}(i,j) = (B_{k,l}(i,j) - \overline{B}_{k,l})/\overline{B}_{k,l} \tag{3.15}$$

where $C_{k,l}(i,j)$ is the contrast in band (k,l), $B_{k,l}(i,j)$ is the filtered image and $\overline{B}_{k,l}(i,j)$ is the mean of band (k,l). All of the mean filtered images except the baseband mean $\overline{B}_{k,l}(i,j)$ is zero, making the above equation indeterminate. To overcome this we use we use the baseband mean $\overline{B}_K$ as the denominator and leave numerator mean as zero, yielding,

$$C_{k,l}(i,j) = B_{k,l}(i,j)/\overline{B}_K \tag{3.16}$$

3.2.3.3 Masking function

Masking is the decrease in visibility due to the presence of a suprathreshold background. The masking function quantifies this as a function of background contrast. It is modeled using a threshold elevation image. Let us consider a image (the background) has a contrast of c; the other (the background plus the increment) has a contrast of $c + \Delta\ c$. The increment $\Delta\ c$ is varied to find the threshold at which $c + \Delta\ c$ can just be discriminated from c. When the background contrast is zero ($c = 0$), the measured contrast-detection threshold is C. As the contrast of the background increases the threshold increases and this is called the threshold elevation. The threshold elevation image

gives the threshold used in the psychometric function. The masking function describes the threshold of a test signal as a function of the contrast of the mask (background). It is plotted on a log-log scale, the horizontal axis is the contrast of the mask, and the vertical axis is the threshold of the signal in the presence of the mask. The plot shows two asymptotic regions, one whose slope is zero near low contrast and a positive slope which shows that threshold increases as the mask contrast increases. The zero slope occurs for masks of low contrast, indicating there is no masking effect, near the uniform field. This uniform field threshold is a function of frequency and is equal to the inverse of the CSF. The masking curve described in the Figure 3.4 occurs for all signal and mask frequency combinations, the location of the asymptotes depend on the frequencies of the test and mask signals. Both the CSF and the spatial frequency selectivity of the visual system play important part in the modeling of masking.

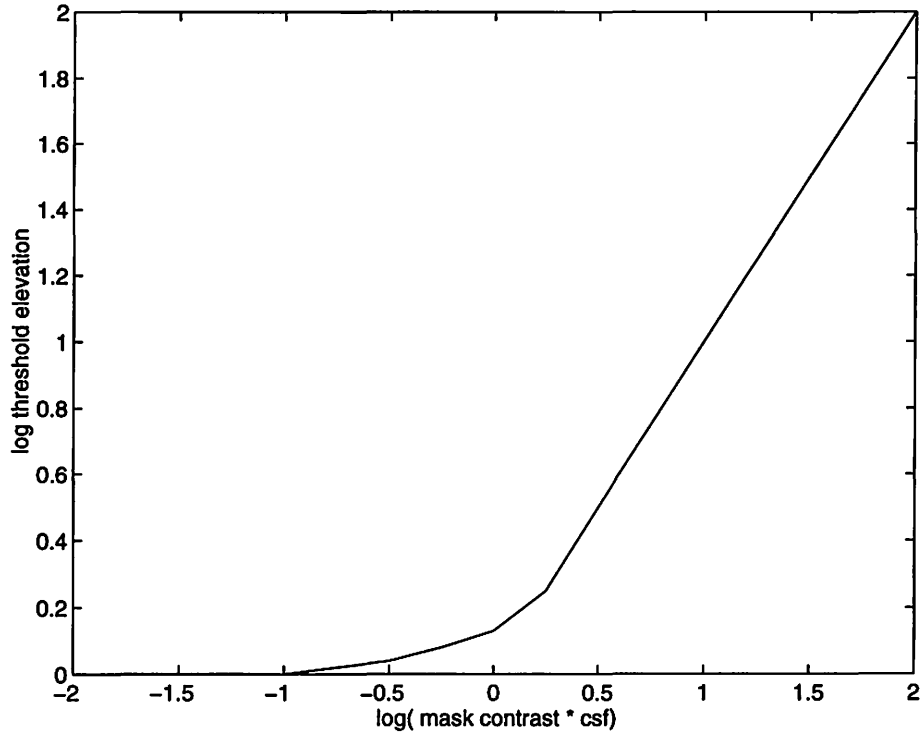

Figure 3.4 Masking function normalized by inverse of CSF on both axes

Let us consider the simple case of where the test and the mask signal frequency is

the same. We normalize both the test threshold and mask contrast axes by the test frequency's threshold ($\frac{1}{csf}$) , the curve in Figure 3.4 can describe all frequencies. This normalization removes the CSF, and the vertical axis is labeled threshold elevation T_e,

$$T_e(\rho, m) = \frac{T(\rho, m)}{T(\rho, 0)} = T(\rho, m).csf(\rho) \tag{3.17}$$

which describes masking as a functions of frequency ρ and mask contrast m. If we change the horizontal axis by normalizing for the filter shape of the visual systems responsible for detecting the test frequency F, the curve in Figure 3.4 represents threshold elevation. The mask contrast axis now represents contrast as seen by the detecting mechanism. Since the cortex filter is used to model the detection mechanism's frequency response, the cortex filter normalizes all frequencies in the band to their effective contrast as seen by the detection mechanisms. After filtering the images into separate bands, we need not worry about individual frequencies in the band, because all frequencies in the band have been normalized by the CSF and cortex filter. So a single curve can be used to describe masking in the band. The normalized mask contrast term,

$$m_n(\rho_t, \theta_t) = m(\rho_m, \theta_m).csf(\rho_m, \theta_m).cortex_{k.l}(\rho_m, \theta_m) \tag{3.18}$$

which describes the mask contrast at frequency (ρ_n, θ_m) seen by the mechanism that detects the test signal at frequency (ρ_t, θ_t). The indices k, l on the cortex filter correspond to the cortex filter centered on the test signal frequency.

In the following paragraphs the concept of phase coherent and phase-incoherent masking is discussed. In phase coherent masking sine waves were used for test and mask frequencies. The special case where the test and mask frequency is the same is known as contrast discrimination, because the detection task is simply to discriminate differences in contrast between the signal and the mask contrast. In this case the threshold drops below the uniform field threshold near the intersection of the asymptotes and is known as the dipper effect. The phase coherent masking results in a high mask con-

trast asymptote with a slope of 0.7 (s).The masking function is modeled using power laws as a function of the normalized mask contrast m_n,

$$T_e(m_n) = \frac{T(m_n)}{T(0)} = [\frac{m_n}{T(0)}]^s \; for \; m_n \; > \; T(0) \tag{3.19}$$

$$T_e(0) = 1.0 \; for \; m_n \; \leq \; T(0) \tag{3.20}$$

In these equations the frequency term is omitted because they are assumed to be equivalent.

For phase incoherent masking the high contrast asymptote has a slope of 1 indicating a Weber-law behaviour. The facilitation effect does not occur for noise masks. The results of the noise masks and test signals are fit with the following equation

$$T = W\sqrt{n_i^2 + m^2} \tag{3.21}$$

where T is the threshold, m is the mask contrast, W is the detection signal-to-noise ratio and n_i^2 is proportional to the internal noise of the visual mechanism responsible for detecting the signal. The parameters W and n_i can be determined from the asymptotes of the experimental data as

$$W = \frac{T(m_n)}{m_n} \; when \; n_i \; \ll \; m_n \; (high - contrast \; asymptote) \tag{3.22}$$

$$n_i = \frac{T(0)}{W} \; when \; m_n = 0 \; (zero - slopeasymptote) \tag{3.23}$$

We can rewrite equation 3.21 in terms of threshold elevation,

$$T_e(m_n) = \sqrt{1 + (Wm_n)^2} \tag{3.24}$$

In the VDP application the image content is regarded as noise, whereas the distortion is regarded as the signal whose detectability we wish to predict. The difference in the high contrast asymptote slope between the phase coherent and phase incoherent case is

due to the uncertainty caused by the masking pattern. This uncertainty can be reduced by learning. Since the sine masking pattern is easy to learn as the eye just needs to look for differences in the expected appearance. The noise pattern is difficult to learn, which results in higher slope that is indicative of more masking. The masking function including the learning effect is given by,

$$T_e(m_n) = (1 + (k_1(k_2 \,.\, m_n)^s)^b)^{1/b} \tag{3.25}$$

where T_e is the threshold elevation, and s is the slope of the high contrast asymptote, which ranges between 0.6 and 1.0. The parameters k_1 and k_2 determine the pivot of all the learning slopes. Parameter b determines how closely the curve follows the asymptote in the transition region and varies for the different bands. The parameters k_1, k_2 are related to W by

$$k_1 = W^{(1/(1-Q))} \tag{3.26}$$

$$k_2 = W^{(1/(1-Q))} \tag{3.27}$$

where Q is the slope of the high-contrast asymptote when the intersection of the two asymptotes is 1 on the normalized mask contrast axis.

If the threshold and the mask contrast are normalized by the uniform field threshold ($\frac{1}{csf}$), then a single curve describes the masking for all frequencies. The masking function described above is for the visual systems spatial frequency selectivity rather than for the discrete approximation of the cortex filter. There is not a detector centered at every frequency. In the cortex filter the test frequency includes all frequencies in that particular cortex band. We need to broaden test frequency to include all frequencies in the cortex band. Similarly the mask frequency consists of all frequencies in the cortex band. Many frequencies contribute to the normalized mask contrast m_n within a particular band (k,l) as a function of pixel location (i,j). It is given by,

$$m_n^{k,l}(i,j) = F^{-1}(R(u,v) \cdot csf(u,v) \cdot cortex^{k,l}(u,v)) \tag{3.28}$$

where R is the Fourier transform of the input image after modification by amplitude nonlinearity. Only the magnitude of contrast affects the masking function. The threshold elevation image $T_e^{k,l}$ as a function of the pixel location is given by

$$T_e^{k,l}(i,j) = (1 + (k_1(k_2|m_n^{k,l}(i,j)|)^s)^b)^{\frac{1}{b}} \tag{3.29}$$

The band-specific threshold elevation $T_e^{k,l}(i,j)$ is referred to as the threshold elevation image. The masking image cannot be determined from the original image or the compressed image alone. The solution is to form the masking image from both the original and the compressed images, using the minimum of the two threshold elevations. This technique is called mutual masking since this is the only masking that is common to both the images. The the masking threshold is given by

$$T_{em}^{k,l}(i,j) = min[T_{e1}^{k,l}(i,j) \quad T_{e2}^{k,l}(i,j)] \tag{3.30}$$

where the subscripts 1 and 2 refer to the two images input to the VDP.

3.2.3.4 Psychometric function

The Psychometric function describes the increase in the probability of detection as a function of the contrast. It is given by

$$P(c) = 1 - e^{-}(c/\alpha)^{\beta} \tag{3.31}$$

where, P(c) is the probability of detection of a signal of contrast c, α is the threshold and β is the slope of the psychometric function. In the VDP algorithm the probability of detection for each of the cortex bands is given by

$$P_{k,l}(i,j) = 1 - e^{-(\Delta C_{k,l}(i,j)/T_e^{k,l}(i,j))^{\beta}} \tag{3.32}$$

The contrast difference for band k, l as function of pixel location (i,j) is given by the following equation

$$\Delta C_{k,l}(i,j) = \frac{B1_{k,l}(i,j)}{\overline{B}_K} - \frac{B2_{k,l}(i,j)}{\overline{B}_K} \tag{3.33}$$

where $B1_{k,l}$ and $B2_{k,l}$ are the two filtered images for band k, l.

3.2.3.5 Probability summation

After the calculation of the individual probability images for the different cortex bands they need to be combined to obtain a single image. Probability summation is used, with the the equation given by the product series

$$P_t(i,j) = 1 - \Pi \ (1 - P_{k,l}(i,j)) \tag{3.34}$$

where Π denotes the product and P_t is the total probability of detection resulting from all bands. The most common method used to combine the outputs of the different cortex bands is vector summation into a single number. This approach treats the whole image as a single signal and does not show which areas of the image are above and which parts are below the threshold. The method of probability summation given by the above equation is more appropriate for threshold predictions and hence relevant to image fidelity assessment.

3.2.4 Display and interpretation of the visual difference image

The output of the VDP is a probability image of detecting visible differences. The magnitude of the visible difference image ranges from 0 to 1. Considering the polarity of the differences, the probability map extends from -1 to 1. If in the compressed image the gray level is less than the original image and the error is visible then we have a probability of -1. The following methods are used for displaying the detection image

- Free-Field difference map
- In-context difference map

3.2.4.1 Free-field difference map

The free-field difference map, displays the predictions on a uniform field of gray levels. This method uses a linear mapping of the probability values of the detection map given by

$$FF(i,j) = SP_t(i,j) * \frac{(max - min)}{2} + \frac{(max + min)}{2} \tag{3.35}$$

where FF is the free-field difference prediction, SP is the signed probability of detection, max and min are the display device maximum and minimum values. The free-field difference map is a gray scale image. Pixels that are lighter than (max+min)/2 indicate where the distorted image is lighter than reference. Pixels that are full white or black are detected completely (the visual difference between two images at these pixels locations are visible to the eye).

3.2.4.2 In-context difference map

With the free-field difference map it is difficult to judge the relation between the predicted differences and the actually observed differences between the two input images. In-context difference map overcomes this problem by displaying the predicted differences in context of the reference image. The reference image is copied to the RGB planes, and the detection image is scaled and added to one of the color planes as given below

$$IC(i,j) = SP_t(i,j)\frac{(max + min)}{2} + ref(i,j) \tag{3.36}$$

where ref is the reference image or in this case the uncompressed image. If the red layer is used then cyan indicates where the distorted image appears darker than the reference.

3.2.4.3 Interpretation of maps

The probability map indicates the magnitude and location of the predicted visual differences. The output detection maps can be used to see the nature of the distortion

introduced by the different compression methods. The shapes and locations of the predicted differences can be used to improve or choose the appropriate compression routines and parameters. The detection map can be reduced to a single number and used to determine if two images are visually identical. If the peak magnitude of the detection map is less than 0.5, the images are visually equivalent. To compare two different compression methods we perform the VDP for a number of viewing distances and determine the minimum distance where the two images are visually equivalent. This is referred to as the critical distance. The compression method with the lower critical distance is better for the image under consideration.

3.3 Results

The original image 'sub' is shown in Figure 3.5 and the compressed image using fractal compression is shown in Figure 3.6. The visual difference predicted image is shown in Figure 3.10 and it evident that the VDP image agrees well for high frequencies and is deficient at low frequencies as can be seen from the low gray scale values in the left portion of the image. Figure 3.9 shows the intensity values that are used for the gray levels when calculating the VDP image. The intensity of the gray levels was obtained by measuring the intensity of uniform gray level images using a exposure meter which calculates the luminous intensity. Figure 3.12 shows the visual difference predicted image for the JPEG image and it agrees well with the visual evaluation. The JPEG compression introduces the least amount of visual distortion. The VDP image has high detectability values for the flaw and this might be due to the approximate intensity values which have been used. From the figures presented it is seen that the VDP overestimates the predictability of difference at the low gray values. The parameters used to calculate the VDP are W=6, Q=0.7, s=0.7 and b=4. Figure 3.16 shows the VDP image for the upper right portion (128x128) of Figure 5.4 for the original image 'post12'. It is observed from

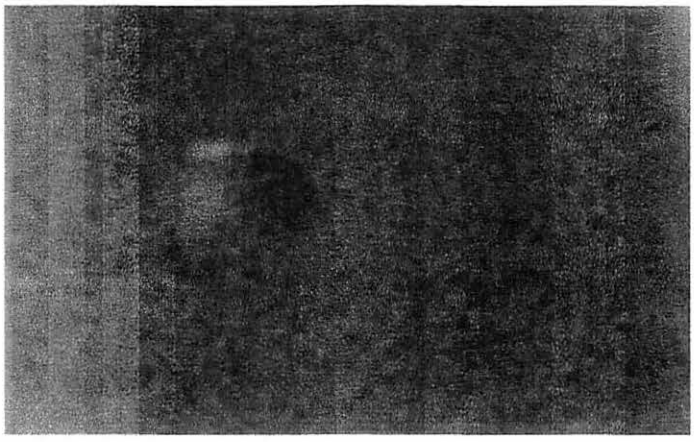

Figure 3.5 Original image 'sub'

Figure 3.17 that the fractal image compression introduces artifacts along the partitions which are more annoying due to the discontinuity across the partitions. Image smoothing might help reduce the VDP values. The performance of fractal compression method is better relative to JPEG in terms of compression ratio but JPEG introduces errors which are less visible when compared to fractal compressed images. Figure 3.8 shows the thresholded VDP image for vector quantization and it predicts the error along the border on the flaw reasonably well but is deficient in the inner regions of the flaw. One of the areas which need to be further explored is how the values k_1, k_2 and s change with the band values (k,l). In this thesis we have used constant values for all the bands. The high frequency errors are well predicted by the VDP. Consequently the performance in the lower frequency bands need to be investigated to estimate values of k_1, k_2 and s that are required to obtain better VDP values.

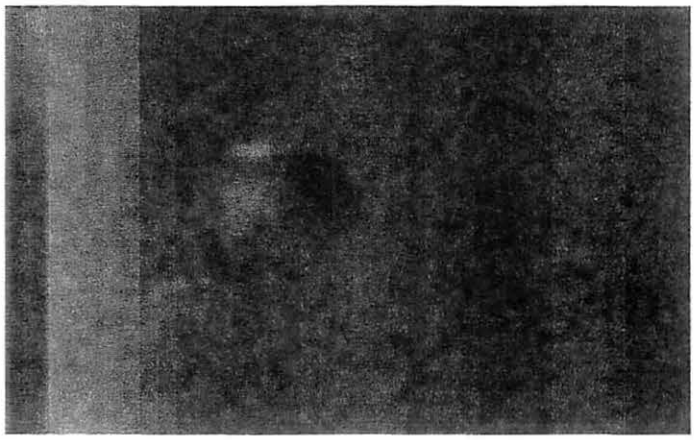

Figure 3.6 Compressed image using fractal compression

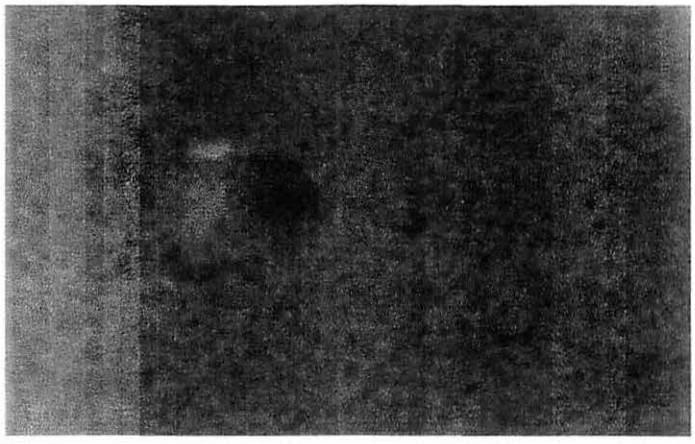

Figure 3.7 Compressed image for 'sub' using JPEG with Q=0.7

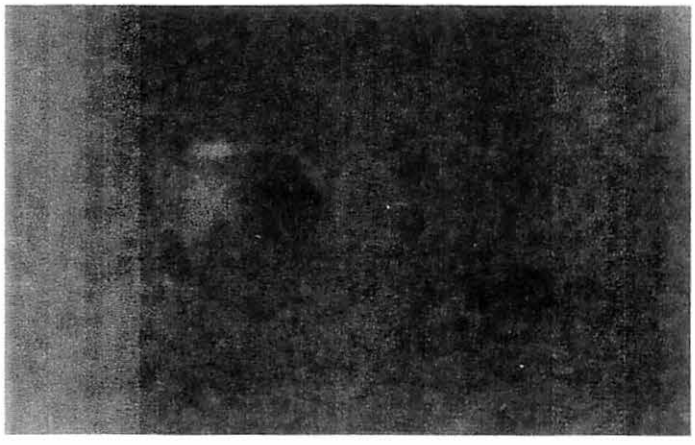

Figure 3.8 Compressed image using vector quantization

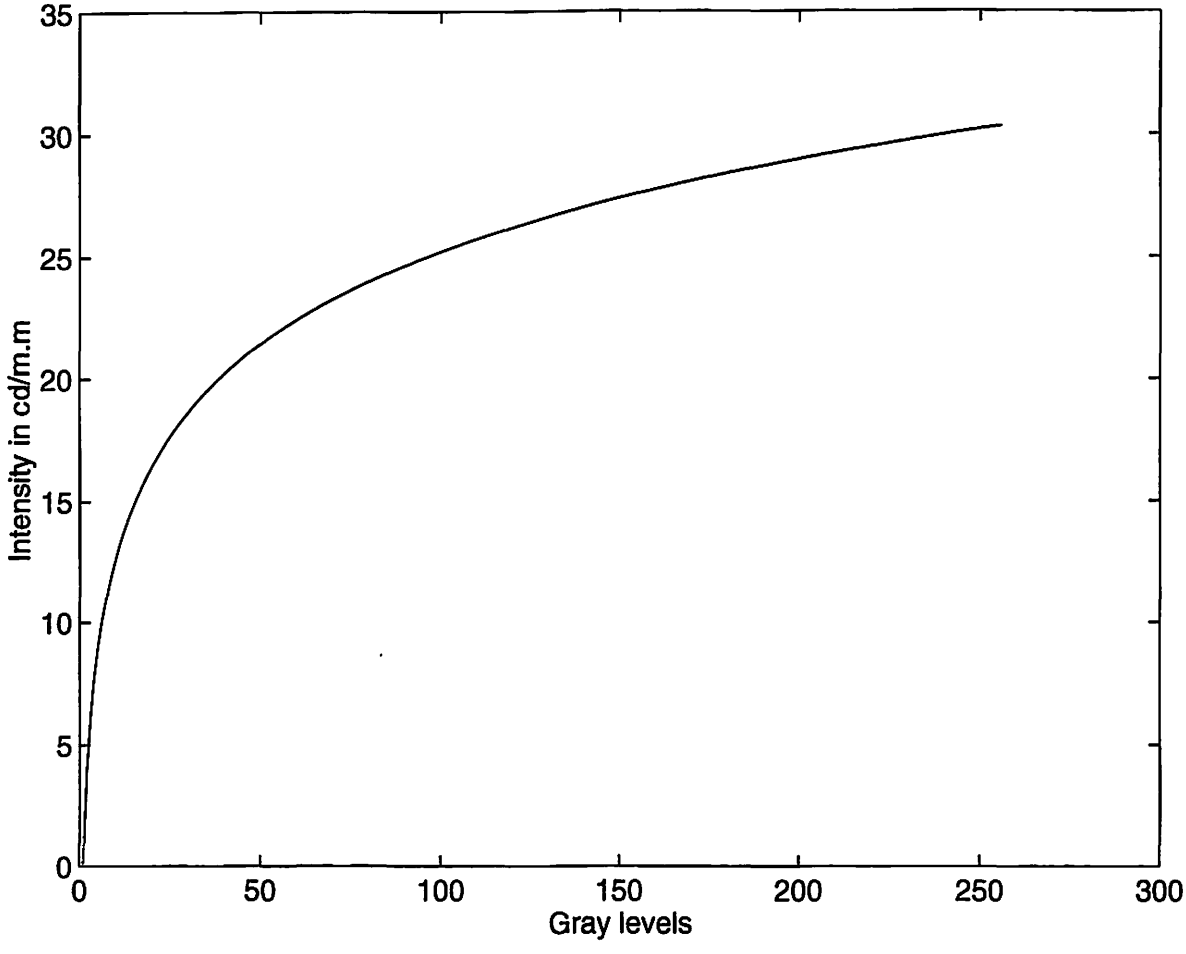

Figure 3.9 Relationship between gray scale and intensity

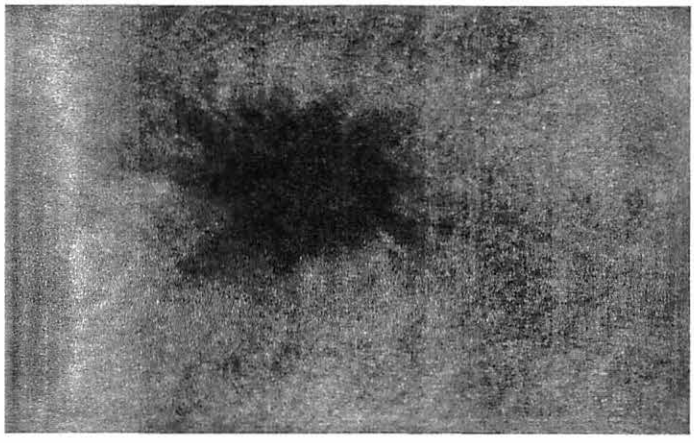

Figure 3.10 Visual difference predicted image for fractal compressed image

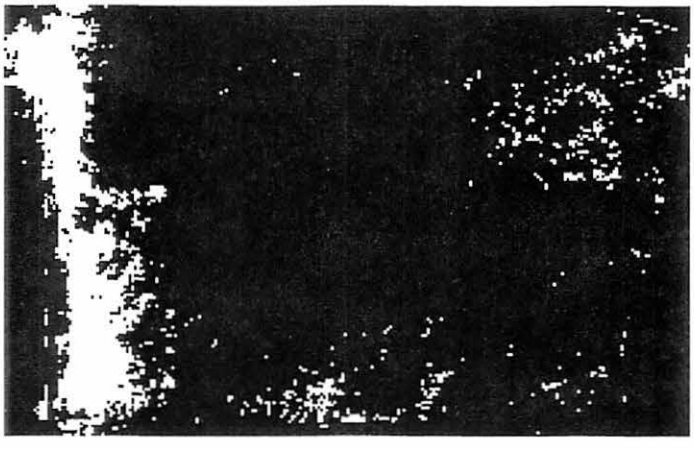

Figure 3.11 Visual difference predicted image thresholded at 0.5

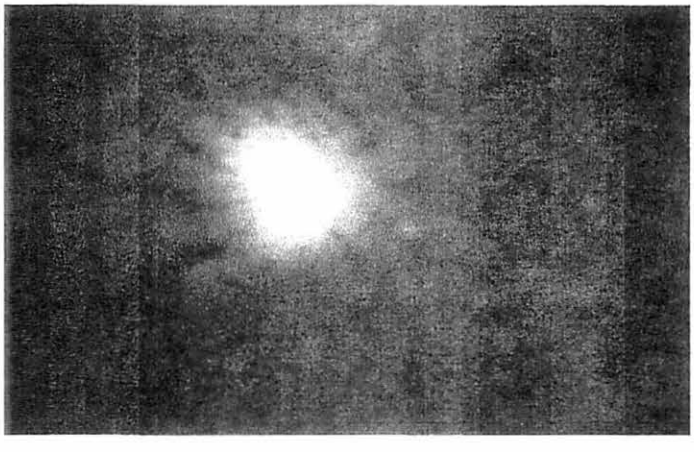

Figure 3.12 Visual difference predicted image for JPEG image with Q=0.7

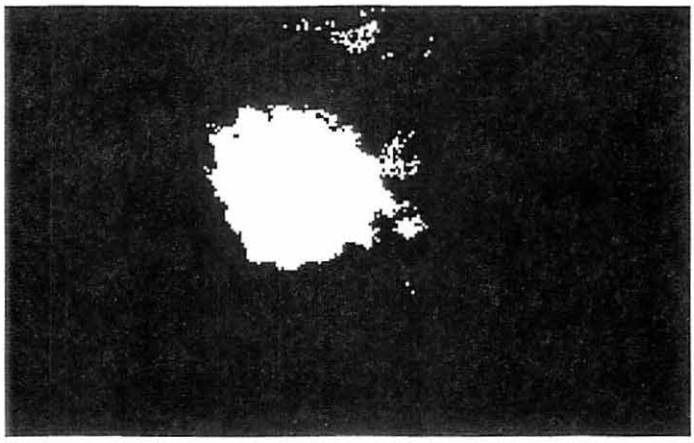

Figure 3.13 Visual difference predicted image thresholded at 0.5 for JPEG with Q=0.7

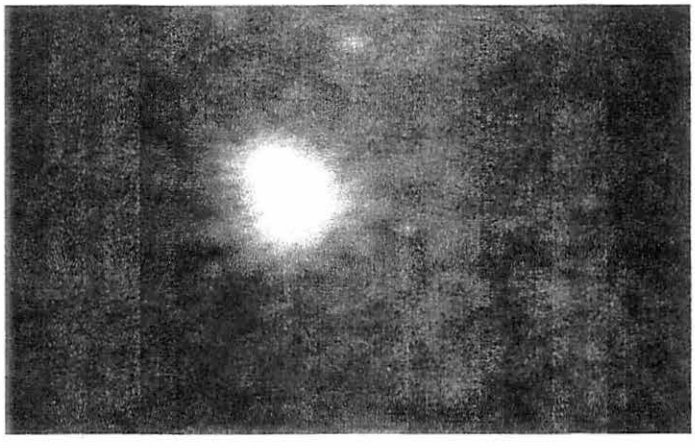

Figure 3.14 Visual difference predicted image for vector quantized image with $T_e = 0.15$ *and* $T_l = 2$

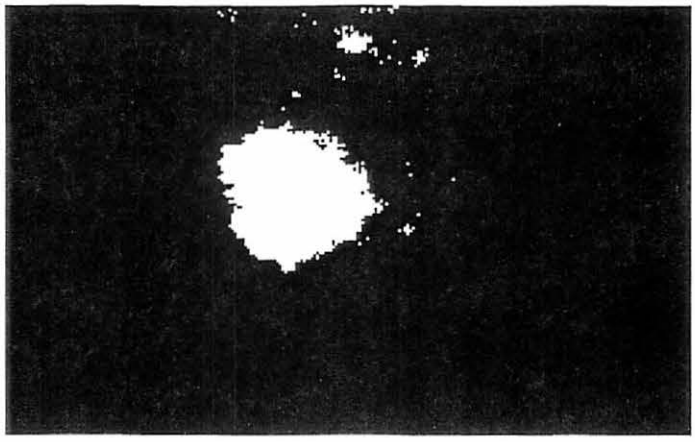

Figure 3.15 Visual difference predicted image thresholded at 0.5 for vector quantization

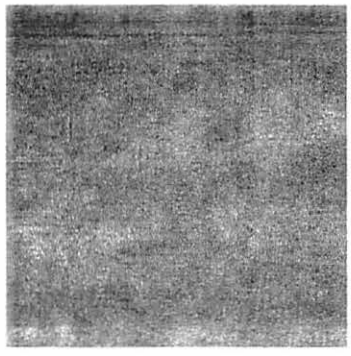

Figure 3.16 Visual difference predicted image for fractal compressed image of 'post12'

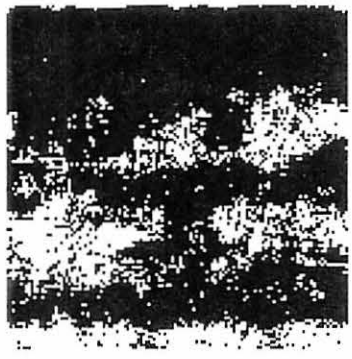

Figure 3.17 Visual difference predicted image thresholded at 0.5 for 'post12'

4 APPLICATION OF DETECTABILITY MODEL TO IMAGES

4.1 Detectability measures

The VDP measure presented in the previous chapter predicts the visual difference between two images at every pixel. The probability of detection (pod) presented in this chapter deals with the detection of a signal in the presence of noise. The method which produces the radiograph also introduces noise and this method determines whether in a small portion of the image a signal is present. Once the radiograph has been digitized, we can apply statistical techniques based on the known noise characteristics of the measurement process to determine the statistical significance of the flaw. The procedure optimally thresholds a digitized radiograph within regions of stationary data. The image data within a stationary region is assumed to be a mathematical combination of the flaw signal and noise. The threshold is selected based on a comparison of the statistics of the candidate region with a region whose statistics are known to be caused by noise alone. Consider a region within the image having the same average background but does not include a flaw signal. We estimate the probability density functions of the candidate region and flaw-free region, p_{n+s} and p_n by their normalized histograms. The optimal threshold is selected by maximizing [18]

$$Q = \sum_{g=t}^{M} p_{s+n}(g) - \sum_{g=t}^{M} p_n(g) \tag{4.1}$$

over all gray levels g, where M is the maximum gray scale value and t is the threshold.

The distribution of thresholded pixels within the image window is dictated by the noise distribution $p_n(g)$. If we accept the hypothesis, we assume no flaw is present, if we reject it, we assume a flaw is present. Under the assumption of the above hypothesis, each thresholding operation is a Bernoulli trial with a probability of success defined by

$$p_w = \sum_{g=t}^{M} p_n(g) \tag{4.2}$$

The resulting number of pixels within the window follows a binomial distribution and is represented as:

$P(n_w) = \binom{A}{n_w} p_w^{n_w}(1-p_w)^{A-n_w}$

where, n_w is the number of thresholded pixels and A is the total number of pixels. The hypothesis can be tested as

$$H_0 : \frac{n_w}{A} = P_w H_A : \frac{N_w}{A} > p_w \tag{4.3}$$

For a large sample, the binomial distribution can be approximated by a normal distribution having mean p_w and variance $p_w(1-P_w)/A$, which yields the following test statistic.

$$z_T = \frac{\frac{n_w}{A} - P_w}{\sqrt{(p_w(1-p_w)/A)}} \tag{4.4}$$

The computed value of z_T is compared against z_α and the decision rule is:

Accept H_0 if $z_T \leq z_\alpha$ (no flaw)

Reject H_0 if $z_T > z_\alpha$ (flaw)

The window size should be comparable to that of the expected flaw size.

4.2 CRT colorimetry

This section describes how the intensity values on the cathode ray tube (CRT) can be calculated from the digital gray values of the image [3]. The presentation in this

section is necessary because all digitized radiographs are viewed on a CRT and the characteristics of the CRT influence the luminance of the pixels. The luminance of the pixels influences the image quality. For example some cathode ray tubes have poor pixel resolution. Here a computer controlled CRT display system is considered. The graphics hardware contains a frame buffer, a lookup table (LUT), and video-signal generation circuitry. The digital information is processed in the LUT. We assume that the LUT does not alter the digital values. The digital values are quantized linearly to voltages in the digital-to-analog (DAC) converter. The model relates DAC values to the properties of phosphor emission. The DAC values d_r are quantized linearly to video voltages v_r:

$$v_r = (v_{max} - v_{min})(\frac{d_r}{2^N - 1}) + v_{min} \tag{4.5}$$

where v_{max} is the maximum video voltage representing white level and v_{min} is the minimum voltage used for defining the black level. The video voltage is modulated between v_{max} and v_{min}. The electrons emitted by the cathode are accelerated towards the the phosphor screen (anode). The control grids modulate the flow of the electrons. The video amplifiers transform the positive video voltages into negative voltages v_G that are applied to the first control grids:

$$v_G = av + b \tag{4.6}$$

where a is the video amplifier gain and b is the video amplifier offset.

The relationship between the electron beam current j modulated by the control grid and the amplified video voltage is nonlinear. As v_G become more negative with respect to the cathode, electrons are repelled, reducing the beam current and resultant phosphor excitation. Below a cutoff voltage v_C, the grid voltage does not affect the beam current. The relationship between beam current j and grid voltage v_G is given by

$$\begin{aligned} j &= (v_G - v_C)^{\gamma_r} \quad v_C \leq v_G \\ &= 0 \quad v_C > v_G \end{aligned} \tag{4.7}$$

Plotting the optical density against the logarithm of exposure results in a sigmoid function; the calculated slope of the straight-line portion of the sigmoid curve is the γ of the particular emission.

4.3 Results

The probability of detection (pod) values are calculated using 30 images. It was found by experimental methods that we need at least 25 images to calculate pod effectively. The pod values can be used in the determination of presence of flaw. The pod values do not reflect the visibility of the flaw. Table 4.1 shows the the probability of detection (pod) values for a flat plate with different flaw sizes and how the compression affects the pod values. The file 'flat0.5mil' is the simulated radiograph (using XRSIM) for a flat plate of thickness 1 inch with a 0.5 mil flaw. The fractal coding parameters are tolerance 1, mimimum quadtree level 4, maximum quadtree level 6. The image was obtained after 20 iterations with no post processing. The false alarm rate is 0.05. A 16x16 window was considered in a 256x256 image. The pod values for fractal image compression are high since in the compressed image both the noise and the signal regions are smoothed out. So the pod increase is not an actual pod increase since the gray levels of the pixels are considerably different from the original image. Vector quantized images have low errors in the individual pixels and hence their pod values are close to the pod values in the original image.

Table 4.1 POD values for different compression routines

Compression Method	flat0.5mil	flat1.25mil	flat1.5mil
Original	0.4692	0.8729	0.9506
Fractal	0.9018	0.9951	0.9951

5 RESULTS

Vector quantization achieves compression ratio of the order of 7. When the edge threshold T_e is increased above 0.15 the differences are visible. JPEG achieves a higher compression when compared to vector quantization for a class of images in which the transform coefficients are better correlated than in the image domain. When the quality factor Q is below 0.7 JPEG breaks down and artifacts are introduced. Fractal image compression ratios are dependent on the images, since the transformations are constructed from the blocks in the images. When the quadtree level is decreased the degradation of the compressed image is evident since the block size increases. The compression is controlled by the quadtree level and the amount of error we are willing to tolerate. One of the advantages of fractal image compression is that the image can be decoded at any desired resolution. It acts like an interpolator for decoding at resolutions higher than the original image size.

The image post12 was obtained using a real time x-ray imaging system. Figure 5.1 through 5.4 represent the decompressed images for post12 for one, two, four and eight iterations respectively of the code. The initial image used was all black. From the variation between the decompressed images between successive iterations, it is observed that it takes at least eight iterations before the decompressed image stabilizes. Figures 5.5 through 5.8 are decompressed images for post12 for one, two, four and eight iterations where the initial image was post9. It can be inferred from Figure 5.1 and Figure 5.8 that after eight iterations the decompressed images do not differ much, even though the starting images are different. The initial image does not affect the uncompressed image,

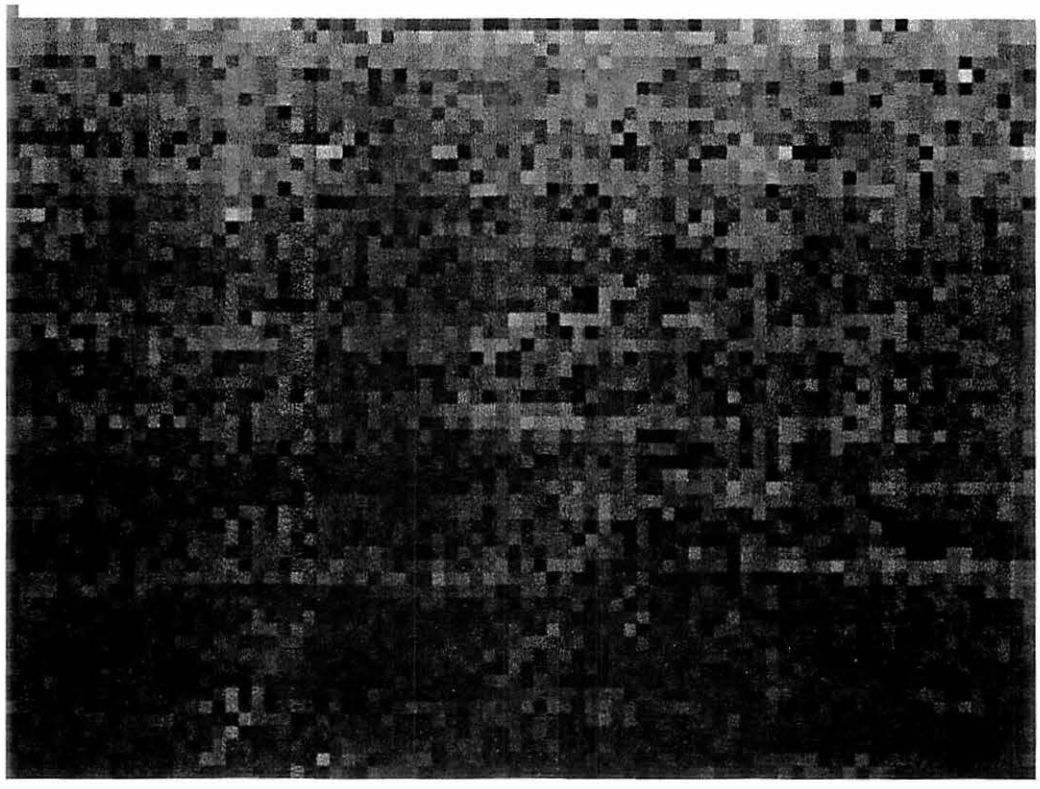

Figure 5.1 Decompressed image for 'post12' after one iteration of the code

it is the combination of all transformations on the blocks that describes the compression.

For images visual difference predictor (VDP) values less than 0.5 can be tolerated. VDP and probability of detection are two different measures which measure image quality.

From the Table 5.1 it can been inferred that the compression ratio achieved by VQ is close to that of JPEG. Fractal coding does better than VQ and JPEG for image like post12 where there is much self similarity among the blocks. Fractal coding achieves good coding of the flat5mil whereas other methods have poor compression ratios.

Future Work

Vector quantization in the transform domain needs to be explored. The classification of edges and shades can be improved. A weighted vector quantization will weight the

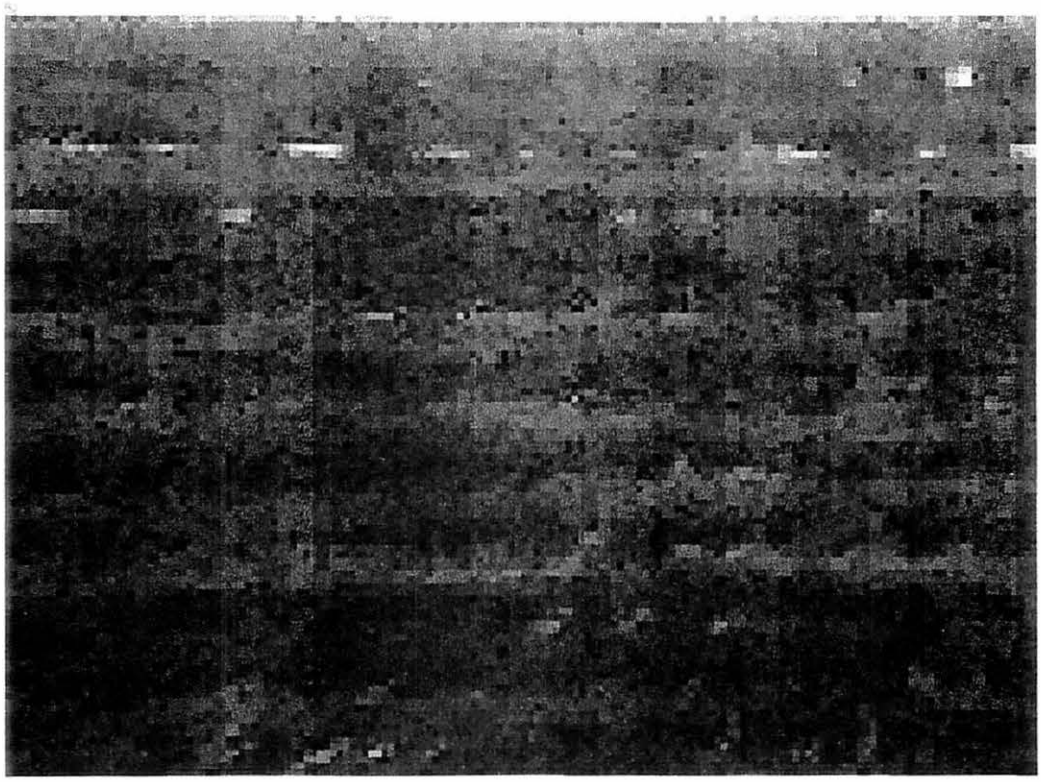

Figure 5.2 Decompressed image for 'post12' after two iterations of the code

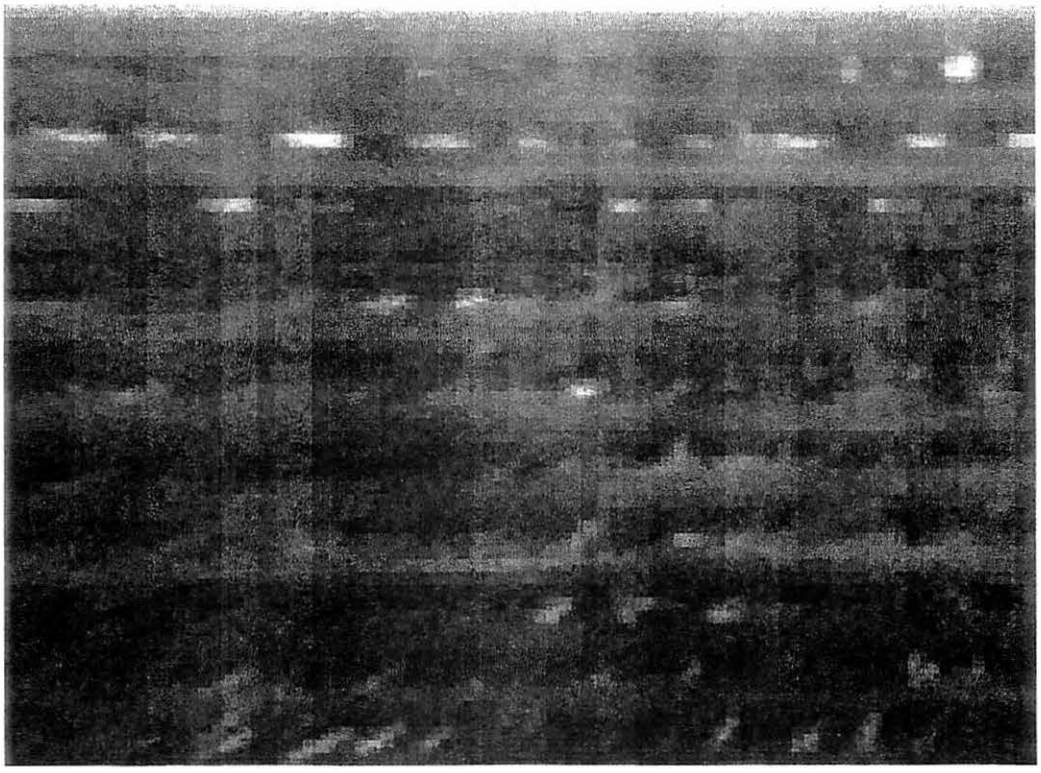

Figure 5.3 Decompressed image for 'post12' after four iterations of the code

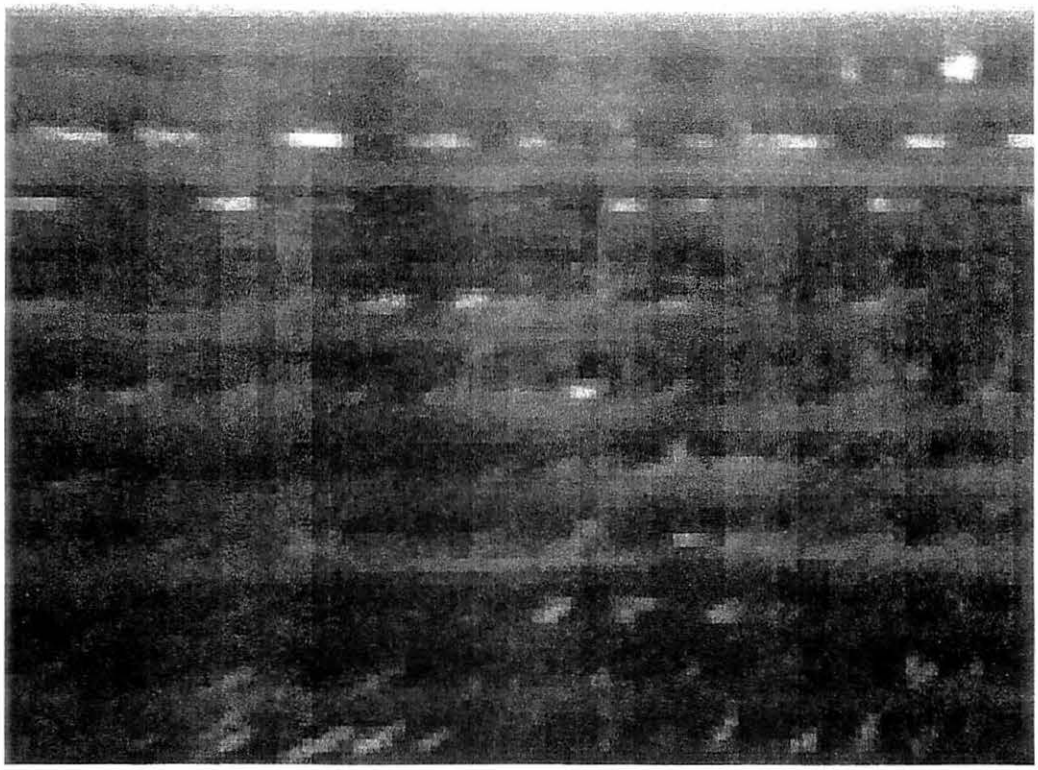

Figure 5.4 Decompressed image for 'post12' after eight iterations of the code

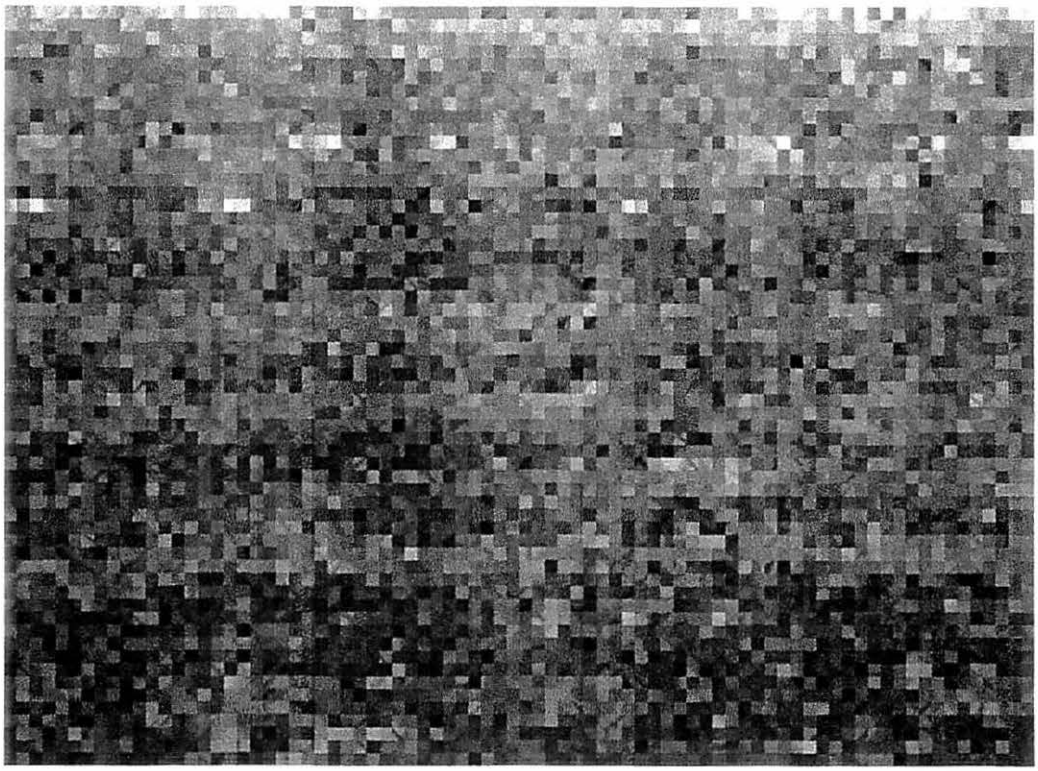

Figure 5.5 Decompressed image for 'post12' after one iteration with 'post9' as the initial image

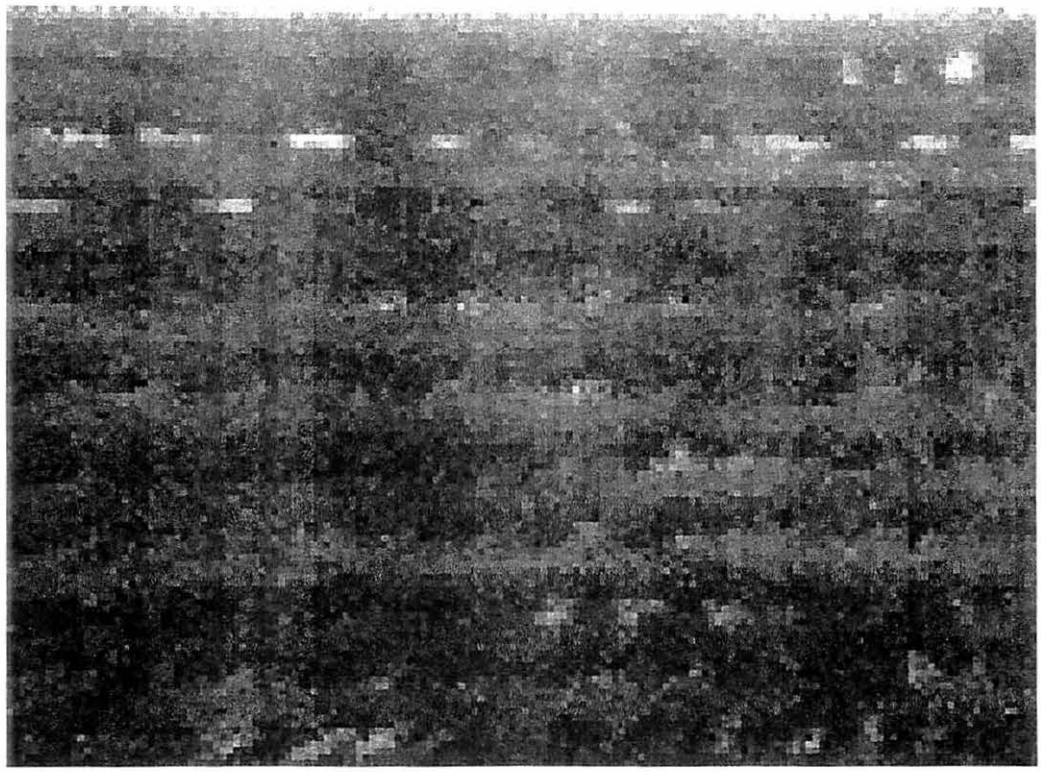

Figure 5.6 Decompressed image for 'post12' after two iterations with 'post9' as the initial image

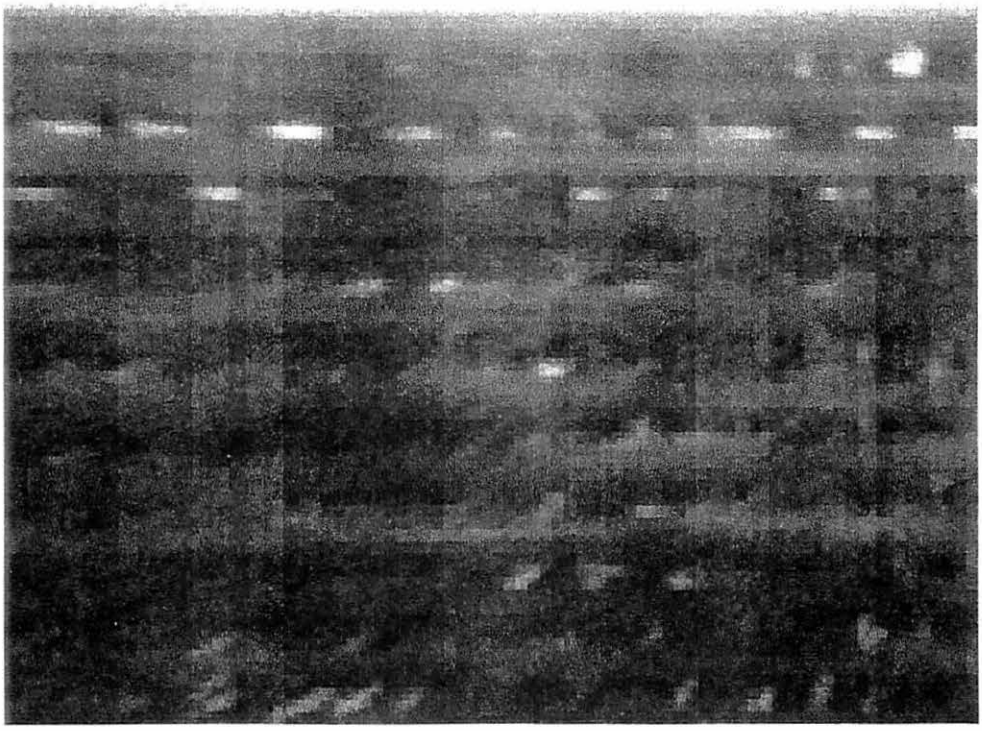

Figure 5.7 Decompressed image for 'post12' after four iterations with 'post9' as the initial image

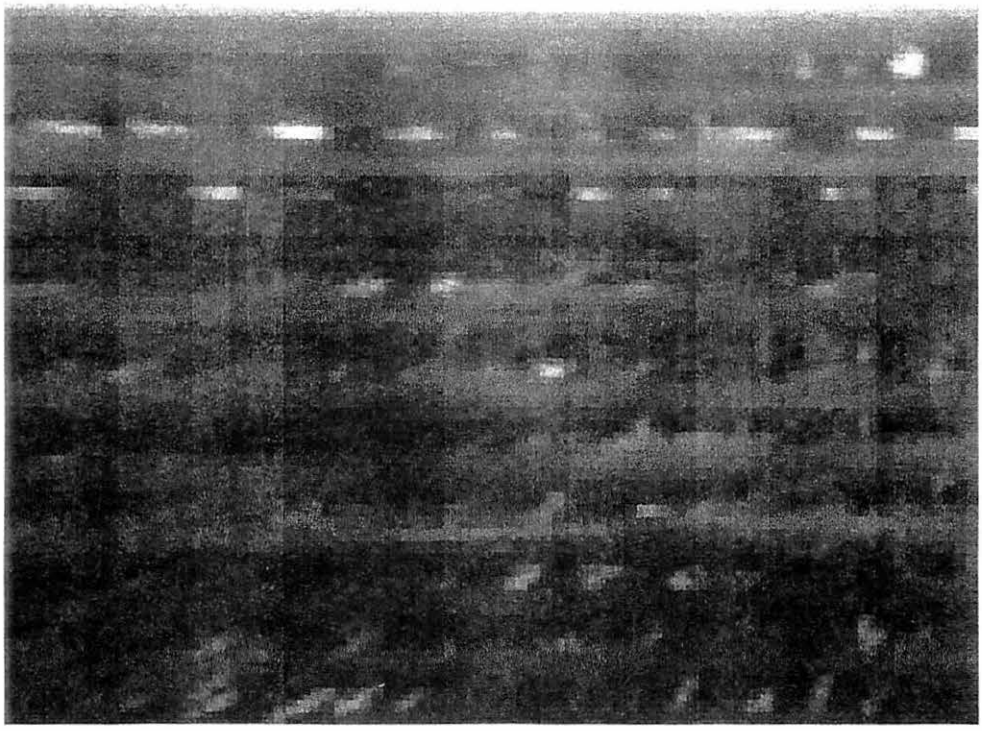

Figure 5.8 Decompressed image for 'post12' after eight iterations with 'post9' as the initial image

Table 5.1 Compression Ratio

Image	Lossless	VQ THL=1 THS=0.1	VQ THL=2 THS=0.15	JPEG Q=0.70	Fractal
autof	1.654	8.100	11.520	11.060	5.281
post9	1.410	7.465	7.717	10.450	4.520
post12	1.015	1.927	1.926	4.067	4.596
flat5mil	1.110	1.613	1.607	2.530	5.258

pixels along the sides of the blocks and it might be able to reduce the inter block discontinuities. In fractal image compression the mean square error is used to optimize the transformations. Instead of the mean square error a weighted error can be used which will minimize the distortions depending on the values in the block. The cortex transform can be used for image compression. The quantizer design while using the cortex transform for compression needs to be explored as the compression algorithm must introduce aritfacts which are not visible.

The cortex transform is one of the approximations used to represent the multiresolution properties of the cortex. Computation of the cortex transform is time consuming. Other efficient techniques to represent the cortex transform have to be explored like Gabor Transform, Wavelet Transform etc. The Gabor Transform is optimal for space-frequency localization. In the VDP method the parameters k_1, k_2 and s need to be estimated for each of the individual bands in the cortex filter. The probability summation to arrive at the final VDP image needs to be investigated further to improve the performance of the algorithm. The orientation and spatial bandwidths have been simplified in the current implementation. Allowing variable orientation and spatial bandwidths for the individual filters in the cortex filter might improve the performance of the visual difference predictor. The relationship between visual difference predictor (VDP) and probability of detection (pod) needs to be explored. This might help the image compression in terms of the error and also in terms of the appearance of the error when the compressed image is viewed by the eye.

Since the cortex filter is a multiresolution filter parallel implementations of it will speed up the the computation of VDP. In the fractal image compression algorithm, further methods to reduce the domain space and efficient computations of the transformations should be developed.

APPENDIX CORTEX TRANSFORM

The cortex transform maps an image into a set of images that vary in resolution and orientation. The cortex transform is used to model human vision.

Mathematical analysis of visual Cells

Each neuron in the cortex is characterized by its receptive field (RF), the region of the retina within which light will cause some response of the cell. If the response is linear it can be characterized by a weighting function. Total response of the cell can be computed by cross correlation of the weighting function and the input image. To model the above effects the images, weighting function, and neural images are expressed in discrete form on a rectangular array.

Cortical cells

The area of the brain that plays an important role in vision is the portion of the visual cortex known as V1. In the V1 area half of the cells are called simple cells, which appear to have approximately linear integration over their receptive fields, their output subjected to thresholding, compression, saturation and half wave rectification. The simple cells have receptive fields characterized by radial spatial frequency and orientation. The cell will respond to frequencies and orientations within a particular range about its center value. The bandwidth is proportional to the center frequency. De valois found that the median spatial frequency bandwidth is 1.4 octaves and median orientation bandwidth is 40°. Each cell receptive field occupies a small area that is much

smaller than the total frequency region to which the group of cells is sensitive. The ideal receptive field is a 2D Gabor function.

Pyramid decomposition

For a number of applications in image processing the image is decomposed into a number of low-pass or band-pass subimages. The subimages are subsampled in proportion to their resolution resulting in a pyramid. Pyramid structure permits the execution of the algorithm in parallel at several scales. It allows coarse-to-fine searches and to match image quantization to frequency dependent contrast discrimination in humans.

Mesa filter

A low-pass filter with unit gain within the pass-band, and Gaussian fall-off beyond some corner frequency, f, which we define as the frequency at which the gain is $\frac{1}{2}$. The filter is given below

$$m_0(u,v) = (\gamma/f)^2 exp[-\pi(r\gamma/f)^2] * \Pi(r/2f) \tag{A.1}$$

where $r = (u^2 + v^2)^{\frac{1}{2}}$, and where Π is a rectangular pulse of unit height and width centered around the origin. The blurring of the edges can be varied by the sharpness parameter γ. Sharpness parameter is proportional to the corner frequency, divided by the sharpness parameter. We would like to pass as much of the images as possible and modest truncation of the Gaussian taper. To balance these a parameter β which is the ratio of the corner frequency f to the Nyquist frequency is introduced. If the image size is N, the corner frequency is

$$f = \beta N/2 \tag{A.2}$$

Mesa filter has a variable control of the corner frequency and the sharpness of the cutoff. New mesa filters at different scales can be generated by shrinking the original by a scale

factor of s. The result is a blurry disk, with a corner frequency a factor of 2 lower and a cutoff twice as sharp. Thus

$$m_k(u,v) = m_0(s^k u, s^k v) \tag{A.3}$$

where s is the scale factor and k is the filter level.

Difference of mesa filter

The difference of mesa (dom) filter is constructed by subtracting the smaller disk m_1 from the larger disk m_0. The goal is to get a filter with approximately gaussian shape and the dom filter has a roughly gaussian shape. Higher values of gamma the deviation from the gaussian is more. By subtracting adjacent pairs of mesa filters dom filters are generated. The high residue consists of the corners that remain after the largest dom filter is removed. Let d be the frequency domain representation of the dom filter and k be the resolution level. The largest filter has resolution 0, then

$$d_k(u,v) = m_k(u,v) - m_{k+1}(u,v) \tag{A.4}$$

The dom filters provide a set of self-similar partitioning of the frequency region, each of approximately Gaussian shape.

Bisection filter

A filter that divides the frequency space, is considered along the horizontal frequency axis.

$$b(u,v) = step(v) * \omega exp[-\pi\omega^2 v^2] \tag{A.5}$$

The edge has variation in the v dimension only. ω is the sharpness parameter. Convolution with an edge is equivalent to integration, hence the above expression is rewritten as

$$b(u,v) = \int_{-\infty}^{v} \omega exp(-\pi\omega^2 r^2), dr = cum(\omega v) \tag{A.6}$$

This filter bisects along the u axis, passing the upper half of the frequency plane. The expression for a blurry edge rotated by α is given by

$$b_\alpha(u,v) = cum[\omega(v\cos\alpha - u\sin\alpha)] \tag{A.7}$$

Fan filter

The radial frequency of a sinusoid is given by the distance of the pixel from the origin, and the orientation is given by the angle relative to the horizontal axis. To select different bands in the orientation we use fan shaped regions. The fans are created by repeated bisection of the frequency space. This will ensure that the fans sum to one at all points in the frequency domain. The first cut is placed arbitrarily at an angle of 0. This methods can be used for any number of orientations that is a power of two. The fan filters of orientation bandwidth $\Delta = \pi/\Omega$ are generated by the

$$f_i = b_{i\Delta}(1 - b_{(i+1)\Delta}) + b_{(i+\Omega)\Delta}(1 - b_{(i+\Omega+1)\Delta}) \quad i = 0,..,\Omega - 1 \tag{A.8}$$

Cortex filter

The dom filter and the fan filters are multiplied to get the cortex filter. The orientation and spatial bandwidths are variable and are set here to 45° and one octave. The orientation parameter ω is set midway between the sharpness of the inner and outer tapers of the corresponding dom filter.

BIBLIOGRAPHY

[1] Algazi V.R., Kato Y., Miyahara Y., and Kotani, *Comparison of Image Coding Techniques with a Picture Quality Scale*, Proceedings of SPIE, Vol 1771, pp. 396-405, 1992.

[2] Barnsley M.F, *Fractal Everywhere*, Academic Press Professional, Massachusetts, 1993.

[3] Berns, R.S., Motta, R.J., Gorzynski,M.E., *CRT Colorimetry. Part 1:Theory and Practice*, Color Research and application, Vol 18, No 5, pp 299-314, October 1993.

[4] Chen, J., *Image Vector Quantisation using subjective criteria*, M.S Thesis, Iowa State University, Ames, Iowa, 1994.

[5] Clarke, R.J., *Transform Coding of Images*, Academic Press, Orlando, Florida, 1985.

[6] Devalois, R.L., Albrecht, D.G., and Thorell, L.G., *Spatial frquency selectivity of cells in the macaque visual cortex*, Vis.Res.Vol 22, pp 545-559, 1982.

[7] Gersho, A. and Gray, R.M, *Vector Quantisation and Signal Compression*, Kluwer Publishers, Boston, MA, 1991.

[8] Gersho, A., Vaisey, J., Zeger, K., *Globally Optimal Vector Quantizer Design by Stochastic Relaxation*, IEEE Transactions on Signal Processing, Vol 40, No. 2, pp 311-322, February 1992.

[9] Jacquin, A.E, *Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation*, IEEE Transactions on Image Processing, Vol 1, pp 18-30, January 1992.

[10] Hetherington, M.D, *Image compression using vector quantisation and lossless index coding*, M.S Thesis, Iowa State University, Ames, Iowa, 1993.

[11] Philips, G., Wilson, H., *Orientation bandwidths of spatial mechanisms measured by masking* , Journal of the Optical Society of America, Vol 1, pp 226-232, 1984.

[12] Ramamurthi, B. and Gersho, A, *Classified Vector Quantisation of Images*, IEEE Transactions on Communications, Vol 34, No 11, pp 1105-1115, November 1986.

[13] Scott, D., *The visible differences predictor: an algorithm for the assessment of image fidelity*, Proceedings of SPIE, Vol 1666, pp 2-16, 1992.

[14] Scott, D., *The Visible Differences Predictor: An Algorithm for the assessment of Image Fidelity*, Digital images and human vision edited by Andrew B. Watson,Cambridge, Massachusetts, MIT Press, pp 178-205, 1993.

[15] Sezan, M.I., Yip, K.L., Daly, S., *Uniform perceptual quantisation:Application to digital radiography*, IEEE Transactions on Sytems, Man And Cybernetics, V. SMC-17 4, pp 622-634, 1987.

[16] Stork, D.G., and Wilson, H.R., *Do Gabor functions provide appropriate descriptions of visual cortical receptive fields?*, Journal of the Optical Society of America, Vol 7, pp 1362-1373, 1990.

[17] Virsu, V., Lehtio, P., and Rovamo, J., *Contrast sensitivity in normal and pathological vision*, Doc. in Oph, Proc. Series, Ed. by L.Maffei. Vol 30, pp 263-272, 1981.

[18] Wallingford, R.M. and Gray, J.N., *Use of an X-ray process model to determine crack detectability in a multi-layer geometry*, Review of progress in QNDE, Vol 12A, pp 319-326, 1994.

[19] Watson, A., *The Cortex Transform:rapid computation of simulated neural images*, Comp. Vis. Graphics and Image Processing, Vol 39, pp 311-327, 1987.

[20] Watson, A.B., *Efficiency of a model human image code*, Journal of Optical Society of America, Vol 4, No 12, pp 2401-2417, December 1987.