A Locally Adaptive Perceptual Masking Threshold Model for Image Coding

by

Trac Duy Tran

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Bachelor of Science

and

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1994

Author ....
Department of Electrical Engineering and Computer Science
May 6, 1994

Certified by ....
Robert J. Safranek
Member of Technical Staff, AT&T Bell Laboratory
Thesis Supervisor

Certified by ....
David H. Staelin
Professor, Assistant Director of Lincoln Laboratory
Thesis Supervisor

Accepted by ....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

A Locally Adaptive Perceptual Masking Threshold Model for Image Coding

by

Trac Duy Tran

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 1994, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science
and
Master of Science in Computer Science and Engineering

Abstract

This project involved designing, implementing, and testing of a locally adaptive perceptual masking threshold model for image compression. This model computes, based on the contents of the original images, the maximum amount of noise energy that can be injected at each transform coefficient that results in perceptually distortion-free still images or sequences of images.

The adaptive perceptual masking threshold model can be used as a pre-processor to a JPEG compression standard image coder. DCT coefficients less than their corresponding perceptual thresholds can be set to zero before the normal JPEG quantization and Huffman coding steps. The result is an image-dependent gain in the bit rate needed for transparent coding. In an informal subjective test involving 318 still images in the AT&T Bell Laboratory image database, this model provided a gain on the order of 10 to 30 %.

Thesis Supervisor: Robert J. Safranek
Title: Member of Technical Staff, AT&T Bell Laboratory

Thesis Supervisor: David H. Staelin
Title: Professor, Assistant Director of Lincoln Laboratory

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Signal compression has long played a pivotal role in the technologies of long-distance communication, high-quality signal storage and message encryption. In spite of the recent promise of optical transmission media of relatively unlimited bandwidth, signal compression still remains a key technology because of our continued and increasing usage of bandlimited media such as radio, satellite links, and space-limited storage media such as solid-state memory chips and CD-ROM's. Signal compression has various applications, ranging from telephone speech, wideband speech, wideband audio, still images to digital video.

The foundations of signal compression date back to the exceptional work of Shannon in the field of information theory [16]. Shannon defined the information content of a source signal as its entropy, and mathematically showed that the source could be coded with zero error if the encoder used a transmission rate equal to or greater than the entropy, and a long enough processing delay. In particular, in the case of discrete-amplitude sources, the entropy is finite, and therefore the bit rate needed to achieve zero encoding error is also finite. We can take advantage of the statistical redundancy in the uncompressed signal to achieve a rate near or equal to the entropy.

However, there are inadequacies in this classical source coding theory. One of the most important is that the human receiver does not employ a tractable criterion such as the mean-squared error to judge the difference or similarity between the raw signal and the encoded signal. Therefore, a much more practical method of signal coding is to match the compression algorithm to the human perceptual mechanism; in the case of image coding, the Human Visual System (HVS). This leads to the development of the Perceptual Coding field.

In Perceptual Coding, the ultimate criterion of signal quality from signal compression is that perceived by the human receiver. In other words, we can push the bit rates in the digital representations of the coded signals even lower by designing the compression algorithm to minimize the perceptually meaningful measures of signal distortion rather than the mathematical criteria used in traditional source coding. Although the idea of maximizing perceived image quality rather than minimizing mean-squared error has been known and practiced for a long time, significant progress in the field of Perceptual Coding can still be made thanks to a more thorough understanding of the human visual system, as well as more aggressive, more dynamic, and more sophisticated compression algorithms. Moreover, the capabilities of digital signal processing chips have increased dramatically recently to the point where the computational complexity of such algorithms can be supported in practical hardware.

This project involves designing and testing of a new locally adaptive model for calculating the perceptual masking threshold for the Human Visual System. This model can be applied to both still images or sequences of images. Also, the model will be compatible with different coder types, i.e. general enough to be easily incorporated into any existing DCT-based image coders. A simple linear mapping with the cortex bands can also make the model compatible with other transform coders.

Chapter 2

A Review of Important Human Visual System Properties

A simplified model of the Human Visual System (HVS) is depicted in Figure 2-1 [3].

The lowpass filter in the first box represents the optical properties of the pupil. The nonlinearity helps the eye to be able to perceive a very large range of intensities. This nonlinearity is usually modeled as a logarithmic, or other similar, function. The highpass filter attempts to model the spatial response of the eye due to the interconnection of the numourous receptor regions of the retina.

The Human Visual System possesses two well-known properties that perceptual image coders have exploited. They are frequency response and texture masking.

Figure 2-2 on the following page depicts the *frequency sensitivity* of the Human Visual System. In general, the HVS acts as a peaky lowpass system. Therefore, features with high spatial frequency content require higher energy than low spatial frequency features to be visible. Special care will be given to the lower frequency region because this is where most of the image information is concentrated. Most of

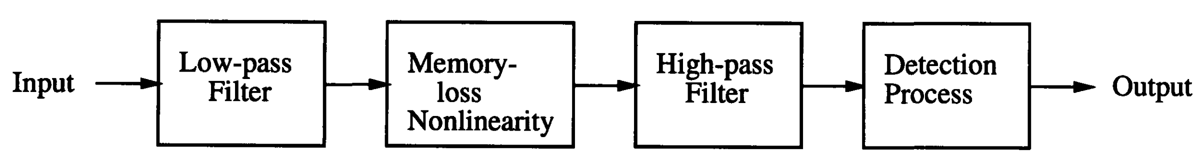

Figure 2-1: A Simple Model of the Human Visual System (HVS).

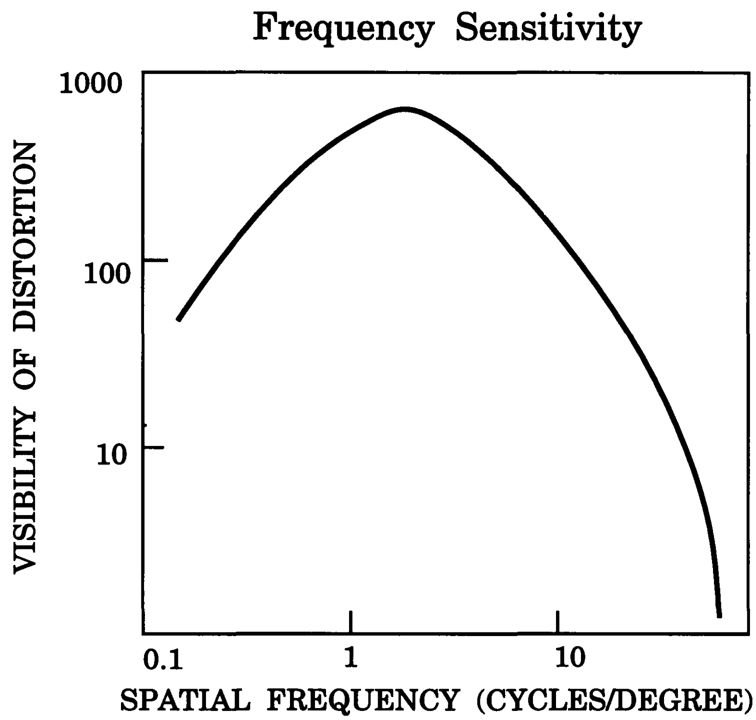

Figure 2-2: Distortion visibility as a function of spatial frequency.

the early work has taken advantage of the HVS's frequency sensitivity as described by the *modulation transfer function* (MTF) [2]. This function describes the HVS's response to sine wave gratings at various frequencies.

However, if the thresholds are obtained only from the base sensitivity of the HVS, they are certainly very conservative approximations because the fact that human eyes are far more sensitive to noise in *flat fields* than in *textured regions* has not been taken into account. But first, let's try to answer the simple question: *what is texture*? For the purpose of this project, texture can be defined as any deviation from a flat field. An image which contains a lot of texture energy is definitely not smooth. In other words, in an image region with a lot of texture, many pixels have dramatically different values.

A simple example depicted in Figure 2-3 can help clarify the HVS response to texture masking. A flat field as Region B is defined to have no texture at all. Region C has some texture, and Region A has a lot of texture energy. If a fixed amount of uniform white noise is injected into both images, the noise will be easiest to detect

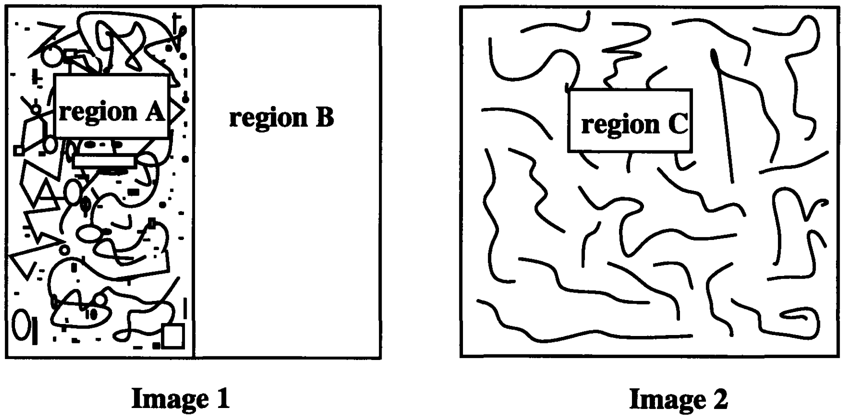

Figure 2-3: An example of texture and texture masking.

in Region B (no texture), more difficult to detect in Region C (more texture), and almost impossible to detect in Region A (most texture).

Another question about the HVS that must be answered is: *what is masking?* Simply, masking is just the change of visibility or detectability of a signal because of the presence of another signal in the same spatial frequency locality. As previously observed from the two images in the texture example (see Figure 2-3), all the white noise can be partially masked by the somewhat moderate texture in Region C, or totally masked by the heavy texture in Region A.

Besides the frequency and texture sensitivity, the HVS is also known to be more sensitive to noise at mid-grey level than at darker or lighter grey levels. Noises at the two ends of the pixel spectrum are more difficult for the eye to detect [8]. This is called the HVS *contrast sensitivity.* A more detailed and complete description of the Human Visual System's behavior can be found in Cornsweet [4].

Chapter 3

Brief Summary of Previous Related Work

3.1 Common Methodology

There has been considerable work done in the field of Perceptual Coding by engineers and researchers in the past [8]. The most common perceptual coding methodology is decribed in Figure 3-1 [8]. This methodology not only provides the framework for perceptually lossless coding at the lowest possible bit rate for common coding algorithms, but can also provide a framework for perceptually optimum performance given a certain bit rate constraint (in other words, when the available bit rate is lower than the one needed to provide transparent compression).

In the first stage of this process, a short-term or spatio-temporally local analysis of the input image is performed. In this stage, important properties of the image, such as its frequency, intensity, texture and temporal activities, are measured. These local properties are then used in the second stage of the process where the perceptual distortion thresholds are estimated. These thresholds can be a function of space or frequency, depending on the type of the coder. They are called the *just-noticable distortion profile* (JND) or the *minimally-noticable distortion profile* (MND). If the distortion or noise introduced by the compression algorithm is at or below these thresholds at all points in the space or frequency domain, the output image is guar-

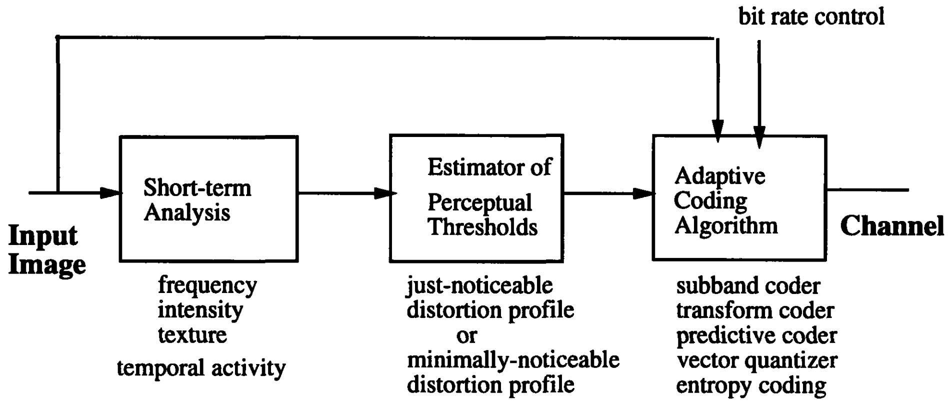

Figure 3-1: A common perceptual coding methodology.

anteed to be perceptually distortion-free. After the JND or MND profile calculation, the rest of the process is relatively straightforward. The coding algorithm uses the JND profile to introduce distortion accordingly, and this leads to minimizing the bit rate for a given image quality level or maximizing the quality level given a certain bit rate.

3.2 Image-Independent Approach

This is a very common and popular image coding method. In this approach, the JND or MND profiles are calculated independently of the images. The HVS's sensitivity to texture is not taken into account. The most popular system using this approach is the JPEG standard, which features 8x8 block-size DCT coding.

In the JPEG standard, the image is divided into 8x8-pixel blocks. Each block is then transformed to 64 DCT coefficients $I_{m,n}$. Each coefficient block is then quantized by dividing it element-wise by a quantization matrix QM with each entry labeled as $Q_{m,n}$, and rounding to the nearest integer: $U_{m,n} = \text{Round}\left[I_{m,n}/Q_{m,n}\right]$. In DCT domain, the resulting quantization error is: $E_{m,n} = I_{m,n} - U_{m,n}$. In this approach, researchers measure threshold $T_{m,n}$, or in other words, the JND profiles psychophysically. Since the maximum possible quantization error is half of the step-size $Q_{m,n}/2$, the image-independent approach can ensures that all the errors are below the thresholds, and hence invisible, by setting: $Q_{m,n} = 2\,T_{m,n}$. Details of how to design such a quantization matrix are presented in the next chapter. Finally, all the quantized coefficients $U_{m,n}$ from all of the blocks are then passed through an entropy coder to become compressed image data. See Wallace [18] for more details on JPEG.

3.3 Image-Dependent Approach

The image-dependent approach exploits the HVS's contrast and texture sensitivity. Some models based on this technique have been developed and employed by Watson [20], Daly [5], and Legge and Foley [9]. In this section, the author chooses to concentrate only on the models developed and used at AT&T Bell Laboratory at Murray Hill, where he practiced his engineering internship.

There are two perceptual masking threshold models already existing at AT&T Bell Labs. The first one is incorporated in Safranek and Johnston's Perceptually Based Sub-Band Image Coder [15]. The second one is developed by Mathews. It is called A Perceptually Masking Threshold Model for Multichannel Image Decompositions [12].

3.3.1 Safranek and Johnston's Model

This perceptual masking threshold model is simple; it only provides an approximate description of the HVS. However, it appears to work very well in practice. This model is composed of three separate components, utilizing the aforementioned well-known properties of the HVS. See [15] for a more complete description of the model.

To obtain the base sensitivity profile, Safranek and Johnston carried out numerous perceptual experiments using three trained subjects. A square of uniformly distributed random noise of known energy was added to the center of a synthetic image with mid-grey level (most sensitive to noise). Then, for each sub-band, the noise was adjusted until the subjects could not reliably determine whether the reconstructed image contained the noise square or not. Since the experiments were carried out under the most severe viewing conditions, i.e. using a stimulus that is most sensitive for the human eyes, this base model provided an overly conservative estimate of the perceptual threshold.

The base thresholds were then adjusted based on each input image's local properties. Since the HVS is more sensitive to noise at mid-grey than at lighter or darker grey levels, the thresholds were adjusted accordingly with the brightness of each input image's block. Again, subjective perceptual experiments were carried out to obtain a brightness correction curve for each sub-band. Since all these curves were similar, one brightness correction curve was utilized for all sub-bands.

The next components of the model dealt with texture masking adjustments. Texture energy was estimated by the average value of the AC energy over each analysis block in each sub-band. Then, depending on the texture energy present, a correction factor was assigned for the particular analysis block.

Obviously, this model is not locally adaptive enough. It is only adaptive block by block. In other words, all 64 transform coefficients share one common texture correction factor. Also, the masking energy measurement is also crude and inaccurate.

3.3.2 Mathews' Perceptual Masking Threshold Model for Multichannel Image Decompositions

Mathews [12] took a similar approach in designing his perceptual masking threshold model. This model consists of two components: (1) A base threshold model that does not take into account the response of the eye to the spatial details of the input image, but only describes the minimum possible threshold value for each channel, and (2) a threshold elevation model that describes how these base threshold values get elevated by the spatial details of the input image.

Mathews' base threshold model was similar to Safranek and Johnston's base sensitivity profile. His major contributions came from the threshold elevation model. Mathews observed that the threshold of detection at radial frequency f can be raised by the presence of another signal component at frequency f' depending on the following factors:

1. The ratio of the frequencies $\frac{f'}{f}$

2. The relative orientation of the two frequencies

3. The contrast (or the intensity) of the masking signal.

Based on these observations, Mathews classified the frequency coefficients into radial bands. For each radial band, he calculated the threshold elevation factor proportionally to the log of the texture energy in that band :

$$\text{threshold elevation factor} = \log_2(2 + \alpha \text{ masking energy})$$

where α was a constant that could be tuned to be just right through subjective testing.

The final threshold was then obtained as a product of the base threshold values calculated from the first component with the threshold elevation factor calculated in the second component.

This model outperformed Safranek and Johnston's model. It predicted the amount of undetectable distortion that could be injected into the lower frequency radial band reasonably well. With perceptually distortion free output images, Mathews' model provided much larger threshold values. However, it still leaves a lot of room for improvement. The model does not seem to perform as well at higher frequency bands. Also, the model is still not locally adaptive enough. All the frequency coefficients in the same radial frequency band have the same threshold elevation factor.

Chapter 4

The Peterson-Ahumada-Watson Threshold Model

This threshold model accounts for the HVS's frequency and contrast sensitivity, but not texture sensitivity. It is implemented from the detection model presented in Peterson, Ahumada, and Watson [13]. This detection model is developed to predict visibility thresholds for DCT coefficient quantization error, based on the viewing conditions and the modulation transfer function. This detection model serves as an excellent base model since it is image-independent, and is designed for various display conditions, as well as for compression in different color space. The model takes into account different pixel sizes, different viewing distances, and also different display luminances.

The thresholds are first computed in YOZ color space [13]. A simple transformation can provide the equivalent quantization matrices in other color spaces. In this project, the YC_rC_b color space is of primary interest because this is the color space utilized in digital television systems.

4.1 Quantization Matrix Design in YOZ Color Space

From various visibility threshold contrast ratio measurements, *Peterson et al* approximates that the luminance threshold of the m, nth DCT coefficient is given by:

$$\log T_{L,m,n} = \log \frac{s\, b_L}{r_L + (1 - r_L)\cos^2\theta_{m,n}} + k_L(\log f_{m,n} - \log f_L)^2, \tag{4.1}$$

with $m, n = 0, \ldots, N-1$.

The log of the luminance threshold is approximated by a parabola in log spatial frequency. The spatial frequency, $f_{m,n}$, associated with the m, nth DCT coefficient, is given by:

$$f_{m,n} = \frac{1}{2N}\sqrt{\left(\frac{m}{W_x}\right)^2 + \left(\frac{n}{W_y}\right)^2}, \tag{4.2}$$

where W_x and W_y are the horizontal and vertical size of a pixel in degrees of visual angle respectively. W_x and W_y can be calculated by the following relations:

$$W_x = \frac{\alpha_H}{\text{number of horizontal pixels}}, \text{and } W_y = \frac{\alpha_V}{\text{number of vertical pixels}}, \tag{4.3}$$

where α_H, defined as the horizontal visual angle in degrees, and α_V, the vertical visual angle, are computed as a function of the viewing distance VD measured as multiple of image heights (see illustration in Figure 4-1):

$$\alpha_H = 2\,.\,\text{Radian-to-Degree}\left(\arctan \frac{image\_width/image\_high}{2VD}\right), \tag{4.4}$$

and

$$\alpha_V = 2\,.\,\text{Radian-to-Degree}\left(\arctan \frac{0.5}{VD}\right) \tag{4.5}$$

The angular parameter, which accounts for the HVS orientational dependency, is given by:

$$\theta_{m,n} = \begin{cases} 0.0, & m = n = 0 \\ \arcsin \frac{2 f_{m,0} f_{0,n}}{f_{m,n}^2}, & \text{otherwise}. \end{cases} \tag{4.6}$$

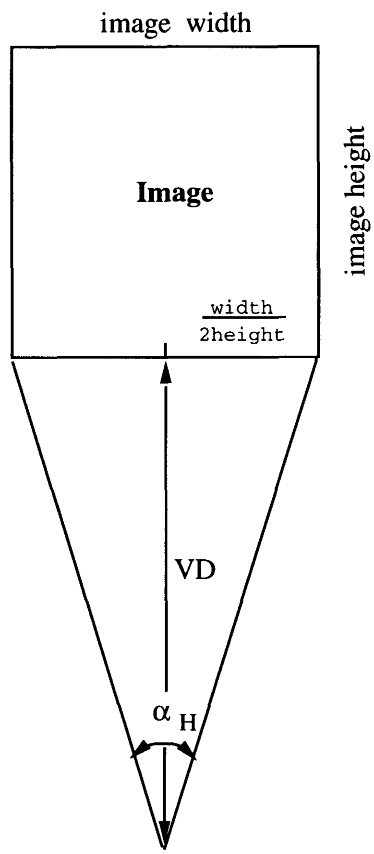

Figure 4-1: Calculation of visual angle α.

model channel	parameter values s	r	f	k	b
Y	0.25	0.6	3.1	1.34	0.0219 Y_o
O	0.25	0.6	1.0	3.0	0.0080 Y_o
Z	0.25	0.6	1.0	3.0	0.0647 Z_o

Table 4.1: Parameter Values for Peterson *et al*'s Base Model.

The factor $r_{OZ} + (1 - r_{OZ}) \cos^2 \theta_{m,n}$ is to account for the summation-obliqueness effect of the Fourier components. The magnitude of this effect is controlled by the parameter r_L. Based on the fourth power summation rule for the two Fourier components [1], r_L is set to 0.6. The minimum luminance threshold sb_L, occuring at spatial frequency f_L, and the remaining parameter k_L determines the steepness of the luminance parabola. The parameter $0.0 < s < 1.0$ accounts for visual system summation of quantization errors over a spatial neighborhood.

Similar measurements were carried out for the chrominance channels, and the resulting log chromatic thresholds for the m, nth DCT basis function are given by:

$$\log T_{O,m,n} = \begin{cases} \log \frac{sb_O}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}}, & \text{if } f_{m,n} \leq f_{OZ} \\ \log \frac{sb_O}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}} + k_{OZ}(\log f_{m,n} - \log f_{OZ})^2, & \text{if } f_{m,n} > f_{OZ}. \end{cases} \tag{4.7}$$

and

$$\log T_{Z,m,n} = \begin{cases} \log \frac{sb_Z}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}}, & \text{if } f_{m,n} \leq f_{OZ} \\ \log \frac{sb_Z}{r_{OZ}+(1-r_{OZ})\cos^2\theta_{m,n}} + k_{OZ}(\log f_{m,n} - \log f_{OZ})^2, & \text{if } f_{m,n} > f_{OZ}. \end{cases} \tag{4.8}$$

All of the parameters used to implement this new base threshold model are listed in Table 4.1. $Y_o = 41.19$ and $Z_o = 29.65$ are the CIE values of average white (D65).

4.2 Conversion of Quantization Matrix to YC_rC_b Color Space

As described in the previous section, the thresholds in color space YOZ can be calculated from Equations 4.1, 4.7, and 4.8, the pixel sizes W_x, W_y, and the parameters given in Table 4.1. These thresholds can be transformed to the YC_rC_b color space in the following way.

The transformation can be thought of as limiting the errors in each of the channels Y, C_r, C_b such that the resulting errors in the Y, O, and Z channels are all below the previously calculated thresholds. The linear transformation matrix $M_{Y'C_rC_b \to YOZ}$ relates the errors in the two color spaces. For example, a unit error in a DCT coefficient in channel C_r induces errors of magnitude $|M_{2,1}|$, $|M_{2,2}|$, and $|M_{2,3}|$ in the Y, O, and Z channels respectively:

$$M_{Y'C_rC_b \to YOZ} = M_{Y'C_rC_b \to XYZ} \times M_{XYZ \to YOZ} = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix} \tag{4.9}$$

The transformation matrix from color space Y', C_r, C_b, to color space YOZ is given below. Y' is used to help clear up the notational confusion only.

$$M_{Y'C_rC_b \to YOZ} = \begin{bmatrix} 66.9 & -1.1 & 48.2 \\ -17.8 & 17.1 & -4.5 \\ -7.0 & 0.6 & 67.9 \end{bmatrix}. \tag{4.10}$$

The YOZ model thresholds are then converted to the Y' threshold. $T_{Y \to Y',m,n}$ is the threshold imposed on channel Y' by the threshold of channel Y.

$$T_{Y \to Y',m,n} = \frac{T_{Y,m,n}}{|M_{1,1}|},\ T_{O \to Y',m,n} = \frac{T_{O,m,n}}{|M_{1,2}|},\ \text{and}\ T_{Z \to Y',m,n} = \frac{T_{Z,m,n}}{|M_{1,3}|}. \tag{4.11}$$

Similarly,

$$T_{Y\to C_r,m,n} = \frac{T_{Y,m,n}}{|M_{2,1}|},\ T_{O\to C_r,m,n} = \frac{T_{O,m,n}}{|M_{2,2}|},\ \text{and}\ T_{Z\to C_r,m,n} = \frac{T_{Z,m,n}}{|M_{2,3}|}. \tag{4.12}$$

$$T_{Y\to C_b,m,n} = \frac{T_{Y,m,n}}{|M_{3,1}|},\ T_{O\to C_b,m,n} = \frac{T_{O,m,n}}{|M_{3,2}|},\ \text{and}\ T_{Z\to C_b,m,n} = \frac{T_{Z,m,n}}{|M_{3,3}|}. \tag{4.13}$$

Then the minimum rule is used to decide the final thresholds. The minimum rule ensures the most conservative approximations of the visible quantization errors.

$$T_{Y',m,n} = \min\{T_{Y\to Y',m,n},\ T_{O\to Y',m,n},\ T_{Z\to Y',m,n}\}, \tag{4.14}$$

$$T_{C_r,m,n} = \min\{T_{Y\to C_r,m,n},\ T_{O\to C_r,m,n},\ T_{Z\to C_r,m,n}\}, \tag{4.15}$$

$$T_{C_b,m,n} = \min\{T_{Y\to C_b,m,n},\ T_{O\to C_b,m,n},\ T_{Z\to C_b,m,n}\}, \tag{4.16}$$

The final quantization matrix entries in Y'$C_r C_b$ space are obtained by dividing the new thresholds by the DCT normalization constants α (given in Equation 5.2):

$$Q_{Y',m,n} = 2\,\frac{T_{Y,m,n}}{\alpha_m\,\alpha_n},\quad Q_{C_r,m,n} = 2\,\frac{T_{C_r,m,n}}{\alpha_m\,\alpha_n},\quad Q_{C_b,m,n} = 2\,\frac{T_{C_b,m,n}}{\alpha_m\,\alpha_n}. \tag{4.17}$$

Actually, in this project, since we are interested in the base threshold value, i.e. the maximum tolerable quantization error, we only have to compute the quantity $\frac{T_{m,n}}{\alpha_m\,\alpha_n}$. The factor 2 in Equation 4.17 refers to the obvious fact that the maximum possible quantization error is half the quantizer's step size. See [13] for a more detailed discussion on this base model.

4.3 Implementation of Base Thresholds for CIF Images

Since the test images or sequences are available in CIF standard, we have to implement the Peterson-Ahumada-Watson base threshold model accordingly. The implementation is almost exactly the same as described in the previous two sections of the chapter. There are only a few minor changes.

CIF standard images are in Y'$C_r C_b$ color space, but the two chrominance channels are down-sampling by a factor of 2. For display, the chrominance channels are then up-sampled, (while the luminance channel stays the same), and the whole image is converted to RGB space. All the CIF standard images have dimension 360x240. Therefore, the chrominance channels have dimension 180x120.

For the luminance channel we have the full number of pixels in both dimensions. For a fixed viewing distance, this translates to a value for α as demonstated in Figure 4-1. From this value of α, we can calculate the corresponding W_x and W_y for the luminance channel. However, for chrominance channels, we have the same viewing conditions, hence the same value of α, but the chrominance channels have been down-sampled by a factor of 2 in both dimensions. This means we only have half the number of pixels which results in W_x and W_y for the chrominance channel being double the luminance values. The base weights computed for a viewing distance of 3 image heights for CIF images in color space Y'$C_r C_b$ are given in Table 4.2 in the following page.

In order to make the base model design more robust, several modifications were added. The first modification accounts for the dependence of the detection thresholds on viewing distance. The aforementioned design procedure is performed for the minimum given viewing distance. The viewing distance is then increased, and the thresholds are recomputed. The output thresholds are now set to the minimum of the two calculated thresholds, and the procedure is repeated until a certain maximum viewing distance is reached. The iteration ensures that there is no visible distortion at any viewing distance greater than the minimum.

The second modification is to account for the dependence of the detection thresholds on viewing condition. It is implemented in a similar fashion. In this case, for each iteration, instead of increasing the viewing distance, a new set of white point is installed (by changing the Y_0 and X_0 values in Table 4.1). The final output thresholds are set to the minimum of all the thresholds computed from all the white points.

Y' base weights	2.0	3.5	3.5	3.5	3.5	3.5	4.0	4.0
	3.5	3.0	3.5	3.0	2.5	2.5	3.0	3.0
	3.5	2.5	3.0	4.0	3.5	3.5	3.5	3.5
	3.5	2.5	3.0	3.5	4.0	4.5	4.5	4.5
	4.0	2.5	3.0	3.5	4.0	5.0	6.0	5.5
	4.5	3.0	3.5	3.5	4.0	5.0	5.5	7.0
	5.0	3.5	4.0	4.0	4.5	5.0	6.0	7.0
	6.0	4.5	4.5	5.0	5.0	5.5	6.5	7.0
C_r base weights	7.0	7.0	7.0	7.5	9.5	13.0	15.0	16.5
	7.0	3.5	3.5	4.0	5.5	8.5	11.0	12.0
	7.5	4.5	4.0	5.0	6.5	9.0	13.0	14.0
	11.0	7.0	7.0	7.5	9.5	12.5	16.5	17.5
	15.0	11.0	12.0	13.0	15.0	18.5	22.5	22.0
	17.0	12.5	13.0	14.5	16.5	19.0	22.0	26.0
	20.0	14.5	15.0	16.5	18.0	20.0	23.0	26.5
	24.0	17.0	17.5	18.5	20.0	22.5	25.0	28.0
C_b base weights	14.0	14.0	14.0	14.5	18.0	25.5	36.5	42.0
	14.0	7.0	7.0	8.0	11.0	16.5	24.5	31.5
	14.5	8.5	8.0	10.0	13.0	17.5	25.5	36.0
	21.5	14.0	13.5	15.0	18.5	24.5	32.5	44.5
	36.5	25.0	24.5	25.5	29.5	36.5	47.0	56.5
	44.0	32.0	34.0	37.0	42.0	48.5	56.5	67.0
	51.5	37.0	39.0	42.0	46.0	52.0	59.0	68.0
	61.0	43.5	45.0	48.0	52.0	57.0	63.5	72.0

Table 4.2: Base weights for CIF images in $Y'C_rC_b$ color space (for VD=3).

Chapter 5

Mapping of DCT coefficients on the Cortex Filters

In perceptual image coding, the choice of the filterbank which has the HVS's structure is very important to the performance of the compression system. The DCT (Discrete Cosine Transform) does not meet this crucial criterion. This leads to difficulty in creating an effective masking model for DCT-based coder since there is a mismatch between the underlying structure of the model and the structure of the DCT. The algorithm presented in this chapter maps the DCT transform coefficients onto the Cortex transform filters, which mimics the the visual system's structure [19]. The mapping helps to decide which DCT coefficients contribute how much energy to which Cortex transform's critical bands. This is a pivotal component of the perceptual masking threshold model since it provides the model's local adaptability, and it solves the aforementioned mismatch problem.

5.1 The Discrete Cosine Transform

The DCT has recently become a standard method of image compression. The JPEG, MPEG, and CCITT H.261 image compression standards all employ the DCT as a basic mechanism. In the Forward DCT, the image pixels are divided into 8x8 blocks; each block is then transformed into 64 DCT coefficients. The DCT transform

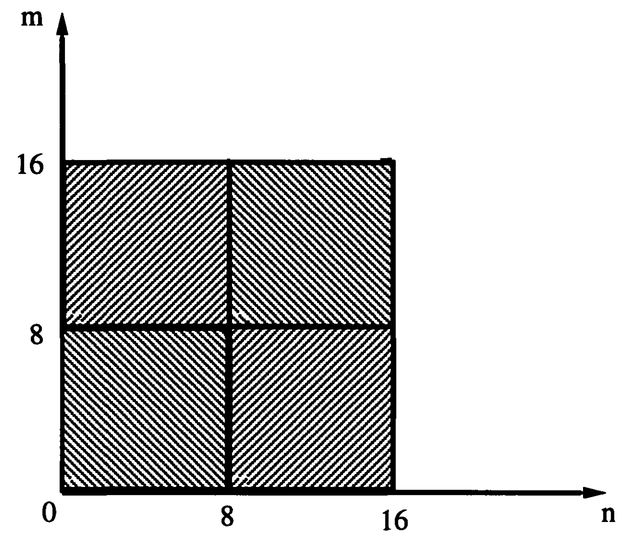

Figure 5-1: Symmetric Replication of an Image Block.

coefficients $I_{m,n}$ of an $N \times N$ block and its image pixels $i_{j,k}$ are related by the following equations:

$$I_{m,n} = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} i_{j,k} c_{j,m} c_{k,n} \ , \qquad \text{with } m,n = 0,\ldots,N-1, \tag{5.1}$$

where

$$c_{j,m} = \alpha_m \cos\left(\frac{\pi m}{2N}[2j+1]\right) \ , \qquad \text{and } \alpha_m = \begin{cases} \sqrt{1/N}, & m = 0 \\ \sqrt{2/N}, & m > 0 \end{cases} \tag{5.2}$$

and

$$i_{j,k} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} I_{m,n} c_{j,m} c_{k,n} \ , \qquad \text{with } j,k = 0,\ldots,N-1, \tag{5.3}$$

The coefficient with zero frequency in both dimensions m,n is called the *DC coefficient*; the remaining 63 coefficients are called the *AC coefficients.*

The DCT is closely related to the Discrete Fourier Transform (DFT). If the 8x8-pixel block is flipped and replicated in such a way that the new 16x16-pixel block is symmetric as demonstrated in Figure 5-1, the 16x16-point DFT of the new image block are very closely related to the 8x8-point DCT [10]. The general framework is shown below:

$$N \times N\, i_{j,k} \longleftrightarrow 2N \times 2N\, y_{j,k} \overset{DFT}{\longleftrightarrow} 2N \times 2N\, Y_{m,n} \longleftrightarrow N \times N\, I_{m,n}\,,$$

where the relation between $Y_{m,n}$ and $I_{m,n}$ is given by:

$$I_{m,n} = e^{-j\frac{\pi}{N}\frac{m}{2}}\, e^{-j\frac{\pi}{N}\frac{n}{2}}\, Y_{m,n}\,. \tag{5.4}$$

The above relation ensures the mapping's validity since the cortex transform is also performed in DFT domain with symmetrically replicated data.

5.2 The Cortex Transform

The Cortex transform was first introduced by Watson as a rapid computation of simulated neural images [19]. It was later modified and used by Daly in his visible differences predictor (VDP) [5]. The Cortex transform originates from researches in neurophysiology [7] [6] and psychophysical studies in masking [2] [17]. These studies have found a radial frequency selectivity that is essentially symmetric on a log frequency axis with bandwidths nearly constant at one octave. Furthermore, these studies also discovered that the HVS's orientation selectivity is symmetric about a center peak angle with tuning bandwidths varying as a function of radial frequency, ranging from 30 degrees for high frequencies to 60 degrees for low frequencies [14]. These familiar properties of the HVS were also noted and exploited by Mathews [12] in his previously mentioned masking model design.

The frequency selectivity of the HVS was modeled by Watson, and then modified by Daly, as a hierarchy of filters called the *Cortex filters*. The radial selectivity and orientational selectivity in the Cortex transform are modeled with separate classes of filters that are cascaded to give the combined radial and orientational selectivity of the HVS. Note that this is only an attempt to approximate the human visual system. By splitting the original image spectrum into many spatial images with the Cortex filters, we can model the space-frequency localization aspects of the HVS.

The Cortex transform, named after the striate cortex where neurons demonstrating the radial and orientational effects are found, is picked because it proves to model the HVS very accurately as demonstrated in Daly's Visible Differences Predictor. Moreover, the cortex transform is reversible, flexible, and also easy to implement. Its disadvantages, such as non-orthogonality and computational complexity, do not concern us since we do not have the perfect-reconstruction constraint, and we only have to run the mapping algorithm once. Once the mapping has been found out, the result can be used for all DCT blocks.

The cortex filters are formed as a separable product of the radial frequency and the orientational frequency filters. In order to ensure the reversibility of the cortex filters' set, i.e. the sum of the filters is 1, the radial frequency bands are formed as differences of a series of 2D low-pass *mesa* filters which have a flat pass-band, a flat stop-band, and a Hanning-window transition-band. The mesa filter can be completely characterized by its half-amplitude frequency, $\rho_{\frac{1}{2}}$, and its transition width, *tw*:

$$mesa(\rho) = \begin{cases} 1.0 & \text{for } \rho < \rho_{\frac{1}{2}} - \frac{tw}{2} \\ \frac{1}{2}\left(1 + \cos\left(\frac{\pi(\rho - \rho_{\frac{1}{2}} + \frac{tw}{2})}{tw}\right)\right) & \text{for } \rho_{\frac{1}{2}} - \frac{tw}{2} < \rho < \rho_{\frac{1}{2}} + \frac{tw}{2} \\ 0.0 & \text{for } \rho > \rho_{\frac{1}{2}} + \frac{tw}{2} \end{cases} \tag{5.5}$$

The *k*th *dom* (differences of mesas) filter is simply the difference of two mesa filters evaluated at two different half-amplitude frequencies:

$$dom_k(\rho) = mesa(\rho)|_{\rho_{\frac{1}{2}}=2^{-(k-1)}} - mesa(\rho)|_{\rho_{\frac{1}{2}}=2^{-k}} \tag{5.6}$$

The lowest frequency filter, called the *base*, is designed differently. A truncated Gaussian function is used instead of the mesas to get rid of the unacceptable ringing in the base-band:

$$base(\rho) = \begin{cases} e^{-\left(\frac{\rho^2}{2\sigma^2}\right)} & \text{for } \rho < \rho_{\frac{1}{2}} + \frac{tw}{2} \\ 0.0 & \text{for } \rho \geq \rho_{\frac{1}{2}} + \frac{tw}{2} \end{cases} \tag{5.7}$$

where

$$\sigma = \frac{1}{3}\left(\rho_{\frac{1}{2}} + \frac{tw}{2}\right) \; ; \qquad \rho_{\frac{1}{2}} = 2^{-K}, \tag{5.8}$$

with K being the total number of radial filters. The transition width tw of each filter is defined to be a function of its half-amplitude frequency, as given by

$$tw = \frac{2}{3}\,\rho_{\frac{1}{2}} \; . \tag{5.9}$$

This choice of transition width gives the Cortex bands constant behavior on a log frequency axis with a bandwidth of 1.0 octave and symmetric response.

The HVS's orientational frequency selectivity is modeled by a set of *fan* filters. A Hanning window is also used for these filters. The orientation transitions are functions of angular degrees θ in Fourier domain. The lth fan filter is given by,

$$fan_l(\theta) = \begin{cases} \frac{1}{2}\left(1 + \cos\left(\frac{\pi|\theta-\theta_c(l)|}{\theta_{tw}}\right)\right) & \text{for } |\theta - \theta_c(l)| \le \theta_{tw} \\ 0.0 & \text{for } |\theta - \theta_c(l)| > \theta_{tw} \; , \end{cases} \tag{5.10}$$

where θ_{tw} is the angular transition width, and $\theta_c(l)$ is the orientation of the center angular frequency of fan filter l, given by,

$$\theta_c(l) = (l-1)\,\theta_{tw} - 90^o \; ; \qquad \theta_{tw} = \frac{180^o}{L} \; , \tag{5.11}$$

with L being the total number of fan filters.

The cortex filters are then formed by a simple polar multiplication of the corresponding dom and fan filter:

$$cortex_{k,l}(\rho,\theta) = \begin{cases} dom_k(\rho).fan_l(\theta) & \text{for } k = 1,\ldots,K-1;\; l = 1,\ldots,L \\ base(\rho) & \text{for } k = K \; . \end{cases} \tag{5.12}$$

The total number of cortex filters is L(K-1)+1. For the mapping, we use K=6 and L=6, combining for a total of 31 critical cortex bands. Notice that there is no orientation selectivity in the baseband, and the choice of L yields an orientation

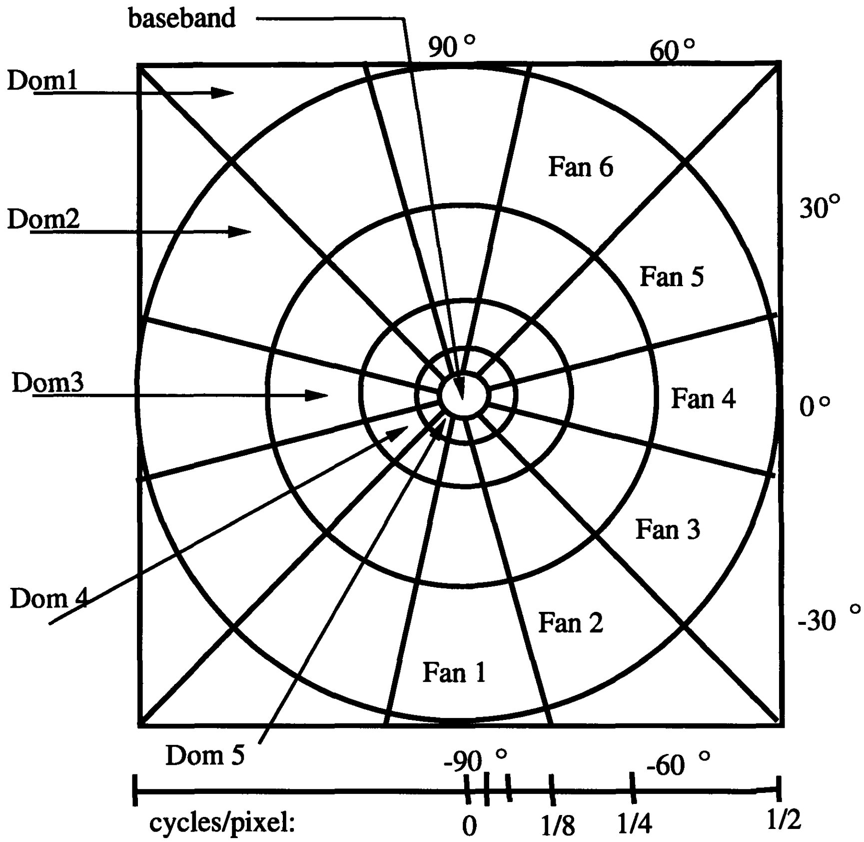

Figure 5-2: Complete set of cortex filters for K=6 and L=6.

bandwidth of 30 degrees, which is consistent with studies in [14]. Also, the set of cortex filters is invertible, i.e.

$$\sum_{k=1}^{K}\sum_{l=1}^{L} cortex_{k,l}(\rho,\theta) = 1 \qquad \text{for all } \rho,\theta\,. \tag{5.13}$$

Details of the radial and orientational dissections of the frequency space are shown in Figure 5-2. See [5] and [19] for more detailed description and implementation of the cortex filters.

5.3 The DCT-Cortex Transform Mapping

This DCT-Cortex transform masking serves as the heart of our locally adaptive texture masking model. As previously mentioned, texture masking, or contrast masking in some other literatures, refers to the reduction of visibility or detectability of one image signal by the presence of another. The texture masking characteristic of the HVS is known to be dependent on three major factors. The masking is strongest (and, therefore, the thresholds can be elevated highest) when both signals are of the same location, orientation, and spatial frequency [12] [20]. The cortex filters divide up the image spectrum in a similar fashion. They are nothing more than a set of windows that cover the whole frequency spectrum. Signal components at the same location, orientation, and spatial frequency are grouped together in the same Cortex band. Moreover, in this project, we only consider texture masking within a DCT block. Therefore, a simple mapping of the two can help us decide which DCT coefficients contribute how much energy to which cortex band, and from that information, elevate these coefficients' base thresholds accordingly with the intensity of the texture energy present in that cortex band. Notice that this idea can also be applied to masking across the DCT blocks. The performance of the model would definitely be enhanced by such algorithm thanks to an increase in masking accuracy. However, the computation for such a model would also be more costly because of the increase in complexity.

The mapping algorithm's complexity lies heavily on the implementation of the cortex filters. Once this task is done, the mapping reduces to 64 numerical integrations of 64 DCT bins over each cortex band.

The algorithm takes in a resolution number, a threshold, K, and L as its inputs, and produces one 8x8 overlap-area matrix for each cortex band. K, chosen to be 6 in this project, is the number of dom filters; $L = 6$, is the number of fan filters. The threshold is used to produce a *binary overlap area matrix* for convenience. If an overlap area is greater than the threshold, then it is set to 1. Otherwise, it is set to 0. These binary matrices are used in the early stages when we were setting

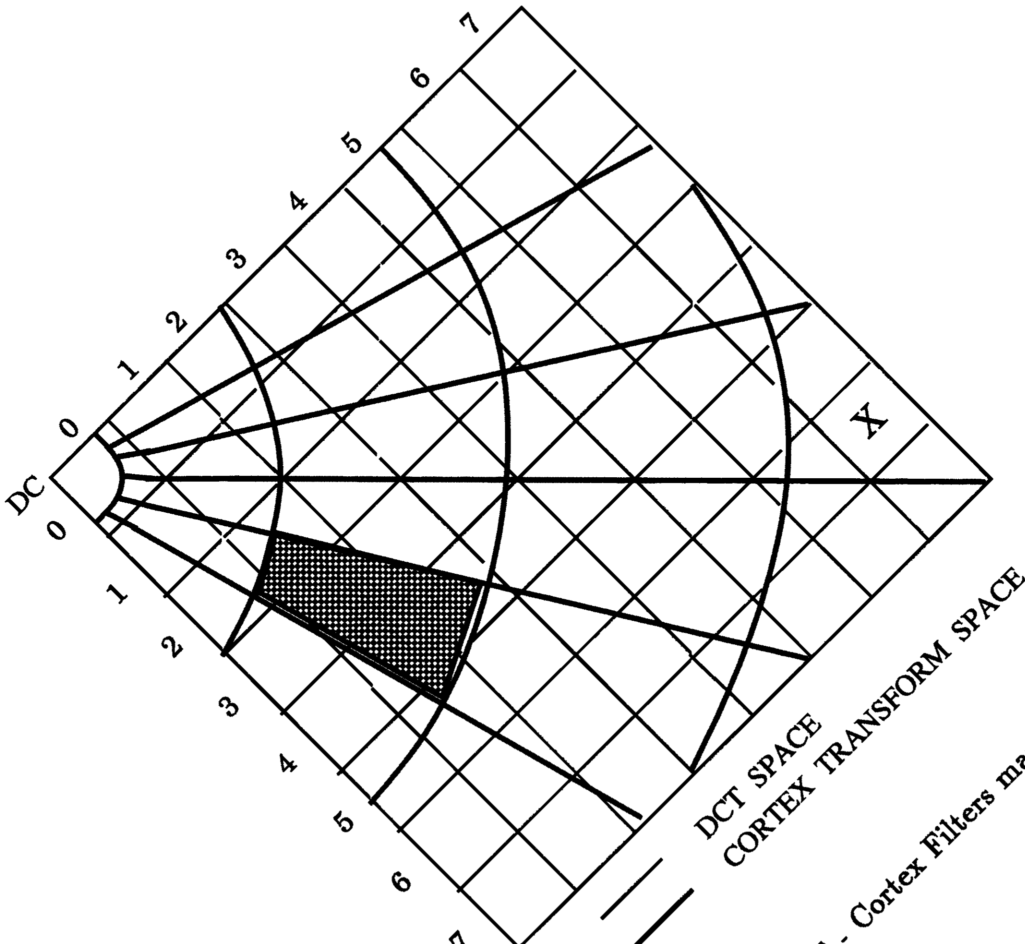

Figure 5-3: DCT - Cortex Filters mapping.

up the overall framework for the project. In later stages, when the elevation model's accuracy becomes an important issue, the actual overlap area values are always used. The resolution *resnum* provides the finer scale for the numerical integrations. Notice that more accuracy can be achieved by higher resolution. However, computational complexity is the trade off. A commonly used value for *resnum* is 32.

Each DCT bin is further divided into $resnum \times resnum$ subbins. The cortex transform of length $8 \times resnum$ is then performed to give us 31 sets of 31 cortex filters' cefficients. For the kth, lth cortex band, the mth, nth entry of the *overlap area matrix*, $\text{Overlap-area}_{k,l,m,n}$ – the overlap area between the DCT coefficient $I_{m,n}$ and the aforementioned cortex band is computed by the summation:

$$\text{Overlap-area}_{k,l,m,n} = \sum_{kk=m.resnum}^{kk=(m+1)resnum} \sum_{ll=n.resnum}^{ll=(n+1)resnum} cortex_{k,l}(kk, ll)\,. \tag{5.14}$$

The final output of the DCT-Cortex mapping algorithm is a set of 31 8x8 matrices. Each matrix contains 64 overlap-area values of 64 DCT bins and the corresponding cortex band. Since we are only mapping the primary quadrant of the cortex space to the DCT space (with the DC value line up in the middle of the base band), only 21 out of 31 cortex bands participate in the mapping. The remaining 10 bands have all-zero overlap-area matrices. These overlap-area matrices serve as the basis for the elevation model described in the next chapter. The matrices computed at a resolution of 32 are included in Appendix A.

Chapter 6

The Threshold Elevation Model

This image-dependent threshold elevation model estimates the texture energy, i.e. the amount of spatial details, in each DCT block, and computes a threshold elevation factor for each DCT coefficient.

6.1 Basic Strategy

The threshold elevation model uses a mapping of the DCT coefficients on the Cortex filter bands as described in the previous chapter. For each Cortex band , the model decides which coefficients contribute how much energy to that band, and then increases the elevation factor of those coefficients linearly with the intensity of the Cortex band's masking energy. For the shaded Cortex band in Figure 5-3, the DCT transform coefficients $I_{1,2}, I_{1,3}, I_{1,4}$, and $I_{2,4}$ contribute most of the texture energy in the band, whereas coefficient $I_{7,6}$ (marked X) has zero contribution. Therefore, the elevation factor of $I_{7,6}$ should not be dependent on the amount of texture present in the shaded Cortex band. This idea comes from the HVS's tendency to be strongly dependent on locality. Notice also that all the coefficients in the same cortex band share very close spatial frequencies, orientations, and locations. This ensures that the noise introduced by the threshold elevation will be appropriately masked by the texture energy of the corresponding cortex band.

If a certain DCT coefficient gets involved in more than one Cortex band, and thus correspondingly has more than one elevation factor, we apply the *minimum-of rule*, i.e. the smallest value will be used to prevent overly aggressive estimation.

If there is zero or very little texture energy in the analysis block, there should be no elevation at all, that is, the elevation factor is 1. In this case, the model uses only the conservative base threshold value. There should also be a maximum cut-off value for the elevation factors because noise masking can achieve transparency only up to a certain level. Through observation in the subjective visual tests in this project, if the noise energy exceeds roughly 25 percent of the masking signal energy, distortion will be most likely visible in the reconstructed image. For the cortex band which contains energy between the minimum and the maximum cut-off point, the elevation factors of the contributing DCT coefficients are increased accordingly with the elevation curve. The maximum cut-off value parameter, as well as the characteristics of the elevation curve, can be determined and fine-tuned through subjective testing.

In an image block, the final perceptual masking threshold of any DCT coefficient is then obtained as the product of its base threshold value and its elevation factor. If an image-dependent quantization matrix is desired, each entry of the matrix is simply twice the corresponding final masking threshold.

6.2 Implementation

Each image pixel block of size 8x8 is transformed to its equivalent 64 DCT coefficients $I_{m,n}$. For a fixed viewing condition and a fixed viewing distance, an 8x8 base threshold matrix $T_{base,m,n}$ for each channel is computed according to the method described in Chapter 4. These thresholds can be elevated in accordance with the block's texture energy intensity. A simple threshold elevation model for texture masking in the luminance channel can be implemented in the following ways. (Note that the design and implementation of the chrominace channels' elevation model are exactly the same.)

The algorithm reads in as input the overlap-area matrices generated as described in Chapter 5. For each cortex band (matrix), a set of elevation factors are calculated,

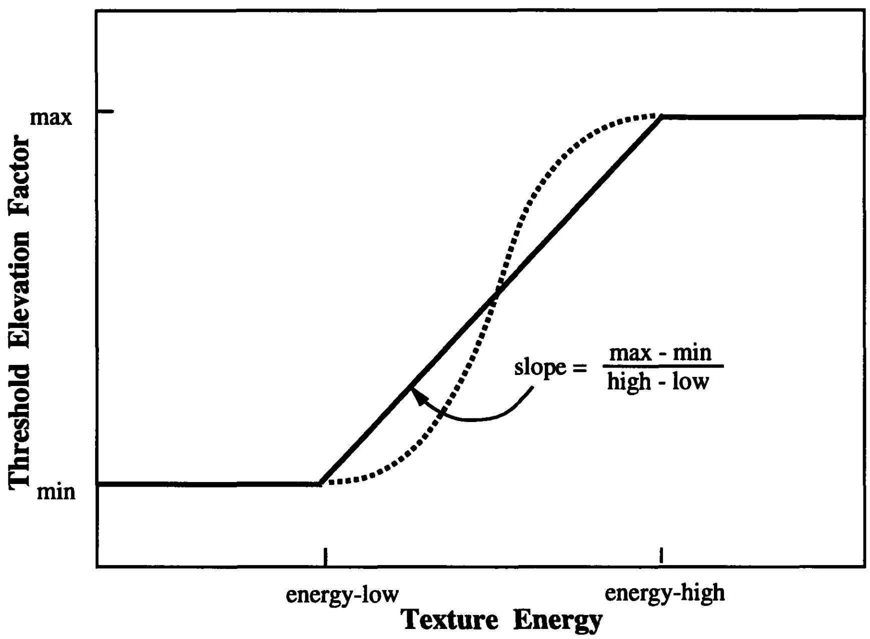

Figure 6-1: A Simple Threshold Elevation Model.

based on the 3-segment piecewise linear texture elevation model shown in Figure 6-1.

The texture energy of the *k, l* th cortex band is computed by summing up the energy of all the 63 AC coefficients that overlap with that cortex filter's passband, and then taking the square root of the summation:

$$\text{total\_energy}_{k,l} = \sqrt{\sum_{m=0}^{7}\sum_{n=0}^{7}(I_{m,n}\,.\text{Overlap-area}_{k,l,m,n}\,/\,T_{base,m,n})^2} \text{ for } m \neq 0 \text{ or } n \neq 0. \tag{6.1}$$

The DC value is the average of the pixel values in the block, so it has substantially more energy than the AC coefficients. Especially in the case of uniform light background, (i.e. no texture energy in the block but the pixel values are high), the DC coefficient is large while the AC coefficients are all in the vicinity of 0, 1, or -1. Therefore, the DC term is excluded from the energy calculation. The elevation model also takes into account the viewing distance and the viewing conditions by normalizing the total energy of the cortex band by the base thresholds $T_{base,m,n}$.

If the total energy just computed is less than the low energy threshold, then the elevation is set to a minimum. Obviously, the minimum value is picked to be 1, meaning that there is no threshold elevation. If the total energy is greater than the high energy threshold, then the elevation factor is set to a maximum value. If the total energy is in between the two energy thresholds, then the elevation factor increases linearly with the energy. In short, for the involved DCT coefficients (overlap-area$_{k,l,m,n} \neq 0$), the elevation factors can be calculated by:

$$\text{elevation\_factor}_{k,l,m,n} = \begin{cases} min, & \text{total\_energy}_{k,l,m,n} < energy\_low \\ max, & \text{total\_energy}_{k,l,m,n} > energy\_high \\ else: & \\ min+ & \\ \frac{max-min}{energy\_high-energy\_low}(\text{total\_energy}_{k,l,m,n} - energy\_low). & \end{cases} \tag{6.2}$$

This elevation curve does not have to be linear. In fact, a cubic curve (the dotted line in Figure 6-1), is probably a more logical choice because a smoothing function makes a more accurate approximation of the HVS's sensitivity to texture than segmented lines with discrete decision regions. However, the linear elevation model is the easiest and most straightforward to implement. It is also the most computationally inexpensive choice. It serves as a good cornerstone for the elevation model. Moreover, it appears to work quite well in practice.

As previously mentioned, if a coefficient contributes energy to more than one cortex band, its final elevation factor is the minimum of all the values calculated. A variable called *etemp* keeps the current value computed for the current iteration. It is then compared with the minimum elevation factor calculated so far from the previous iterations. If this minimum-so-far value is greater than *etemp*, then it is updated. Otherwise, it stays the same. Notice that among 64 DCT coefficients, everyone of them belongs to at least one cortex band. So, none of them gets left out from the iterations. Second, for a particular iteration for a particular cortex band, all the coefficients belonging to that band have the same temporary elevation factor *etemp*. The "minimum of" rule can make two coefficients that contribute about the same amount of energy to the same cortex band have different elevation factors.

There is one exception for this "minimum of" rule. As shown in the overlap-area matrices in Appendix A, there are several coefficients that have major energy contribution in certain cortex band, say more than 80 percent. They are also involved in some other cortex bands; however, the contribution level is much lower, say 10 percent or less. For a particular coefficient of this type, we would like to use the elevation factor calculated from the cortex band that it is most influential, not the aforementioned minimum value. We call this *high energy contribution reconsideration.*

Another little adjustment for the elevation model is the *low frequency post - processing.* Not only is the DC coefficient $I_{0,0}$ sensitive to noise, but it is also known from the HVS's low-pass nature that the DC's low frequency neighbors $I_{0,1}, I_{0,2}$, $I_{1,0}, I_{1,1}$, $I_{1,2}, I_{2,0}$, and $I_{2,1}$ are very important to be coded right. Therefor, we set all elevation factors of these low frequency coefficients to $min = 1$.

The final perceptual masking threshold of a coefficient is obtained as a product of its base threshold and its elevation factor. This final threshold is most likely different for the 64 coefficients in the same block. Also, the threshold for coefficient $I_{m,n}$ at frequency bin *m, n* in the i th block is most likely different from the threshold of the coefficient at the same spatial frequency in block j. For the JPEG standard, these two thresholds are exactly the same. These two facts show the locally adaptive nature of the new perceptual masking threshold model.

It should be noted that such a model designed in this fashion does not guarantee a performance at perceptually distortion-free level. However, through subjective testing, we can fine-tune the parameters enough to achieve this goal. The parameters do not have to be the same for all of the cortex bands. In fact, they should be different. For example, for the cortex bands that cover the lower frequency spectrum, the elevation model has to be more conservative. The model can be more aggressive with the cortex bands in the high frequency regions.

Chapter 7

Block Classification

The need for block type classification arose when we conducted early subjective tests of the threshold elevation model. Noises resulting from high elevation factors of coefficients in high textured region within an image block spead out to the remaining uniform background region of the block. This noise spreading is similar to the familiar pre-echoing problem in perceptual audio coding.

7.1 Problem Description and Early Results

Let us take a close look at what we label an *edge-block* in Figure 7-1, and the threshold elevation model's performance on the corresponding image data.

The definition of an edge in this case is not the same as the one used in numerous edge-detection techniques. An edge-block in our definition is an image block that contains two obvious regions: one contains very high texture energy (the left shaded region in Figure 7-1), and the other is a "clean" uniform background, i.e. has almost zero texture energy (the region on the right). Such an image block has pixel values given in Table 7.1; its equivalent 64 DCT coefficients are shown in Table 7.2, with the DC coefficient $I_{0,0} = 937$ at the upper left corner and the highest frequency coefficient $I_{7,7} = -87$ at the lower right corner. Since the textured region of the block has quickly varrying pixel values, the DCT coefficients are quite large, even at high frequencies.

The coefficients in Table 7.2 are then coded using the perceptual masking threshold

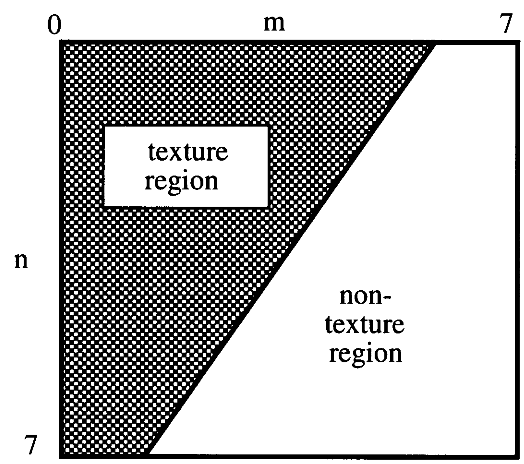

Figure 7-1: Example of an edge-block.

12	89	23	231	202	7	130	130
189	33	76	91	240	130	130	130
200	230	55	23	19	130	130	130
42	220	35	67	130	130	130	130
99	127	3	244	130	130	130	130
77	24	11	130	130	130	130	130
166	183	27	130	130	130	130	130
209	82	130	130	130	130	130	130

Table 7.1: Edge-block pixel values

937	-73	31	98	111	-24	-131	-67
-32	-20	-28	-29	17	-42	-57	42
28	17	-45	-16	88	53	31	98
-52	-109	-157	-49	118	46	-77	-30
6	-18	-95	-89	19	3	-68	-32
-34	18	41	-47	-79	-84	-94	-50
-59	-2	42	-39	-34	33	16	-18
34	71	50	-18	-1	29	-42	-87

Table 7.2: Edge-block's DCT coefficients

234	-18	8	25	28	-6	-33	-17
-8	-5	-7	-7	4	-11	-14	11
7	4	-11	-3	16	8	4	8
-13	-27	-34	-10	19	6	-8	0
2	-5	-18	-15	3	0	-6	0
-9	5	6	-6	-9	-8	-7	-3
15	0	5	-4	-3	0	0	0
9	18	4	0	0	0	0	-3

Table 7.3: Coded Coefficients with Maximum Threshold Elevation = 5

elevation model. Specifically, the base thresholds obtained in Chapter 4 are elevated by the texture elevation model described in Chapter 6. The locally adaptive quantization matrix $Q_{m,n}$ is obtained as twice the product of the two. The coded coefficients $I\_coded_{m,n}$ shown in Table 7.3 are simply : $I\_coded_{m,n} = \text{round}\,(I_{m,n}/Q_{m,n})$. In this example, the elevation model used is extremely aggressive with a maximum elevation factor of 5. In this example, where even the original coefficients are high, we still manage to zero-out 11 coefficients. A quick comparison between the DCT coefficients in Table 7.2 and the luminance base weights in Table 4.2 shows that, if we use the base thresholds and no elevation, we can zero-out only 3 coefficients.

From the coded coefficients in Table 7.3, the reconstructed image block can be obtained through normalization and the inverse DCT transform (Equation 5.3). The recontructed pixels are shown in Table 7.4, and the absolute values of the pixels' differences are in Table 7.5. The elevation model demonstrates well its accuracy and local adaptibility. The left section of the block is overcoded since that is where all of the texture energy located in space domain. In the right, there are not much

13	86	33	212	219	0	131	130
191	30	68	113	217	135	138	128
198	241	46	14	25	146	108	134
39	220	30	82	122	115	145	126
106	114	19	226	145	135	123	133
72	37	5	129	135	108	142	127
173	175	26	136	116	148	121	129
206	85	125	132	132	128	131	131

Table 7.4: Equivalent Reconstructed Image Block Pixels for Max Elevation = 5

1	3	10	19	17	7	1	0
2	3	8	22	23	5	8	2
2	11	9	9	6	16	22	4
3	0	5	15	8	15	15	4
7	13	16	18	15	5	7	3
5	13	6	1	5	22	12	3
7	8	1	6	14	18	9	1
3	3	5	2	2	2	1	1

Table 7.5: Magnitude of Error in Space Domain

234	-10	4	14	16	-3	-16	-8
-5	-3	-4	-5	3	-8	-10	7
4	3	-8	-2	13	8	4	14
-7	-22	-26	-7	15	5	-9	-3
1	-4	-16	-13	2	0	-6	-3
-4	3	6	-7	-10	-8	-9	-4
-6	0	5	-5	-4	3	1	-1
3	8	6	-2	0	3	-3	-6

Table 7.6: Coded DCT coefficients with No Threshold Elevation

noise introduced to the pixels far away from the edge. However, near the edge, we can notice that there is serious error spreading from the left heavily textured region. Differences of 22, 18, or 15 of pixel values in the sensitive mid-grey level of the HVS can cause serious degradation in the reconstructed image quality.

A question arises for the curious: what would have happened if there was no threshold elevation? With the quantization matrix entries set to be twice the base weights, the resulting coded DCT coefficients are shown in Table 7.6.

The reconstructed pixels, with no threshold elevation, are shown in Table 7.7, and the absolute value of the pixels' differences in the space domain are shown in Table 7.8.

Since the base thresholds are obtained image-independently, the base threshold model does not take advantage of the heavy texture in the left region of the edge-block. In this case the model codes both regions the same way which results in the same amount of error in both (see Table 7.8). With threshold elevation, much more error is injected into the textured region as expected.

10	90	22	235	203	2	134	128
191	27	78	90	242	131	128	128
202	228	58	23	14	129	128	126
38	222	35	67	128	130	129	127
102	127	7	242	130	132	129	133
78	23	13	127	136	127	129	128
170	179	26	135	125	131	130	132
207	80	132	130	134	131	129	129

Table 7.7: Reconstructed Pixels with No Threshold Elevation

2	1	1	4	1	5	4	2
2	6	2	1	2	1	2	2
2	2	3	0	5	1	2	4
4	2	0	0	2	0	1	3
3	0	4	2	0	2	1	3
1	1	2	3	6	3	1	2
4	4	1	5	5	1	0	2
2	2	2	0	4	1	1	1

Table 7.8: Error in Space Domain with No Threshold Elevation

7.2 Classification Methods

The methods presented next are for detecting edge-blocks. They are designed to discriminate textured blocks based on whether the texture is either structured (edge-blocks), or unstructured. The problem is more complicated then the one-dimentional switching from short block to long block to prevent pre-echoing in perceptual audio coding. The difficulty comes from a very basic question: what exactly is texture? (see Chapter 2). However, we can follow a similar approach — breaking the analysis block into sub-blocks.

7.2.1 Over-Under Method

The 8x8 pixel block is broken up into 16 2x2 sub-blocks. In each sub-block, the variance of the pixels is calculated:

$$\text{variance} = \sum_{j=0}^{1}\sum_{k=0}^{1}(i_{j,k} - average)^2 \tag{7.1}$$

where

$$\text{average} = \frac{1}{4} \sum_{j=0}^{1} \sum_{k=0}^{1} i_{j,k} \,. \tag{7.2}$$

If a sub-block's variance is over some high texture energy threshold, that sub-block is labeled as an *over*. Similarly, if its variance is lower than the no texture energy threshold, it is labeled an *under*. Otherwise, the sub-block is labeled a *between*. A block is labeled edgy when the number of *over* sub-blocks and the number of *under* sub-blocks are close, and there are not many *betweens*. The two energy thresholds, as well as the other parameters of the decision rules, are set through numerous experiments. A simple test of the detection model's effectiveness is to zero-out the pixels of the suspected edgy blocks. On display, all of these blocks will be black. We can then subjectively estimate the detection rate, as well as the false alarm rate of the model. Various parameter values can then be tested to increase the model's effectiveness.

The over-under approach does not seem to perform well. If the detection rate is high, then the false alarm rate is also high. If the parameters are reset such that the false alarm rate is low, the detection rate is also low. There are just too many parameters, and it is almost impossible to find a good combination to keep the false alarm rate low and the detection rate high.

7.2.2 Variance Ratio Method

In this approach, the pixel block is also divided into 2x2 sub-blocks. The variance of each sub-block is calculated using Equations 7.1 and 7.2.

Among the computed variances, the ratio of the maximum value and the minimum non-zero one, $\frac{\text{max\_variance}}{\text{min\_variance}}$, is used to decide whether or not the block is edgy. If the ratio is large, it means that certain parts of the image block have significantly more texture energy than others. Also, since we are only worrying about noise spreading in the very "clean" region of a pixel block, the minimum variance has to be under a certain low energy threshold for the block to be labeled an edge-block. An empirical value for the ratio threshold is 25. A typical value used for the low variance threshold is 15. The model seems to perform well with these parameter choices.

This method for edge-block detection still has flaws. One of the most obvious is its sensitivity in extreme cases. In the case of an image block with 15 textured sub-blocks and 1 clean sub-block, or 1 textured and 15 clean sub-blocks, the model will label the block edgy. We can prevent this false alarm by assigning two more parameters: over and under, as in the over-under method. However, we choose not to further increase the complexity of the model since this situation rarely occurs in practice.

7.3 Coding of Edge-blocks

The coding of edge-blocks is still a puzzling question. In the time (or space) domain, it is obvious that which parts of an image are smooth, and which are textured. However, when the pixels are transformed to its frequency domain, the summations of the pixel values projected onto the cosine basis totally destroy the pixels' correlation. As observed from the example in Section 7.1, while it is clear in the space domain that the left part of the block contains very high texture energy, it is unclear in the DCT domain which coefficients contribute to that texture. The threshold elevation model does a good job of injecting most of the noise into the textured region. For this high threshold elevation case, the problem of noise spreading into the uniform region is unavoidable. For now, the only solution is to detect the edge-blocks, and use lower elevation factors on them. One can even be more conservative by just using the base thresholds for these edge-blocks.

Another solution to the noise spreading problem is to process the image with a finer space resolution. The frequency resolution, however, will suffer. Furthermore, we would like to preserve the standard 8x8 DCT decomposition. In this case, a finer resolution, meaning using a smaller size for analysis blocks, can still be achieved by using a DCT with overlapped analysis blocks — the Extended Lapped Transform (ELT) [11]. The elevation factor of a particular coefficient is the minimum of the factors computed from the analysis blocks to which the coefficient belongs. This will significantly improve the accuracy of the elevation model. However, the computational complexity of the coding process also increases accordingly. With an overlapping factor of 2, the cost of coding an image increases approximately four times. This idea needs a more in-depth investigation.

Chapter 8

Subjective Tests and Results

Subjective tests are an essential part of the project. Still images and sequences of different, but known, levels of coding difficulty were tested using the new model, as well as the old ones for performance comparison purposes. Another important contribution of subjective testing experiments was to fine-tune the new masking threshold model's parameters as previously mentioned.

8.1 Set-up

The experiments were carried out on an 8x8 DCT decomposition of images. The test images at AT&T Bell Laboratory are digital images in CIF format with size 360x240 for the luminance channel, and 180x120 for the chrominance channels. All of the pixels have an 8 bit resolution (pixel values ranging from 0 to 255). When displayed, the images are interpolated to be twice the storage dimensions.

Each test image channel was divided into 8x8-pixel blocks. Each pixel block was then transformed to its equivalent DCT. The pre-computed base threshold values were multiplied by the elevation factors computed from the locally adaptive texture elevation model to obtain the final threshold values. Next, each input image was corrupted with the maximum amount of noise allowed by the model, i.e. specifically, each DCT coefficient in each analysis block was randomly either subtracted or added by its computed threshold value:

$$I\_coded_{k,l,m,n} = I_{k,l,m,n} \pm threshold_{k,l,m,n} \, , \tag{8.1}$$

where

$$threshold_{k,l,m,n} = base\_threshold_{k,m,n} \, . \, elevation\_factor_{k,l,m,n} . \tag{8.2}$$

For the above formulae and also for rest of the chapter, k refers to the channel index, l refers the block index, and m and n are the indices of the spatial DCT frequencies. Notice that the base thresholds are not block-dependent; they do not have index l. To obtain the recontructed image pixels, an inverse DCT (Equation 5.3) was performed on the "coded" coefficients $I\_coded_{m,n}$.

This *noise-adding* scheme was used in the early stages of the project. It resulted in huge levels of noise injected to the high frequency DCT coefficients, and it was certainly an overly conservative approximation of the masking threshold model's performance. A more realistic approximation was the *zero-out coefficients* scheme, in which all coefficients below their corresponding thresholds were set to zero. For coefficients that were larger than the thresholds, the noise-adding scheme was applied. Specifically, the "coded" coefficient was obtained as follows:

$$I\_coded_{k,l,m,n} = \begin{cases} 0 \, , & \text{for } |I_{k,l,m,n}| \leq threshold_{k,l,m,n} \\ I_{k,l,m,n} \pm threshold_{k,l,m,n} \, , & \text{otherwise} \end{cases} \tag{8.3}$$

with the thresholds calculated from Equation 8.2

8.2 Subjective Evaluation Tests

Test subjects were invited to subjectively determine if any distortion was perceivable in the resulting reconstructed images. For a standard subjective test, the original image or sequence was always shown first. The original and the coded image or sequence were then loaded onto two high-resolution TV monitors side by side. The subjects were asked to point out which one was the original and which was the coded.

Subjects invited to the numerous subjective tests were mostly members of the Image Group who were experienced and well-trained. They are more sensitive to noise than normal people. The model was fine-tuned until all the subjects could not reliably detect any visual difference between the coded and the original image or sequence.

8.3 Objective Statistics

In perceptual coding, we have two measures of evaluating the performance of our model: one is the subjective measure presented in the previous section, and the other is the objective statistics. The goal is to keep the subjective performance at the perceptually lossless level and then use the objective measures to evalute alternative models.

For each channel of the input image, an 8x8 matrix of average mean-square error for each DCT frequency bin was obtained. For each analysis block, using the zero-out scheme, the block's mean-square error matrix was computed in DCT domain as the square of the difference between the original coefficient and the coded one:

$$mse_{k,l,m,n} = \begin{cases} I^2_{k,l,m,n}\,, & \text{for } |I_{k,l,m,n}| \leq threshold_{k,l,m,n} \\ threshold^2_{k,l,m,n} & \text{otherwise.} \end{cases} \tag{8.4}$$

The average mean-square error matrix for channel k is the summation of all the blocks' mean-square error in that channel normalized by the total number of blocks num_block:

$$average\_mse_{k,m,n} = \frac{1}{num\_block} \sum_{l=1}^{l=num\_block} mse_{k,l,m,n} \tag{8.5}$$

The average mean-square error matrix provides the traditional objective evaluation measure of source coding — the Signal-to-Noise Ratio (SNR). Not only does the SNR show how effective the masking model is, but it also can be used for various demonstration purposes. One popular demonstration had three images displayed side-by-side: the original sequence, the perceptually distortion-free coded sequence, and the original sequence corrupted by uniformly distributed white noise with the same SNR as the perceptually coded sequence.

Besides the mean-square error, the drop percentage is another useful objective statistic. The drop percentage gives an approximation of the compression ratio needed to achieve coding at the perceptually lossless level. For each channel of the coded image or sequence, an 8x8 matrix of dropped coefficients percentage $drop\_percentage_{m,n}$ was kept. Each element of the matrix shows the percentage of how many DCT coefficients in that frequency bin are smaller than the threshold computed at the same frequency (and hence, the coefficient is set to zero):

$$drop\_percentage_{k,m,n} = 100\,\% \cdot \frac{1}{num\_block} \sum_{l=1}^{l=num\_block} drop_{k,l,m,n} \tag{8.6}$$

where

$$drop_{k,l,m,n} = \begin{cases} 1 \,, & \text{for } |I_{k,l,m,n}| \leq threshold_{k,l,m,n} \\ 0 \,, & \text{otherwise} \,, \end{cases} \tag{8.7}$$

with k is the channel index, and l is the block index.

In a similar fashion, an 8x8 average threshold matrix was also obtained:

$$average\_threshold_{k,m,n} = \frac{1}{num\_block} \sum_{l=1}^{l=num\_block} threshold_{k,l,m,n} \tag{8.8}$$

The average threshold values provide a good measure of how the threshold elevation model works in a particular image or sequence. They are also excellent tools for debugging the model's source code.

8.4 Results

The new adaptive perceptual threshold model (APxJPEG) was tested with two other popular image compression models already in use: the JPEG compression standard and the perceptual Johnston-Safranek model (PxJPEG), both described in Chapter 3. 318 still images in the AT&T image database were used to compile this performance comparison statistics. As expected, the adaptive perceptual masking model outperformed JPEG by a large margin. The gain in the bit rate needed for transparent coding was on the order of 10 to 30 %. The race was closer for the two picture-dependent models. In general, the new model had the same or better performance than the Johnston-Safranek model. For images with a lot of directed texture, we got much better performance from the new model thanks to its locally adaptibility. The bit rate savings comparison between APxJPEG and JPEG is depicted in Figure 8-1 The same comparison between APxJPEG's and PxJPEG's performance is shown in Figure 8-2. The complete bit rate saving percentage for each particular image can be found in Appendices B and C. Also included are three *lenna* images: the original image (Figure 8-3), the reconstructed image using JPEG (Figure 8-4), and the reconstructed image using the new adaptive perceptual threshold model as a pre-processor for JPEG (Figure 8-5). The original 512x512 gray-scale image has a bit rate of 8 bits per pixel. The resulting bit rate for the reconstructed JPEG image is 1.026 bits per pixel. The resulting bit rate for the reconstructed APxJPEG image is 0.813 bits per pixel (a 15 % bit rate saving). One can easily verify that both of the reconstructed images were coded at perceptually lossless level.

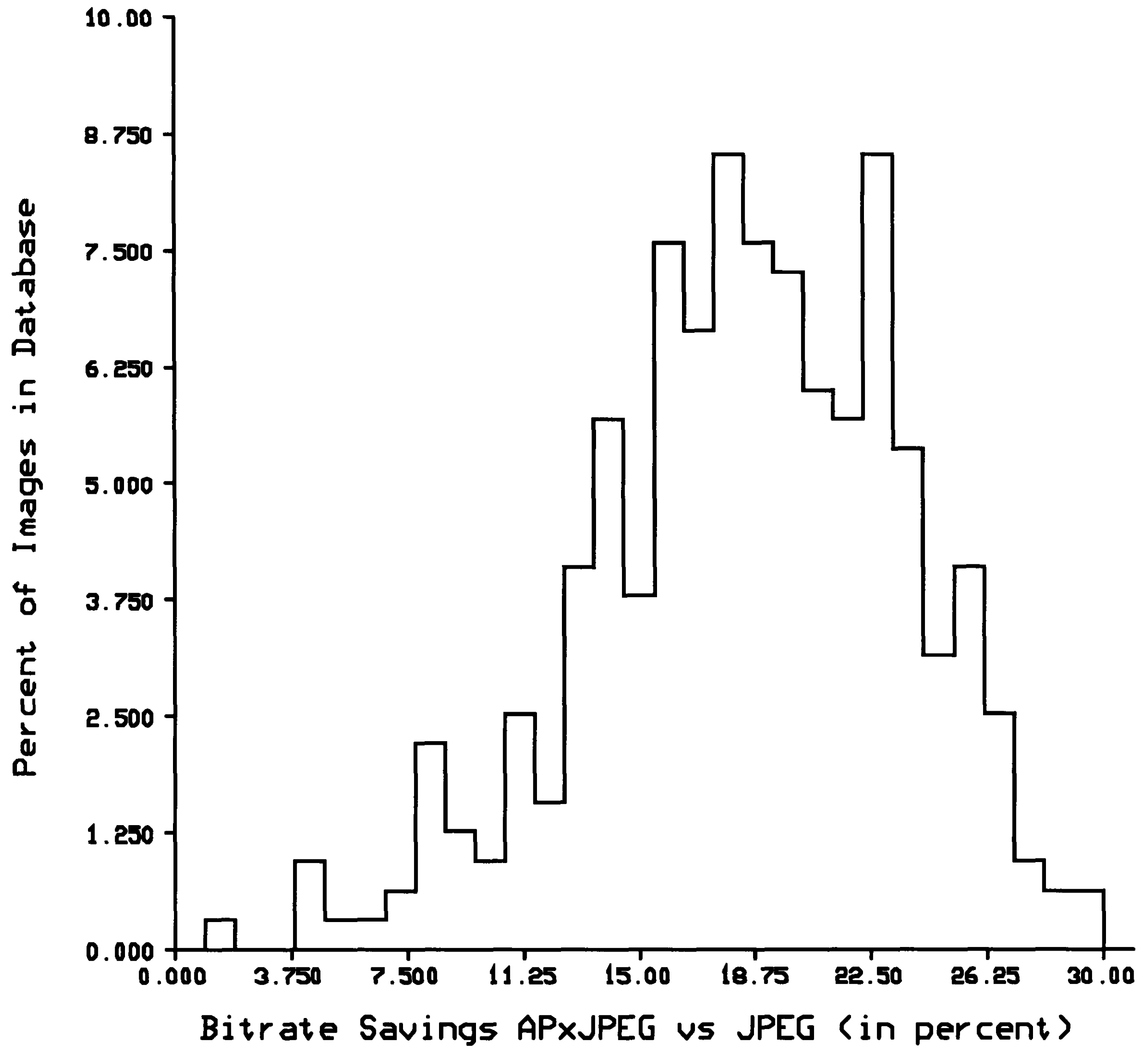

Figure 8-1: Performance Comparison between APxJPEG and JPEG

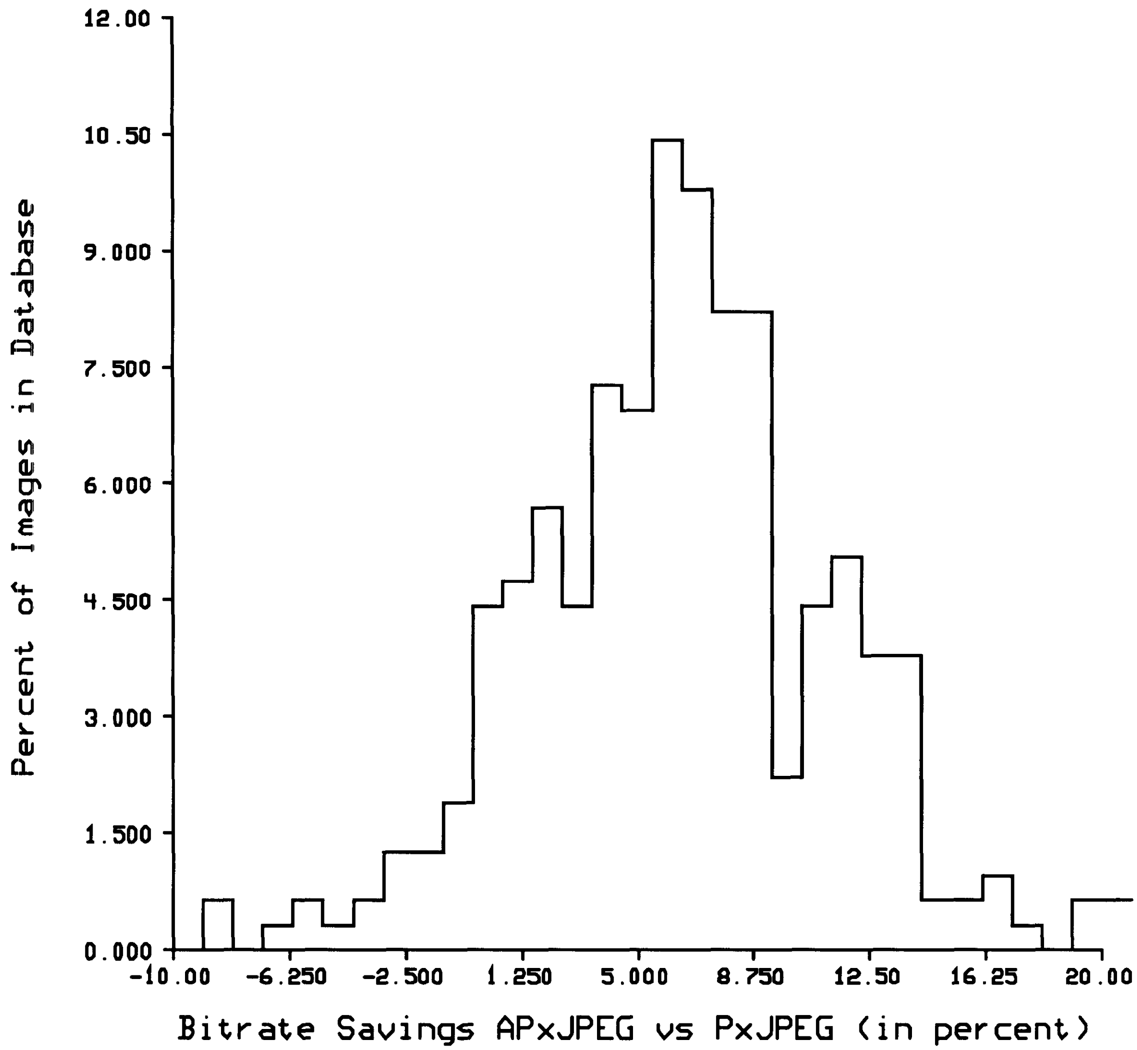

Figure 8-2: Performance Comparison between APxJPEG and PxJPEG

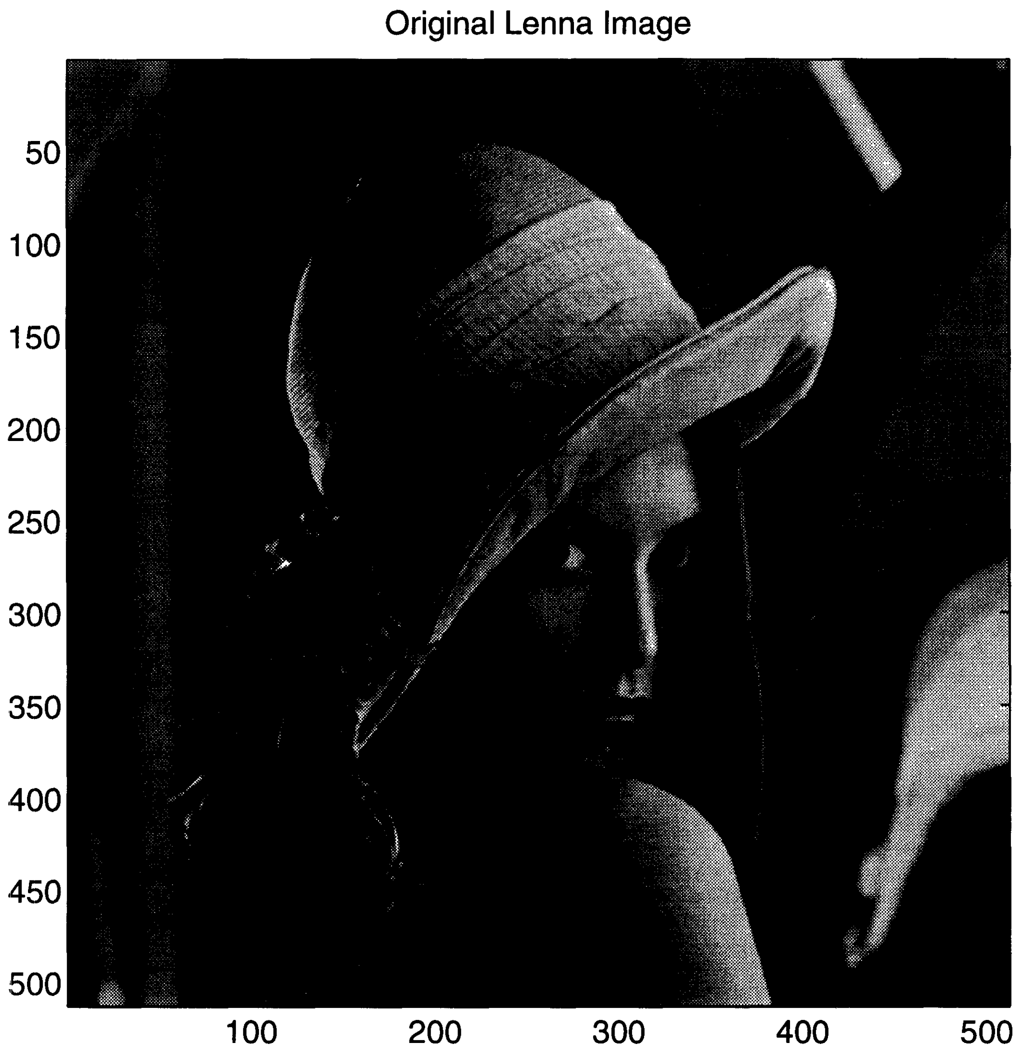

Figure 8-3: Original Image for Reference

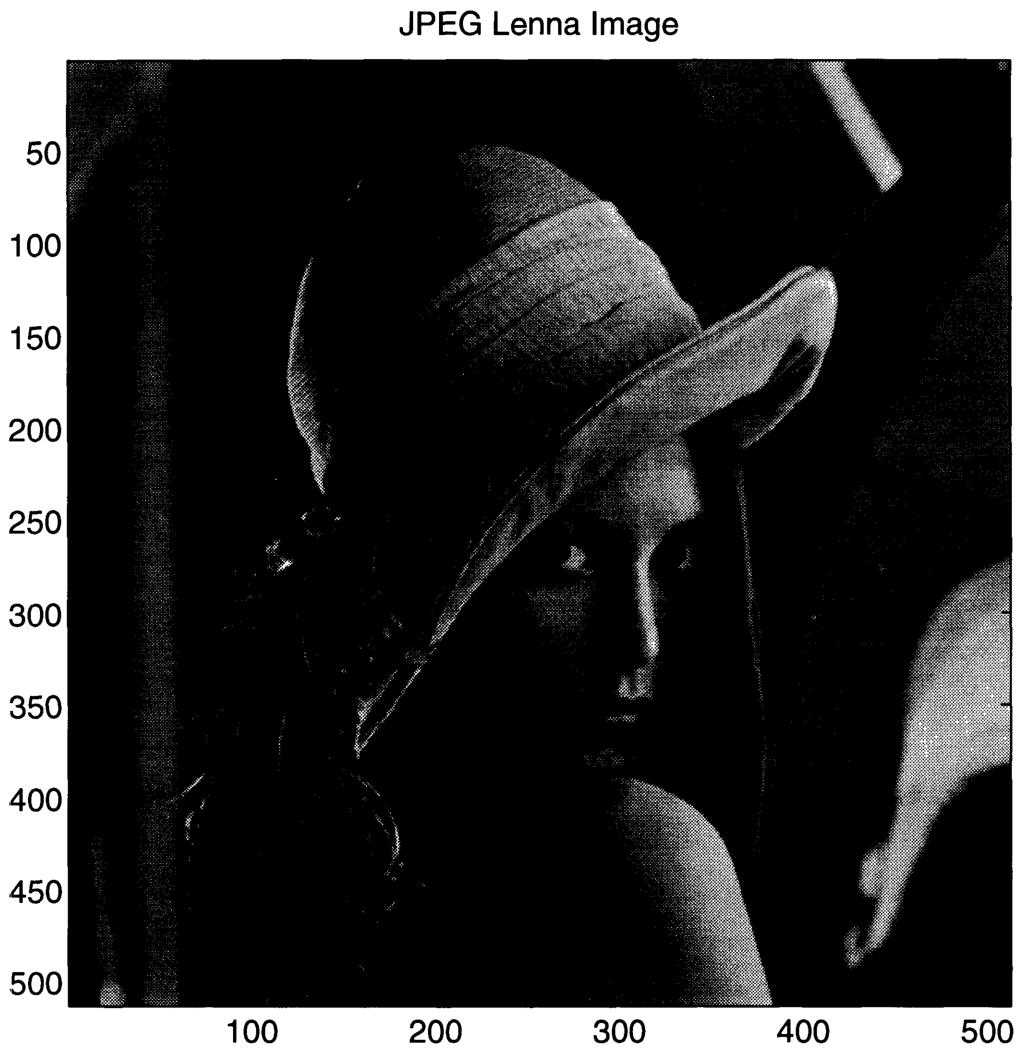

Figure 8-4: Reconstructed JPEG Image coded at 1.026 bits/pixel

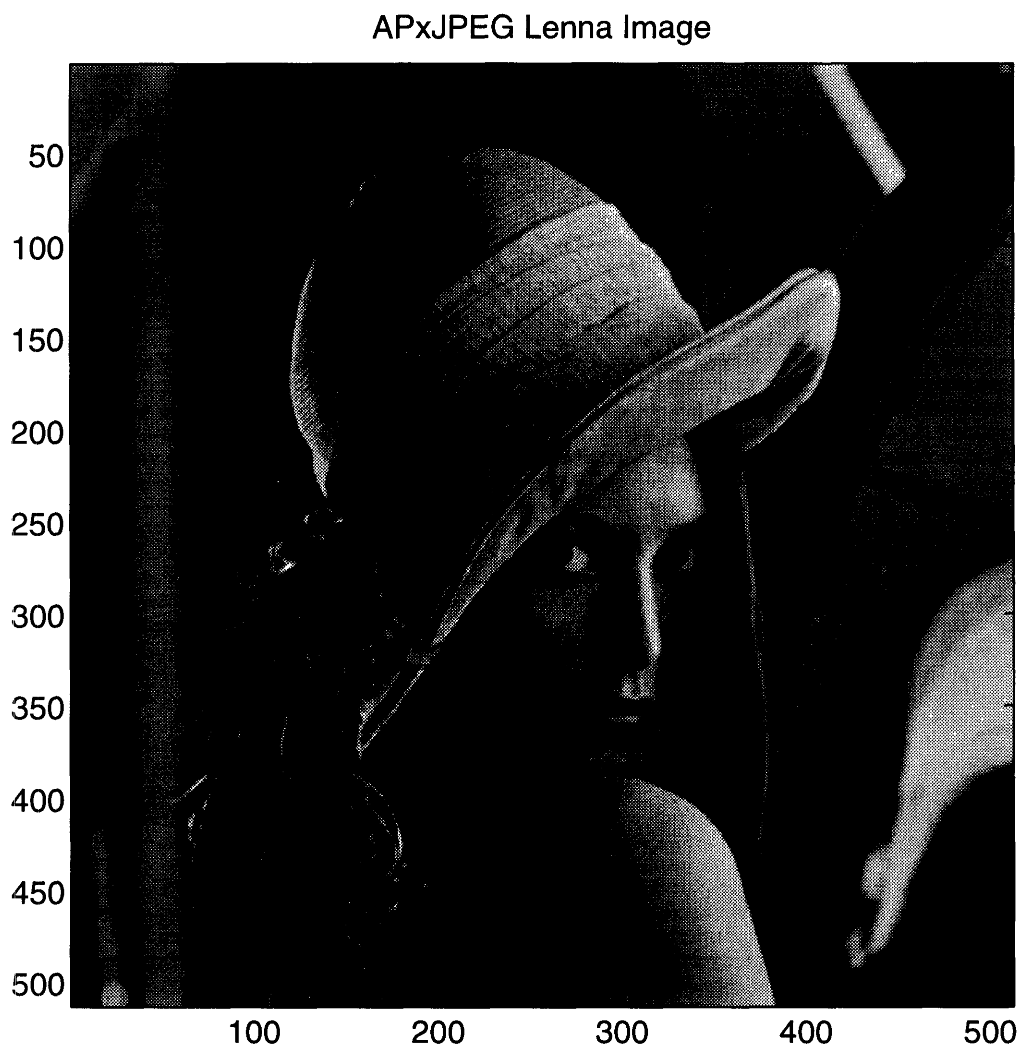

Figure 8-5: Reconstructed APxJPEG Image coded at 0.813 bits/pixel

Chapter 9

Conclusion

In this project, a new locally adaptive perceptual masking threshold model for the human visual system was designed and implemented. The model's development was based on many evident characteristics of the HVS available from numerous psychophysical experiments. The model not only performs much better than the JPEG's standard image-independent perceptually lossless model, but it also out-performs, as expected because of its local adaptibility, AT&T's currently used threshold elevation model developed by Johnston and Safranek [15]. The mapping of the cortex transform's critical bands onto the DCT bins proves to approximate accurately the locality as well as the intensity of the mask. The DCT-cortex mapping is the pivotal basis of the image-dependent texture elevation model.

Despite the success of the project, much more work remains to be done in this area. As we can see from Chapter 7, aggressive elevation in the DCT domain due to the presence of a heavy texture region in an analysis block can cause serious noise spreading to the flat-field region of the same block in space domain. In this case, the noise spread is most vulnerable to detectability. A more robust threshold elevation model that can effectively deal with these edge-blocks needs to be developed. One of the solutions to this problem is to increase the accuracy of our threshold elevation model by extending the masking ideas across DCT blocks. However, this masking-across-DCT-block extension, as discussed in Chapter 7, can be very computationally expensive.

Another important piece totally missing from this perceptual masking threshold model is temporal masking. Although the model was tested with image sequences, this study did not consider temporal masking effects at all. However, we recognize that the human visual system's perception of dynamic noise in image sequences is, in general, very different from its perception of static noise in still images. A full masking model which includes temporal noise masking needs to be studied and applied to coding of image sequences.

Besides the problem of noise spreading and the lack of temporal masking, the threshold elevation model developed in this project is also not robust enough. It was primarily designed and geared to be compatible with DCT-based coders. Its effectiveness when used with other different coder types is doubtful and has not yet been tested.

In short, this project raises more new questions than it resolves. Many aspects of the project need more in-depth investigation. However, it serves as a good building block for the understanding, as well as the advancing of the perceptual image coding field.

Appendix A

DCT-Cortex Overlap Area Matrices

---

binary matrix for cortex band k=0 l=0

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.003981	0.001201	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.201650	0.063120	0.000358	0.000000	0.000000	0.000000	0.000000	0.000000
0.671549	0.292789	0.021787	0.000000	0.000000	0.000000	0.000000	0.000000
0.899004	0.483785	0.096608	0.000954	0.000000	0.000000	0.000000	0.000000
0.822057	0.514997	0.170313	0.013865	0.000000	0.000000	0.000000	0.000000
0.609768	0.416196	0.176334	0.032873	0.000536	0.000000	0.000000	0.000000

binary matrix for cortex band k=0 l=1

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix for cortex band k=0 l=2

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=0 l=3

0.000000 0.000000 0.003981 0.201650 0.671548 0.899004 0.822057 0.609768

0.000000 0.000000 0.001201 0.063119 0.292788 0.483784 0.514997 0.416195

0.000000 0.000000 0.000000 0.000358 0.021787 0.096607 0.170313 0.176334

0.000000 0.000000 0.000000 0.000000 0.000000 0.000954 0.013865 0.032873

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000536

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=0 l=4

0.000000 0.000000 0.002262 0.050175 0.099558 0.088237 0.057264 0.031737

0.000000 0.000011 0.041774 0.325886 0.565647 0.497872 0.335178 0.188772

0.000000 0.001581 0.124940 0.582960 0.931876 0.855549 0.615426 0.355219

0.000000 0.000098 0.074429 0.460742 0.809628 0.851079 0.664670 0.391722

0.000000 0.000000 0.007853 0.171878 0.453653 0.581714 0.495368 0.291054

0.000000 0.000000 0.000003 0.028170 0.169916 0.284505 0.262792 0.144223

0.000000 0.000000 0.000000 0.001010 0.036752 0.094813 0.092761 0.039278

0.000000 0.000000 0.000000 0.000000 0.003020 0.017738 0.016436 0.002742

binary matrix for cortex band k=0 l=5

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000011 0.001581 0.000098 0.000000 0.000000 0.000000 0.000000

0.002262 0.041774 0.124941 0.074429 0.007853 0.000003 0.000000 0.000000

0.050175 0.325886 0.582961 0.460743 0.171879 0.028170 0.001010 0.000000

0.099558 0.565646 0.931875 0.809629 0.453654 0.169917 0.036752 0.003020

0.088236 0.497871 0.855550 0.851079 0.581715 0.284506 0.094813 0.017738

0.057263 0.335177 0.615425 0.664670 0.495369 0.262793 0.092761 0.016436

0.031737 0.188772 0.355219 0.391722 0.291054 0.144223 0.039278 0.002742

binary matrix for cortex band k=1 l=0

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.066070 0.000095 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.589011 0.030599 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.596712	0.100551	0.000143	0.000000	0.000000	0.000000	0.000000	0.000000
0.198945	0.048523	0.000695	0.000000	0.000000	0.000000	0.000000	0.000000
0.003461	0.000305	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=1 l=1

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=1 l=2

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=1 l=3

0.000000	0.066070	0.589011	0.596711	0.198945	0.003461	0.000000	0.000000
0.000000	0.000095	0.030599	0.100551	0.048522	0.000305	0.000000	0.000000
0.000000	0.000000	0.000000	0.000143	0.000695	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=1 l=4

0.000001	0.100496	0.307127	0.151464	0.029949	0.000303	0.000000	0.000000
0.000364	0.305899	0.862663	0.510104	0.093043	0.000213	0.000000	0.000000
0.000000	0.051635	0.375059	0.303556	0.036653	0.000000	0.000000	0.000000
0.000000	0.000242	0.038554	0.039198	0.000803	0.000000	0.000000	0.000000

0.000000	0.000000	0.000261	0.000131	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix for cortex band k=1 l=5

0.000001	0.000364	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.100496	0.305899	0.051635	0.000242	0.000000	0.000000	0.000000	0.000000
0.307127	0.862663	0.375060	0.038554	0.000261	0.000000	0.000000	0.000000
0.151463	0.510103	0.303556	0.039198	0.000131	0.000000	0.000000	0.000000
0.029949	0.093043	0.036653	0.000803	0.000000	0.000000	0.000000	0.000000
0.000303	0.000213	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix for cortex band k=2 l=0

0.015816	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.334752	0.000040	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.065784	0.000199	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix for cortex band k=2 l=1

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix for cortex band k=2 l=2

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=2 l=3

0.015816	0.334752	0.065784	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000040	0.000199	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=2 l=4

0.097028	0.455986	0.031835	0.000000	0.000000	0.000000	0.000000	0.000000
0.012776	0.193956	0.010236	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000112	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=2 l=5

0.097029	0.012776	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.455986	0.193956	0.000112	0.000000	0.000000	0.000000	0.000000	0.000000
0.031835	0.010236	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

binary matrix **for** cortex band k=3 l=0

0.089995	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.018060	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=3 l=1

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=3 l=2

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=3 l=3

0.089995 0.018060 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=3 l=4

0.188985 0.011462 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000036 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=3 l=5

0.188985 0.000036 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.011462 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=4 l=0

0.039365 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=4 l=1

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=4 l=2

0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=4 l=3

0.039365 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=4 l=4

0.065205 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex band k=4 l=5

0.065205 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

binary matrix for cortex base band

0.007209 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

Appendix B

Comparison Between Adaptive Perceptual Threshold Model and JPEG

JPEG image	*APxJPEG image*	*savings in percent*
aelephant.jpg	aelephant.apjpg	25.937414
aelephant2.jpg	aelephant2.apjpg	23.376098
alco.jpg	alco.apjpg	21.948893
anemone1.jpg	anemone1.apjpg	17.613679
anemone2.jpg	anemone2.apjpg	20.139611
anemone3.jpg	anemone3.apjpg	16.897058
angelika.jpg	angelika.apjpg	11.983423
aplcr1.jpg	aplcr1.apjpg	22.374266
appletree.jpg	appletree.apjpg	20.850413
aravind.jpg	aravind.apjpg	18.216714

JPEG image	*APxJPEG image*	*savings in percent*
arizdiv.jpg	arizdiv.apjpg	17.778553
atnight.jpg	atnight.apjpg	14.758003
autumn.jpg	autumn.apjpg	28.633985
balloon.jpg	balloon.apjpg	18.575273
bangalore.jpg	bangalore.apjpg	17.327793
barge.jpg	barge.apjpg	18.089397
barge2.jpg	barge2.apjpg	17.052118
bbear1.jpg	bbear1.apjpg	22.625338
bbear2.jpg	bbear2.apjpg	24.987839
beach1.jpg	beach1.apjpg	13.361764
beach2.jpg	beach2.apjpg	15.950017
beauty.jpg	beauty.apjpg	12.501068
bennevis.jpg	bennevis.apjpg	14.014600
benz.jpg	benz.apjpg	22.482893
bface.jpg	bface.apjpg	16.823244
bface2.jpg	bface2.apjpg	16.751373
bflyfish.jpg	bflyfish.apjpg	15.950166
bird.jpg	bird.apjpg	17.217214
birds.jpg	birds.apjpg	16.928120
blueeyes.jpg	blueeyes.apjpg	10.635309
bluerocks.jpg	bluerocks.apjpg	17.282335
bmfall.jpg	bmfall.apjpg	25.059515
bmfall2.jpg	bmfall2.apjpg	23.275459
bncoal.jpg	bncoal.apjpg	24.811222
boat1.jpg	boat1.apjpg	13.455929

JPEG image	*APxJPEG image*	*savings in percent*
boat3.jpg	boat3.apjpg	13.389240
boat4.jpg	boat4.apjpg	15.183790
boats.jpg	boats.apjpg	15.286319
boattrees.jpg	boattrees.apjpg	18.392541
bosteam.jpg	bosteam.apjpg	13.036086
bowlkid.jpg	bowlkid.apjpg	18.922356
bpelican.jpg	bpelican.apjpg	22.777657
braids.jpg	braids.apjpg	16.661049
brbears3.jpg	brbears3.apjpg	19.636121
bridge.jpg	bridge.apjpg	7.727867
brownthrasher.jpg	brownthrasher.apjpg	14.252335
brunt1.jpg	brunt1.apjpg	18.830988
burchellzebra.jpg	burchellzebra.apjpg	23.257346
bwwarbler.jpg	bwwarbler.apjpg	12.154493
cablecar.jpg	cablecar.apjpg	5.715944
cacol.jpg	cacol.apjpg	18.328592
cacol2.jpg	cacol2.apjpg	18.340735
cactii.jpg	cactii.apjpg	19.451304
caform.jpg	caform.apjpg	22.243409
camelride.jpg	camelride.apjpg	18.686926
cannon.jpg	cannon.apjpg	16.258337
canoe.jpg	canoe.apjpg	21.782540
carbide.jpg	carbide.apjpg	10.911158
carcol.jpg	carcol.apjpg	23.911826
carent.jpg	carent.apjpg	18.713969

JPEG image	*APxJPEG image*	*savings in percent*
cgirl.jpg	cgirl.apjpg	18.771826
chamois.jpg	chamois.apjpg	18.854229
cheetah.jpg	cheetah.apjpg	16.392891
chef.jpg	chef.apjpg	13.648424
cheryl.jpg	cheryl.apjpg	16.670172
chincal1.jpg	chincal1.apjpg	17.650047
chincal2.jpg	chincal2.apjpg	21.592560
chincal3.jpg	chincal3.apjpg	15.988183
clifh2.jpg	clifh2.apjpg	25.479957
clifhb.jpg	clifhb.apjpg	18.082497
clifhb2.jpg	clifhb2.apjpg	18.045901
cloud.jpg	cloud.apjpg	7.636854
cloudleopard.jpg	cloudleopard.apjpg	15.159595
clownfish.jpg	clownfish.apjpg	17.245089
clownfish2.jpg	clownfish2.apjpg	22.598387
clownfish2a.jpg	clownfish2a.apjpg	12.244348
colsky.jpg	colsky.apjpg	13.513723
connel.jpg	connel.apjpg	13.170382
coral.jpg	coral.apjpg	18.133658
coraldetail.jpg	coraldetail.apjpg	16.970363
coralfish.jpg	coralfish.apjpg	19.171521
cougar.jpg	cougar.apjpg	20.672680
cowfish.jpg	cowfish.apjpg	21.760601
cowfish2.jpg	cowfish2.apjpg	18.196800
cranes.jpg	cranes.apjpg	15.376557

JPEG image	*APxJPEG image*	*savings in percent*
crinan.jpg	crinan.apjpg	20.527469
cube.jpg	cube.apjpg	12.404174
cyberbath.jpg	cyberbath.apjpg	25.576851
cyberbridge.jpg	cyberbridge.apjpg	21.685977
cybern.jpg	cybern.apjpg	21.755576
cybwall.jpg	cybwall.apjpg	21.996224
dancers.jpg	dancers.apjpg	17.576163
dancers2.jpg	dancers2.apjpg	19.673307
delwg1.jpg	delwg1.apjpg	17.327963
denvhouse.jpg	denvhouse.apjpg	12.068061
denvrange.jpg	denvrange.apjpg	15.397868
downywood.jpg	downywood.apjpg	14.718023
dragon.jpg	dragon.apjpg	20.090811
dunrobin.jpg	dunrobin.apjpg	21.503113
durango.jpg	durango.apjpg	25.258896
edcas1.jpg	edcas1.apjpg	15.942279
edcas2.jpg	edcas2.apjpg	17.779754
edinwide.jpg	edinwide.apjpg	18.438019
edinwide2.jpg	edinwide2.apjpg	17.750534
elk.jpg	elk.apjpg	22.029632
erieviaduct.jpg	erieviaduct.apjpg	25.114304
f16.jpg	f16.apjpg	13.904496
firegoby.jpg	firegoby.apjpg	13.039327
firehole.jpg	firehole.apjpg	19.925678
firehole1.jpg	firehole1.apjpg	18.971330

JPEG image	*APxJPEG image*	*savings in percent*
firekids.jpg	firekids.apjpg	17.318559
fishgrotto.jpg	fishgrotto.apjpg	4.728929
flow.jpg	flow.apjpg	23.664081
flowerbird.jpg	flowerbird.apjpg	14.496720
foggy.jpg	foggy.apjpg	1.184592
fognelms.jpg	fognelms.apjpg	18.668213
forthbr.jpg	forthbr.apjpg	22.265974
galco.jpg	galco.apjpg	21.944191
gandhi.jpg	gandhi.apjpg	19.363968
generalst.jpg	generalst.apjpg	23.335133
gheron.jpg	gheron.apjpg	15.372410
giraffes.jpg	giraffes.apjpg	15.642646
girl3.jpg	girl3.apjpg	15.878692
giverny.jpg	giverny.apjpg	26.676106
giverny3.jpg	giverny3.apjpg	25.340948
gizaa.jpg	gizaa.apjpg	10.615692
glassfish.jpg	glassfish.apjpg	23.409003
glenfinnan.jpg	glenfinnan.apjpg	22.054587
glorch.jpg	glorch.apjpg	9.755775
goby.jpg	goby.apjpg	17.647859
guanacos.jpg	guanacos.apjpg	19.055930
guncan.jpg	guncan.apjpg	23.423721
hadwall.jpg	hadwall.apjpg	18.454799
hamilnight.jpg	hamilnight.apjpg	15.880024
handbag.jpg	handbag.apjpg	20.215203

JPEG image	*APxJPEG image*	*savings in percent*
hawkfish.jpg	hawkfish.apjpg	21.549733
hbarrack.jpg	hbarrack.apjpg	21.794914
hoatzins.jpg	hoatzins.apjpg	20.084043
housefinch.jpg	housefinch.apjpg	13.453940
househorse.jpg	househorse.apjpg	23.243073
housesparrow.jpg	housesparrow.apjpg	13.733645
housewren.jpg	housewren.apjpg	15.222873
housteads.jpg	housteads.apjpg	13.394086
humbird.jpg	humbird.apjpg	12.547450
hume.jpg	hume.apjpg	15.048593
islay.jpg	islay.apjpg	22.151507
jaguar.jpg	jaguar.apjpg	15.777045
jedcreek.jpg	jedcreek.apjpg	20.873342
jedwater.jpg	jedwater.apjpg	6.676757
jj.jpg	jj.apjpg	13.715144
jlf.jpg	jlf.apjpg	22.886576
jlf1.jpg	jlf1.apjpg	22.777170
jroutine.jpg	jroutine.apjpg	19.052489
jumping.jpg	jumping.apjpg	16.309626
jun1.jpg	jun1.apjpg	24.278401
keiorose.jpg	keiorose.apjpg	20.708261
kew1.jpg	kew1.apjpg	18.556991
kew2.jpg	kew2.apjpg	23.909198
kew3.jpg	kew3.apjpg	22.914034
kew3a.jpg	kew3a.apjpg	21.759062

JPEG image	*APxJPEG image*	*savings in percent*
kingfisher.jpg	kingfisher.apjpg	13.985911
kitty.jpg	kitty.apjpg	18.511569
landsat.jpg	landsat.apjpg	24.702556
lechladeboy.jpg	lechladeboy.apjpg	22.061150
leighonsea.jpg	leighonsea.apjpg	22.434561
lemurs.jpg	lemurs.apjpg	17.337529
lena.jpg	lena.apjpg	15.003331
lily.jpg	lily.apjpg	15.364109
lincas.jpg	lincas.apjpg	23.200561
lincath1.jpg	lincath1.apjpg	22.843993
lincath2.jpg	lincath2.apjpg	7.507972
linespr.jpg	linespr.apjpg	9.277210
lingat.jpg	lingat.apjpg	22.868666
lioncub.jpg	lioncub.apjpg	21.330689
llf.jpg	llf.apjpg	17.219388
lochtay.jpg	lochtay.apjpg	24.517485
lollypop.jpg	lollypop.apjpg	18.899246
londonflwr.jpg	londonflwr.apjpg	20.273855
lonetree.jpg	lonetree.apjpg	22.056785
lupus.jpg	lupus.apjpg	14.469941
lynx.jpg	lynx.apjpg	21.182202
m109.jpg	m109.apjpg	21.862151
macaw.jpg	macaw.apjpg	16.867934
maine1.jpg	maine1.apjpg	13.642234
maine1a.jpg	maine1a.apjpg	10.637335

JPEG image	*APxJPEG image*	*savings in percent*
manatee.jpg	manatee.apjpg	19.251585
mandrill.jpg	mandrill.apjpg	23.514128
mars.jpg	mars.apjpg	4.152466
marsh.jpg	marsh.apjpg	26.093465
marsh2.jpg	marsh2.apjpg	27.755669
marysnow.jpg	marysnow.apjpg	18.710905
mbg.jpg	mbg.apjpg	7.711092
melrose.jpg	melrose.apjpg	20.590121
miamiflwr.jpg	miamiflwr.apjpg	19.505048
midv2.jpg	midv2.apjpg	19.003758
midv3.jpg	midv3.apjpg	20.683426
midv4.jpg	midv4.apjpg	16.479692
midv5.jpg	midv5.apjpg	16.884366
monetcld.jpg	monetcld.apjpg	23.541521
monetcld2.jpg	monetcld2.apjpg	23.639797
moonsky.jpg	moonsky.apjpg	19.151831
moose.jpg	moose.apjpg	26.335323
mushcoral.jpg	mushcoral.apjpg	15.455647
mushcoral2.jpg	mushcoral2.apjpg	15.091491
muskox.jpg	muskox.apjpg	20.365386
nagardome.jpg	nagardome.apjpg	19.881637
nflicker.jpg	nflicker.apjpg	19.742289
njtransit.jpg	njtransit.apjpg	11.288815
nmcloud.jpg	nmcloud.apjpg	7.754137
nmocking.jpg	nmocking.apjpg	10.962585

JPEG image	*APxJPEG image*	*savings in percent*
nmrock.jpg	nmrock.apjpg	21.954203
nmrock2.jpg	nmrock2.apjpg	19.481964
ocrispum.jpg	ocrispum.apjpg	19.708318
opipe.jpg	opipe.apjpg	25.096792
orangutan.jpg	orangutan.apjpg	25.461131
orchid.jpg	orchid.apjpg	12.661055
overland.jpg	overland.apjpg	9.460071
palms.jpg	palms.apjpg	16.155073
panda.jpg	panda.apjpg	20.895739
panda1.jpg	panda1.apjpg	21.104003
peacock.jpg	peacock.apjpg	25.187230
pengnovb.jpg	pengnovb.apjpg	18.504986
penguin.jpg	penguin.apjpg	19.234081
peppers.jpg	peppers.apjpg	15.052010
pheasant.jpg	pheasant.apjpg	13.380100
pinesiskin.jpg	pinesiskin.apjpg	12.258162
pipefish2.jpg	pipefish2.apjpg	18.988012
pipefish3.jpg	pipefish3.apjpg	11.003699
pitts.jpg	pitts.apjpg	15.291357
plane.jpg	plane.apjpg	16.013969
polarbear.jpg	polarbear.apjpg	14.419402
pycarp.jpg	pycarp.apjpg	25.059959
radcotbridge.jpg	radcotbridge.apjpg	18.685973
railcover.jpg	railcover.apjpg	17.177887
rainbow.jpg	rainbow.apjpg	13.651502

JPEG image	*APxJPEG image*	*savings in percent*
rand.jpg	rand.apjpg	8.515232
redbud.jpg	redbud.apjpg	30.222443
redsea.jpg	redsea.apjpg	16.977921
redsweater.jpg	redsweater.apjpg	13.222945
rocks.jpg	rocks.apjpg	19.308048
rooftop.jpg	rooftop.apjpg	18.862166
ruthven.jpg	ruthven.apjpg	19.899327
sailboats.jpg	sailboats.apjpg	16.983849
sailor.jpg	sailor.apjpg	13.123335
scopekids.jpg	scopekids.apjpg	15.813336
scotlet.jpg	scotlet.apjpg	20.589972
sculpture.jpg	sculpture.apjpg	11.179207
sealions.jpg	sealions.apjpg	22.711761
seaturtle.jpg	seaturtle.apjpg	21.614687
seaturtle2.jpg	seaturtle2.apjpg	21.253024
sfhouse.jpg	sfhouse.apjpg	15.788086
shanibaby.jpg	shanibaby.apjpg	10.979312
shark.jpg	shark.apjpg	21.493906
shberry.jpg	shberry.apjpg	21.453398
ship.jpg	ship.apjpg	7.986404
sixmts.jpg	sixmts.apjpg	14.968219
skull.jpg	skull.apjpg	7.341080
skye.jpg	skye.apjpg	12.309977
skye1.jpg	skye1.apjpg	4.450082
snowtree.jpg	snowtree.apjpg	26.198873

JPEG image	*APxJPEG image*	*savings in percent*
snowtree2.jpg	snowtree2.apjpg	27.500391
softcoral.jpg	softcoral.apjpg	22.220488
soowinter.jpg	soowinter.apjpg	18.914500
sphere.jpg	sphere.apjpg	4.024441
splash.jpg	splash.apjpg	10.263445
stowe.jpg	stowe.apjpg	8.319279
suilven.jpg	suilven.apjpg	17.785493
sunset2.jpg	sunset2.apjpg	12.612943
suntree.jpg	suntree.apjpg	26.595378
surrey.jpg	surrey.apjpg	18.431579
swanmaster.jpg	swanmaster.apjpg	14.537410
swingset.jpg	swingset.apjpg	14.234373
syonheron.jpg	syonheron.apjpg	24.900644
taiwansisters.jpg	taiwansisters.apjpg	16.041993
taiwantower.jpg	taiwantower.apjpg	16.322283
tajmahal.jpg	tajmahal.apjpg	16.945276
tarababy.jpg	tarababy.apjpg	12.370033
telescope.jpg	telescope.apjpg	14.170460
tfrog.jpg	tfrog.apjpg	16.023481
thamesbarrier.jpg	thamesbarrier.apjpg	12.231513
thamescover.jpg	thamescover.apjpg	19.769330
thamescover1.jpg	thamescover1.apjpg	19.717251
thamescover2.jpg	thamescover2.apjpg	16.268292
thamescover3.jpg	thamescover3.apjpg	18.147293
thebruce.jpg	thebruce.apjpg	13.477451

JPEG image	*APxJPEG image*	*savings in percent*
till1.jpg	till1.apjpg	15.463503
tm.jpg	tm.apjpg	21.949700
tobermory.jpg	tobermory.apjpg	24.143109
tourist.jpg	tourist.apjpg	20.144869
toys.jpg	toys.apjpg	15.519200
tree.jpg	tree.apjpg	18.086605
treecoral.jpg	treecoral.apjpg	24.044189
trossachs.jpg	trossachs.apjpg	20.531181
tshell.jpg	tshell.apjpg	24.963292
tudor.jpg	tudor.apjpg	21.931340
turkscap.jpg	turkscap.apjpg	21.404880
twees.jpg	twees.apjpg	16.032349
twokids.jpg	twokids.apjpg	17.249995
twokids2.jpg	twokids2.apjpg	15.110038
vball1.jpg	vball1.apjpg	14.922209
vball2.jpg	vball2.apjpg	15.703239
vball3.jpg	vball3.apjpg	16.847371
wcloud.jpg	wcloud.apjpg	11.379444
wcpas2.jpg	wcpas2.apjpg	28.339627
wcpass.jpg	wcpass.apjpg	24.908943
wdw.jpg	wdw.apjpg	9.319569
webleaves.jpg	webleaves.apjpg	19.742838
weed.jpg	weed.apjpg	21.112850
weed2.jpg	weed2.apjpg	20.669769
wineshotel.jpg	wineshotel.apjpg	18.434316

JPEG image	*APxJPEG image*	*savings in percent*
winter87.jpg	winter87.apjpg	25.487933
wintergrip.jpg	wintergrip.apjpg	16.912638
woodthrush.jpg	woodthrush.apjpg	14.192450
world.jpg	world.apjpg	8.496612
yard.jpg	yard.apjpg	24.228248
zebras.jpg	zebras.apjpg	22.100729
zoosheep.jpg	zoosheep.apjpg	20.799392

Appendix C

Comparison Between Adaptive Perceptual Threshold Model and Johnston- Safranek Model

PxJPEG	*APxJPEG image*	*savings in percent*
aelephant.pjpg	aelephant.apjpg	3.547625
aelephant2.pjpg	aelephant2.apjpg	11.947121
alco.pjpg	alco.apjpg	8.322526
anemone1.pjpg	anemone1.apjpg	5.693921
anemone2.pjpg	anemone2.apjpg	5.496196
anemone3.pjpg	anemone3.apjpg	5.104827
angelika.pjpg	angelika.apjpg	0.359097
aplcr1.pjpg	aplcr1.apjpg	5.214909
appletree.pjpg	appletree.apjpg	5.941345
aravind.pjpg	aravind.apjpg	5.990814

PxJPEG	*APxJPEG image*	*savings in percent*
arizdiv.pjpg	arizdiv.apjpg	4.237956
atnight.pjpg	atnight.apjpg	5.854530
autumn.pjpg	autumn.apjpg	20.976329
balloon.pjpg	balloon.apjpg	13.220882
bangalore.pjpg	bangalore.apjpg	8.460686
barge.pjpg	barge.apjpg	6.835809
barge2.pjpg	barge2.apjpg	6.865525
bbear1.pjpg	bbear1.apjpg	4.489245
bbear2.pjpg	bbear2.apjpg	2.094186
beach1.pjpg	beach1.apjpg	3.052658
beach2.pjpg	beach2.apjpg	5.049860
beauty.pjpg	beauty.apjpg	5.485900
bennevis.pjpg	bennevis.apjpg	1.666373
benz.pjpg	benz.apjpg	10.901569
bface.pjpg	bface.apjpg	4.833712
bface2.pjpg	bface2.apjpg	5.128972
bflyfish.pjpg	bflyfish.apjpg	2.193114
bird.pjpg	bird.apjpg	6.937651
birds.pjpg	birds.apjpg	4.665832
blueeyes.pjpg	blueeyes.apjpg	-1.349763
bluerocks.pjpg	bluerocks.apjpg	5.525364
bmfall.pjpg	bmfall.apjpg	11.311873
bmfall2.pjpg	bmfall2.apjpg	10.572251
bncoal.pjpg	bncoal.apjpg	12.113525

PxJPEG	*APxJPEG image*	*savings in percent*
boat1.pjpg	boat1.apjpg	5.394323
boat3.pjpg	boat3.apjpg	5.308356
boat4.pjpg	boat4.apjpg	8.043388
boats.pjpg	boats.apjpg	4.532311
boattrees.pjpg	boattrees.apjpg	5.356961
bosteam.pjpg	bosteam.apjpg	0.946098
bowlkid.pjpg	bowlkid.apjpg	6.993730
bpelican.pjpg	bpelican.apjpg	7.895821
braids.pjpg	braids.apjpg	7.347237
brbears3.pjpg	brbears3.apjpg	6.475461
bridge.pjpg	bridge.apjpg	1.746951
brownthrasher.pjpg	brownthrasher.apjpg	2.628070
brunt1.pjpg	brunt1.apjpg	5.127528
burchellzebra.pjpg	burchellzebra.apjpg	11.628424
bwwarbler.pjpg	bwwarbler.apjpg	-0.646456
cablecar.pjpg	cablecar.apjpg	-2.188825
cacol.pjpg	cacol.apjpg	7.668620
cacol2.pjpg	cacol2.apjpg	5.242532
cactii.pjpg	cactii.apjpg	8.482199
caform.pjpg	caform.apjpg	6.439390
camelride.pjpg	camelride.apjpg	7.908502
cannon.pjpg	cannon.apjpg	3.260491
canoe.pjpg	canoe.apjpg	8.695138
carbide.pjpg	carbide.apjpg	1.182259
carcol.pjpg	carcol.apjpg	8.434260
carent.pjpg	carent.apjpg	5.426691

PxJPEG	*APxJPEG image*	*savings in percent*
cgirl.pjpg	cgirl.apjpg	7.602362
chamois.pjpg	chamois.apjpg	7.490654
cheetah.pjpg	cheetah.apjpg	0.017384
chef.pjpg	chef.apjpg	1.809555
cheryl.pjpg	cheryl.apjpg	6.490043
chincal1.pjpg	chincal1.apjpg	-5.640289
chincal2.pjpg	chincal2.apjpg	2.917057
chincal3.pjpg	chincal3.apjpg	-1.753248
clifh2.pjpg	clifh2.apjpg	7.616217
clifhb.pjpg	clifhb.apjpg	1.066844
clifhb2.pjpg	clifhb2.apjpg	-4.720654
cloud.pjpg	cloud.apjpg	-6.543314
cloudleopard.pjpg	cloudleopard.apjpg	2.142002
clownfish.pjpg	clownfish.apjpg	6.861512
clownfish2.pjpg	clownfish2.apjpg	9.997267
clownfish2a.pjpg	clownfish2a.apjpg	2.218370
colsky.pjpg	colsky.apjpg	8.355048
connel.pjpg	connel.apjpg	4.057555
coral.pjpg	coral.apjpg	7.971923
coraldetail.pjpg	coraldetail.apjpg	5.335881
coralfish.pjpg	coralfish.apjpg	2.879716
cougar.pjpg	cougar.apjpg	7.173478
cowfish.pjpg	cowfish.apjpg	11.751834
cowfish2.pjpg	cowfish2.apjpg	10.614055
cranes.pjpg	cranes.apjpg	6.769609

PxJPEG	*APxJPEG image*	*savings in percent*
crinan.pjpg	crinan.apjpg	7.400517
cube.pjpg	cube.apjpg	1.941900
cyberbath.pjpg	cyberbath.apjpg	10.526216
cyberbridge.pjpg	cyberbridge.apjpg	7.248385
cybern.pjpg	cybern.apjpg	7.544098
cybwall.pjpg	cybwall.apjpg	7.681992
dancers.pjpg	dancers.apjpg	6.644735
dancers2.pjpg	dancers2.apjpg	6.776748
delwg1.pjpg	delwg1.apjpg	5.353195
denvhouse.pjpg	denvhouse.apjpg	-2.233517
denvrange.pjpg	denvrange.apjpg	3.507172
downywood.pjpg	downywood.apjpg	2.712845
dragon.pjpg	dragon.apjpg	12.156541
dunrobin.pjpg	dunrobin.apjpg	6.695841
durango.pjpg	durango.apjpg	10.336789
edcas1.pjpg	edcas1.apjpg	1.270683
edcas2.pjpg	edcas2.apjpg	7.753431
edinwide.pjpg	edinwide.apjpg	-0.523285
edinwide2.pjpg	edinwide2.apjpg	1.209435
elk.pjpg	elk.apjpg	8.221635
erieviaduct.pjpg	erieviaduct.apjpg	13.268022
f16.pjpg	f16.apjpg	0.086194
firegoby.pjpg	firegoby.apjpg	0.366042
firehole.pjpg	firehole.apjpg	5.206308
firehole1.pjpg	firehole1.apjpg	4.877393

PxJPEG	*APxJPEG image*	*savings in percent*
firekids.pjpg	firekids.apjpg	5.853862
fishgrotto.pjpg	fishgrotto.apjpg	-3.683858
flow.pjpg	flow.apjpg	10.857197
flowerbird.pjpg	flowerbird.apjpg	6.642047
foggy.pjpg	foggy.apjpg	-2.154243
fognelms.pjpg	fognelms.apjpg	8.090796
forthbr.pjpg	forthbr.apjpg	11.164734
galco.pjpg	galco.apjpg	8.185555
gandhi.pjpg	gandhi.apjpg	8.034684
generalst.pjpg	generalst.apjpg	7.371742
gheron.pjpg	gheron.apjpg	-0.501486
giraffes.pjpg	giraffes.apjpg	11.362391
girl3.pjpg	girl3.apjpg	-8.766752
giverny.pjpg	giverny.apjpg	14.525311
giverny3.pjpg	giverny3.apjpg	12.903839
gizaa.pjpg	gizaa.apjpg	0.029225
glassfish.pjpg	glassfish.apjpg	12.518786
glenfinnan.pjpg	glenfinnan.apjpg	6.215992
glorch.pjpg	glorch.apjpg	-4.091101
goby.pjpg	goby.apjpg	4.141559
guanacos.pjpg	guanacos.apjpg	4.266794
guncan.pjpg	guncan.apjpg	12.472750
hadwall.pjpg	hadwall.apjpg	4.892719
hamilnight.pjpg	hamilnight.apjpg	7.661226
handbag.pjpg	handbag.apjpg	7.044347

PxJPEG	*APxJPEG image*	*savings in percent*
hawkfish.pjpg	hawkfish.apjpg	9.202068
hbarrack.pjpg	hbarrack.apjpg	9.263757
hoatzins.pjpg	hoatzins.apjpg	4.493224
housefinch.pjpg	housefinch.apjpg	6.021540
househorse.pjpg	househorse.apjpg	9.879551
housesparrow.pjpg	housesparrow.apjpg	3.572157
housewren.pjpg	housewren.apjpg	3.530333
housteads.pjpg	housteads.apjpg	-0.437789
humbird.pjpg	humbird.apjpg	3.702172
hume.pjpg	hume.apjpg	5.533030
islay.pjpg	islay.apjpg	3.372354
jaguar.pjpg	jaguar.apjpg	3.354774
jedcreek.pjpg	jedcreek.apjpg	10.845764
jedwater.pjpg	jedwater.apjpg	-8.957979
jj.pjpg	jj.apjpg	0.731998
jlf.pjpg	jlf.apjpg	18.527757
jlf1.pjpg	jlf1.apjpg	18.444191
jroutine.pjpg	jroutine.apjpg	6.257094
jumping.pjpg	jumping.apjpg	1.321941
jun1.pjpg	jun1.apjpg	10.148423
keiorose.pjpg	keiorose.apjpg	11.454082
kew1.pjpg	kew1.apjpg	5.452851
kew2.pjpg	kew2.apjpg	11.319397
kew3.pjpg	kew3.apjpg	10.965552
kew3a.pjpg	kew3a.apjpg	9.511228

PxJPEG	*APxJPEG image*	*savings in percent*
kingfisher.pjpg	kingfisher.apjpg	2.276683
kitty.pjpg	kitty.apjpg	10.364585
landsat.pjpg	landsat.apjpg	9.628523
lechladeboy.pjpg	lechladeboy.apjpg	10.745773
leighonsea.pjpg	leighonsea.apjpg	2.790901
lemurs.pjpg	lemurs.apjpg	1.586227
lena.pjpg	lena.apjpg	5.273799
lily.pjpg	lily.apjpg	3.963932
lincas.pjpg	lincas.apjpg	7.558814
lincath1.pjpg	lincath1.apjpg	15.314251
lincath2.pjpg	lincath2.apjpg	-2.919644
linespr.pjpg	linespr.apjpg	1.693655
lingat.pjpg	lingat.apjpg	12.331775
lioncub.pjpg	lioncub.apjpg	6.030887
llf.pjpg	llf.apjpg	2.801282
lochtay.pjpg	lochtay.apjpg	10.256743
lollypop.pjpg	lollypop.apjpg	7.773256
londonflwr.pjpg	londonflwr.apjpg	10.491385
lonetree.pjpg	lonetree.apjpg	8.544677
lupus.pjpg	lupus.apjpg	0.492702
lynx.pjpg	lynx.apjpg	5.950436
m109.pjpg	m109.apjpg	8.583278
macaw.pjpg	macaw.apjpg	4.892259
maine1.pjpg	maine1.apjpg	-0.023037
maine1a.pjpg	maine1a.apjpg	0.406018

PxJPEG	*APxJPEG image*	*savings in percent*
manatee.pjpg	manatee.apjpg	5.754405
mandrill.pjpg	mandrill.apjpg	10.701499
mars.pjpg	mars.apjpg	-1.003822
marsh.pjpg	marsh.apjpg	12.702289
marsh2.pjpg	marsh2.apjpg	16.152553
marysnow.pjpg	marysnow.apjpg	5.568978
mbg.pjpg	mbg.apjpg	0.016582
melrose.pjpg	melrose.apjpg	9.844620
miamiflwr.pjpg	miamiflwr.apjpg	9.681772
midv2.pjpg	midv2.apjpg	6.100100
midv3.pjpg	midv3.apjpg	8.319828
midv4.pjpg	midv4.apjpg	3.526044
midv5.pjpg	midv5.apjpg	6.917512
monetcld.pjpg	monetcld.apjpg	9.153924
monetcld2.pjpg	monetcld2.apjpg	8.278782
moonsky.pjpg	moonsky.apjpg	15.705514
moose.pjpg	moose.apjpg	12.721872
mushcoral.pjpg	mushcoral.apjpg	2.429592
mushcoral2.pjpg	mushcoral2.apjpg	3.395472
muskox.pjpg	muskox.apjpg	5.401741
nagardome.pjpg	nagardome.apjpg	8.436143
nflicker.pjpg	nflicker.apjpg	6.659106
njtransit.pjpg	njtransit.apjpg	2.475900
nmcloud.pjpg	nmcloud.apjpg	-3.019582
nmocking.pjpg	nmocking.apjpg	0.435060

PxJPEG	*APxJPEG image*	*savings in percent*
nmrock.pjpg	nmrock.apjpg	8.164593
nmrock2.pjpg	nmrock2.apjpg	0.949114
ocrispum.pjpg	ocrispum.apjpg	3.571392
opipe.pjpg	opipe.apjpg	11.265521
orangutan.pjpg	orangutan.apjpg	12.467088
orchid.pjpg	orchid.apjpg	1.845768
overland.pjpg	overland.apjpg	0.642698
palms.pjpg	palms.apjpg	7.671583
panda.pjpg	panda.apjpg	1.317081
panda1.pjpg	panda1.apjpg	5.994539
peacock.pjpg	peacock.apjpg	16.405053
pengnovb.pjpg	pengnovb.apjpg	7.849453
penguin.pjpg	penguin.apjpg	6.404345
peppers.pjpg	peppers.apjpg	4.277316
pheasant.pjpg	pheasant.apjpg	-2.570283
pinesiskin.pjpg	pinesiskin.apjpg	2.794100
pipefish2.pjpg	pipefish2.apjpg	5.137531
pipefish3.pjpg	pipefish3.apjpg	-0.716316
pitts.pjpg	pitts.apjpg	8.065835
plane.pjpg	plane.apjpg	7.969593
polarbear.pjpg	polarbear.apjpg	7.614578
pycarp.pjpg	pycarp.apjpg	13.313195
radcotbridge.pjpg	radcotbridge.apjpg	5.581491
railcover.pjpg	railcover.apjpg	5.144797
rainbow.pjpg	rainbow.apjpg	4.736172

PxJPEG	*APxJPEG image*	*savings in percent*
rand.pjpg	rand.apjpg	-0.256386
redbud.pjpg	redbud.apjpg	20.393611
redsea.pjpg	redsea.apjpg	4.564212
redsweater.pjpg	redsweater.apjpg	4.164408
rocks.pjpg	rocks.apjpg	6.454761
rooftop.pjpg	rooftop.apjpg	7.028011
ruthven.pjpg	ruthven.apjpg	1.756343
sailboats.pjpg	sailboats.apjpg	4.562212
sailor.pjpg	sailor.apjpg	1.345817
scopekids.pjpg	scopekids.apjpg	4.335482
scotlet.pjpg	scotlet.apjpg	3.531321
sculpture.pjpg	sculpture.apjpg	3.490639
sealions.pjpg	sealions.apjpg	8.461008
seaturtle.pjpg	seaturtle.apjpg	12.260906
seaturtle2.pjpg	seaturtle2.apjpg	11.847343
sfhouse.pjpg	sfhouse.apjpg	4.088487
shanibaby.pjpg	shanibaby.apjpg	0.455603
shark.pjpg	shark.apjpg	12.781818
shberry.pjpg	shberry.apjpg	11.294882
ship.pjpg	ship.apjpg	-0.145469
sixmts.pjpg	sixmts.apjpg	0.362833
skull.pjpg	skull.apjpg	0.283147
skye.pjpg	skye.apjpg	0.101840
skye1.pjpg	skye1.apjpg	-5.944778
snowtree.pjpg	snowtree.apjpg	12.094573

PxJPEG	*APxJPEG image*	*savings in percent*
snowtree2.pjpg	snowtree2.apjpg	13.363265
softcoral.pjpg	softcoral.apjpg	10.409519
soowinter.pjpg	soowinter.apjpg	5.261819
sphere.pjpg	sphere.apjpg	-0.807882
splash.pjpg	splash.apjpg	4.002154
stowe.pjpg	stowe.apjpg	-2.670261
suilven.pjpg	suilven.apjpg	3.558746
sunset2.pjpg	sunset2.apjpg	3.699787
suntree.pjpg	suntree.apjpg	12.923232
surrey.pjpg	surrey.apjpg	6.310225
swanmaster.pjpg	swanmaster.apjpg	3.199945
swingset.pjpg	swingset.apjpg	3.704749
syonheron.pjpg	syonheron.apjpg	12.878478
taiwansisters.pjpg	taiwansisters.apjpg	2.136664
taiwantower.pjpg	taiwantower.apjpg	1.927902
tajmahal.pjpg	tajmahal.apjpg	3.328111
tarababy.pjpg	tarababy.apjpg	-0.451326
telescope.pjpg	telescope.apjpg	5.427575
tfrog.pjpg	tfrog.apjpg	4.298211
thamesbarrier.pjpg	thamesbarrier.apjpg	4.228386
thamescover.pjpg	thamescover.apjpg	7.376517
thamescover1.pjpg	thamescover1.apjpg	6.794550
thamescover2.pjpg	thamescover2.apjpg	3.577009
thamescover3.pjpg	thamescover3.apjpg	6.636730
thebruce.pjpg	thebruce.apjpg	-1.234906

PxJPEG	*APxJPEG image*	*savings in percent*
till1.pjpg	till1.apjpg	5.931842
tm.pjpg	tm.apjpg	8.010993
tobermory.pjpg	tobermory.apjpg	10.655487
tourist.pjpg	tourist.apjpg	9.439468
toys.pjpg	toys.apjpg	4.176418
tree.pjpg	tree.apjpg	7.734048
treecoral.pjpg	treecoral.apjpg	13.377060
trossachs.pjpg	trossachs.apjpg	5.974920
tshell.pjpg	tshell.apjpg	10.011389
tudor.pjpg	tudor.apjpg	5.434752
turkscap.pjpg	turkscap.apjpg	9.843206
twees.pjpg	twees.apjpg	6.136369
twokids.pjpg	twokids.apjpg	6.627915
twokids2.pjpg	twokids2.apjpg	1.509682
vball1.pjpg	vball1.apjpg	5.798769
vball2.pjpg	vball2.apjpg	6.327441
vball3.pjpg	vball3.apjpg	4.537604
wcloud.pjpg	wcloud.apjpg	3.276157
wcpas2.pjpg	wcpas2.apjpg	14.548503
wcpass.pjpg	wcpass.apjpg	11.032840
wdw.pjpg	wdw.apjpg	1.861947
webleaves.pjpg	webleaves.apjpg	8.016907
weed.pjpg	weed.apjpg	11.622188
weed2.pjpg	weed2.apjpg	13.502304
wineshotel.pjpg	wineshotel.apjpg	6.824115

PxJPEG	*APxJPEG image*	*savings in percent*
winter87.pjpg	winter87.apjpg	13.642010
wintergrip.pjpg	wintergrip.apjpg	-0.199143
woodthrush.pjpg	woodthrush.apjpg	2.802191
world.pjpg	world.apjpg	1.022813
yard.pjpg	yard.apjpg	7.616696
zebras.pjpg	zebras.apjpg	11.535427
zoosheep.pjpg	zoosheep.apjpg	6.891399

Bibliography

[1] A. J. Jr Ahumada and H. A Peterson. Luminance-model-based dct quantization for color image compression. *Human Vision, Visual Processing, and Digital Display III*, pages 365–374, 1992.

[2] C. Blakemore and F. W. Campbell. On the existence of neurones in the human visual system selectivity sensitivity to the orientation and size of retinal images. *Journal of Physiology V. 203*, pages 237–260, 1969.

[3] R. J. Clarke. *Transform Coding of Images.* Academic Press, London, 1985.

[4] T. N. Cornsweet. *Visual Perception.* Academic Press, New York, 1970.

[5] S. Daly. The visible differences predictor: an algorithm for the assessment of image fidelity. *Human Vision, Visual Processing, and Digital Display III*, pages 2–15, 1992.

[6] R. L. DeValois, D. G. Albrecht, and L. G. Thorell. Spatial frequency selectivity of cells in the macaque visual cortex. *Vision Research V. 22*, pages 545–559, 1982.

[7] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology V. 160*, pages 106–154, 1962.

[8] N. Jayant, R. J. Safranek, and J. D. Johnston. Signal compression based on models of human perception. *Proc. IEEE*, october 1993.

[9] G. E. Legge and J. M. Foley. Contrast masking in human vision. *Journal of the Optical Society of America V. 70*, pages 1458–1471, 1980.

[10] J. S. Lim. *Two-dimensional Signal and Image Processing.* Prentice Hall, Englewood, NJ, 1990.

[11] H. S. Malvar. *Signal Processing with Lapped Transforms.* Artech House, Boston, 1992.

[12] J. V. Mathews. A perceptual masking threshold model for multichannel image decompositions. *AT&T Bell Labs Technical Memorandum*, 1991.

[13] H. A. Peterson, A. J. Jr Ahumada, and A. B. Watson. An improved detection model for dct coefficient quantization. *Human Vision, Visual Processing, and Digital Display IV*, pages 1–10, 1993.

[14] G. Phillips and H. Wilson. Orientation bandwidths of spatial mechanisms measured by masking. *Journal of the Optical Society of America A V. 1*, pages 226–232, 1984.

[15] R. J. Safranek and J. D. Johnston. A perceptually tuned sub-band image coder with image-dependent quantization and post-quantization data compression. *Proc. ICASSP*, 1989.

[16] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat'l Conv. Rec. Part 4*, pages 142–163, 1959.

[17] C. F. Stromeyer III and B. Julesz. Spatial-frequency masking in vision: Critical bands and spread of masking. *Journal of the Optical Society of America V. 62*, pages 1221–1232, 1972.

[18] G. K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, pages 31–43, april 1991.

[19] A. B. Watson. The cortex transform: Rapid computation of simulated neural images. *Computer Vision Graphics and Image Processing V. 39*, pages 311–327, 1987.

[20] A. B. Watson. Visually optimal dct quantization matrices for individual images. *Proc. Data Compression Conference, IEEE Computer Society Press*, pages 178–187, 1993.